ABSTRACT

A KNOWLEDGE-BASED DESIGNER ASSISTANT SYSTEM WITH APPLICATION TO THE DESIGN OF HELICAL SPRINGS

by
Giridhar Raja

Design problems are generally difficult to solve because the domain of solutions is often infinite (like design of gears, bearings, shafts, springs etc.) and generally requires several iterations before we can arrive at the right solution. The search for the right solution satisfied by several constraints is also time consuming. Thus design intrinsically requires backtracking and several iterations to obtain the desired solution.

In the design of helical springs where the load, deflection, allowable stress, and material are specified, there are an infinite number of solutions. If, in addition to these requirements, the mean coil radius, wire diameter, free height or a combination of these is fixed, the number of solutions are limited or there is only one solution. In the system developed here, each of these cases have been studied and the design procedure is implemented as a rule-based system using VP/Expert. This serves as the front end for the user-friendly application development. The expert system shell links with database files and C programs to suggest the various parameters involved in the design of helical springs. A common procedure adopted is to assume an allowable stress and check for safety based on the material, severity of loading, and the required deflection.

The purpose of this system is to eliminate error prone and time consuming procedure of referring to handbooks, charts and tables and modify iteratively the input constraints until the desired values are obtained. It warns the user of any inconsistencies in the input and the likelihood of buckling in compression springs.

A KNOWLEDGE-BASED DESIGNER ASSISTANT SYSTEM WITH APPLICATION TO THE DESIGN OF HELICAL SPRINGS

by
Giridhar Raja

A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Mechanical Engineering

Department of Mechanical and Industrial Engineering

January 1994

APPROVAL PAGE

A KNOWLEDGE-BASED DESIGNER ASSISTANT SYSTEM WITH APPLICATION TO THE DESIGN OF HELICAL SPRINGS

Giridhar Raja

Dr. Nouri Levy, Thesis Advisor — Date
Associate Professor of Mechanical Engineering, NJIT

Dr. Rajesh N. Dave, Committee Member — Date
Associate Professor of Mechanical Engineering, NJIT

Dr. Zhiming Ji, Committee Member — Date
Assistant Professor of Mechanical Engineering, NJIT

BIOGRAPHICAL SKETCH

Author: Giridhar Raja

Degree: Master of Science in Mechanical Engineering

Date: January 1994

Undergraduate Education and Graduate Education:

- Master of Science in Mechanical Engineering,
  New Jersey Institute of Technology,
  Newark, NJ, 1994

- Bachelor of Engineering in Mechanical Engineering,
  PSG College of Technology,
  Coimbatore, India, 1992

Major: Mechanical Engineering

This thesis is dedicated to my beloved parents.

ACKNOWLEDGMENT

The author wishes to express his sincere gratitude to his advisor, Professor Nouri Levy, for his guidance, friendship, and moral support throughout this research.

The author would like to convey his sincere thanks to Professor Rajesh Dave for his involvement in this research work by offering pertinent suggestions and make it a success.

The author highly appreciates and is very grateful to Professor Zhiming Ji for his interest in this work and his invaluable time and suggestions.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

C	spring index
d	wire diameter, [in]
e	eccentricity of load, [in]
E	modulus of elasticity, [lb/sq.in]
G	modulus of rigidity, [lb/sq.in]
H	solid height of the spring, [in]
H_0	free height, [in]
k	Wahl correction factor
K	spring rate or stiffness, [lb/in]
K_{sf}	factor of safety
L	effective length, [in]
N	number of cycles
n_a	number of active coils in the spring
n_t	number of total coils in the spring
P	nominal load on the spring, [lb]
P_{cr}	critical buckling load, [lb]
P_{max}	maximum load acting on the spring, [lb]
P_{min}	minimum load acting on the spring, [lb]
R	mean coil radius, [in]
S_e	endurance limit of the material, [lb/sq.in]
S_r	range or working stress of the material, [lb/sq.in]
S_u	ultimate tensile strength of the material, [lb/sq.in]
S_v	working stress, [lb/sq.in]
S_y	yield strength of the material, [lb/sq.in]

δ deflection, [in]

δ_0 displacement from zero to maximum load, [in]

δ_{cr} critical buckling deflection, [in]

CHAPTER 1

INTRODUCTION

1.1 Knowledge-Based Systems

1.1.1 Definition

An Expert System is computer system designed to replicate functions performed by a human being. It is used to capture, magnify, and distribute access to judgment. It is also called a Knowledge-Based System (KBS).

Typically, knowledge-based systems enable a user to consult a computer system as they would an expert advisor to diagnose a problem or determine how to solve a problem, do a task, or make a decision. For example, human experts make decisions about to whom to give bank loans or what steps to take to fix a turbine engine malfunction. They also assist and train others to perform tasks, such as adjusting temperature controls in a manufacturing plant. Knowledge-based systems can do these intelligent activities.

Like a human expert, a knowledge-based system can extract additional information from a user by asking questions related to the problem during a consultation. It can also answer questions asked by a user about why certain information is needed. An expert system can make a final recommendation to a user and can explain the reasoning steps followed to reach that conclusion.

Knowledge-based systems are sometimes classified by their application areas, such as engineering, medicine or chemistry. They also can be classified by the generic problem areas they are concerned with, such as diagnosis, monitoring, debugging, repair, instruction, control, prediction, interpretation, design, planning and classification.

Thus, knowledge base is a collection of information, or expert knowledge, about some specific area or field. This knowledge is often composed of both facts, such as found in manuals and textbooks, and heuristics, or "rules of thumb", that a human expert uses to do a task or make a decision. In some cases, heuristics represent years of judgment and experience that are the essence of a human expert. Knowledge engineers are computer specialists with skills to mine heuristics out from human experts and transfer them into a computerized knowledge base.

1.1.2 Selecting an area for KBS Development

Selecting applicable areas for KBS development begins with the examination of needs, including examining answers to questions such as these:

- Is human expertise scarce?
- Is human expertise expensive in relation to a job's value to the company?
- Is the expertise needed in a number of locations?
- Do the job requirements, such as speed and precision, exceed the capabilities of normally available experts?

If it determined that a need exists that can benefit from the development of a computerized system, the next step is to examine the feasibility of the proposed knowledge-based system project. For a KBS project to be feasible, the answer should be "yes" to most of the following questions:

- Do recognized experts exist?
- Can experts do the task better than amateurs and can their skills be taught to others?
- Does the task require reason and informed judgments, as opposed to just common sense?
- Can experts articulate their methods or does the expertise exist in books and manuals?
- Is the task well understood?
- Is the task of manageable size?
- Are typical example cases or situations readily available for testing the validity of a computerized system?

If a proposed project is considered to be feasible, the developer should then determine the risks inherent in a proposed system development project.

1.2.3 Basic KBS Development Activities

Knowledge-based systems are developed by:

- Analyzing or decomposing the situation under study and evaluating relationships
- Reformulating or re conceptualizing the model or representation of the decision situation under study in order to put it into a computer-usable format
- Putting the decision onto the computer

1.2 AI and Expert Systems in Engineering Design

The advent of the technology of Artificial Intelligence (AI) and expert systems has enabled computers to be applied to less deterministic design tasks which require symbolic manipulation and reasoning. The knowledge-based systems generally consist of the following components:

- A knowledge base containing knowledge (facts and information) about a problem domain.
- An inference mechanism (also known as inference engine) for manipulating the stored knowledge to produce solutions to problems.
- A user interface to handle communication with the user in natural language.
- A knowledge acquisition module to assist with the development of the knowledge base.

The three most popular ways of representing knowledge are rules, frames and semantic nets. Rule-based representation is a shallow representation, whereas schemes using frames and semantic nets are deep representations. In a rule-based system, knowledge is represented in terms of facts pertinent to a problem area and rules for manipulating the facts. Facts are asserted in statements which explicitly classify objects or specify the relationships between them, such as "A Flywheel Engine is an Engine", "An Engine is a Prime Mover" and "A Prime Mover has Moving Parts". Rules are modular chunks of knowledge of the form "IF antecedent THEN consequent". In a frame or semantic-net-based system, inferencing is usually achieved by exploiting the inheritance characteristic of frame and net structures mentioned previously.

Some AI techniques applied for design problems are the following: handling design goals and constraints, for creating and validating design solutions, for better representing fundamental design principles, for reasoning with qualitative and uncertain design information and multiple contexts. Design problems are ill-structured in that the mapping of desired functionality onto a physical structure that correctly implements it is generally not straightforward. Furthermore, most design problems call for not only a correct design but a good design. Design looks like a specialization of the software engineering methodology to: design tasks (specified at the knowledge level); design process models (at the algorithm level); and design programs from shells. Thus AI and expert systems come to our rescue to solve engineering design problems using rule-based systems, frames and semantic nets.

The common applications of AI in engineering in which prior research have been carried out are: materials design system, automated configuration generation and innovation, conceptual design, VLSI designs, elevator design, graphic design assistant, cost and manufacturability guide, knowledge-based CAD, automatic tolerance analysis and synthesis, product design and manufacture, Computer-Aided Process Planning, design of forging dies, jigs and fixtures and for designing hydraulic schemes just to mention a few. Thus the implementation of AI and expert systems in design is becoming more widespread.

1.3 Design of Helical Springs

1.3.1 Initial Design Considerations

This module aims at discussing the several factors governing the design of springs and the procedure adopted for a successful design. The several factors affecting the choice of the working stress are as follows:

- Material properties
- Kind of loading
- Corrosion effects
- Creep at elevated temperature
- Vibration and impact effects
- Load eccentricity
- Seriousness of spring failure

In order to specify a spring for a particular application, the designer will be normally aware of the approximate forces and the degree of accuracy required, together with the environmental conditions in which the spring must operate. From this information it is logical to select a material from which the cheapest spring can be made to provide a satisfactory performance in the given environmental conditions. A maximum permissible stress can then be determined, which together with the load requirements will enable suitable dimensions to be selected. It is strongly recommended that, wherever possible, springs should be designed before the details of the housing or other adjacent parts have been finally established. This procedure will minimize restrictions on the design of the spring by ensuring that adequate space is allowed for its accommodation.

Wire should be of circular rather than rectangular section, since the only advantages of rectangular section springs are that they can be designed to give greater volumetric efficiency and that they have superior resistance to buckling. Wire of circular section is cheaper, has a superior surface condition, and is much easier to obtain than that of a rectangular section.

Where design considerations permit the choice of using a spring either in compression or in tension the former is to be preferred, because the stress concentration which occurs at the ends of tension springs will result in loss of performance.

1.3.2 Design Methods

A variety of solution approaches are available for the design of springs:

- Slide-rule-like devices available from spring manufacturers
- Nomographic methods
- Table methods
- Formula method and
- Computer programs and subroutines

Whatever is the method, the above mentioned considerations should be incorporated in the design procedure. When sizing a new spring, one must consider the spring's available working space and the loads and deflections the spring must experience. Refinements dictated by temperature, corrosion, reliability, cost, etc. also enter design considerations. Though several design methods are available, the design adopted by tables, formulas and computer solutions are the most popular methods.

If a spring slide rule is available, it will be of great convenience in making a selection of springs, particularly if a number of designs are to be investigated. These slide rules are based on equations and may usually be obtained by writing to a spring manufacturer. If the allowable stress and load are known, a wide choice of d and D values can be made using the slide rule.

To facilitate selection of helical springs in practical design, tables due to Ross have been developed based on several equations Safe working loads and deflections of helical springs in tension or compression are given in tables in handbooks. Tables have been developed to obtain the d and D using the loads and spring rates. Using these tables, and the basic relationships, the necessary parameters for the spring are calculated.

Charts and nomograms may also be utilized for spring selection. These charts are based on the corrected stress formula and help us to determine loads at given stresses or vice versa, finding rate of deflection or number of coils, finding deflections, tolerances etc.

A design by formula involves manipulating the relationships by separating the known and the unknown in an equation. It is assumed that the springs are in no case stressed beyond the elastic limit (i.e., that they are perfectly elastic) and that they are subject to Hooke's law. The KBS developed follows the method by formula which is the only method that can yield computer solutions and algorithms. The design procedure adopted to calculate the various parameters of the helical spring as discussed by Avallone and Baumeister [8] is illustrated in Chapter 3.

1.4 Development Environment

The general area selected for the KBS development is design of helical springs. The rapid escalation of design and material costs, combined with increasing cost in the spring industry, has generated a substantial interest in alternate method of spring design. This problem of design of helical springs is similar to problems of design of shafts, gears, couplings and several other machine components. This procedure of design can be extended to other similar machine components. The basic structure of the KBS will still remain the same.

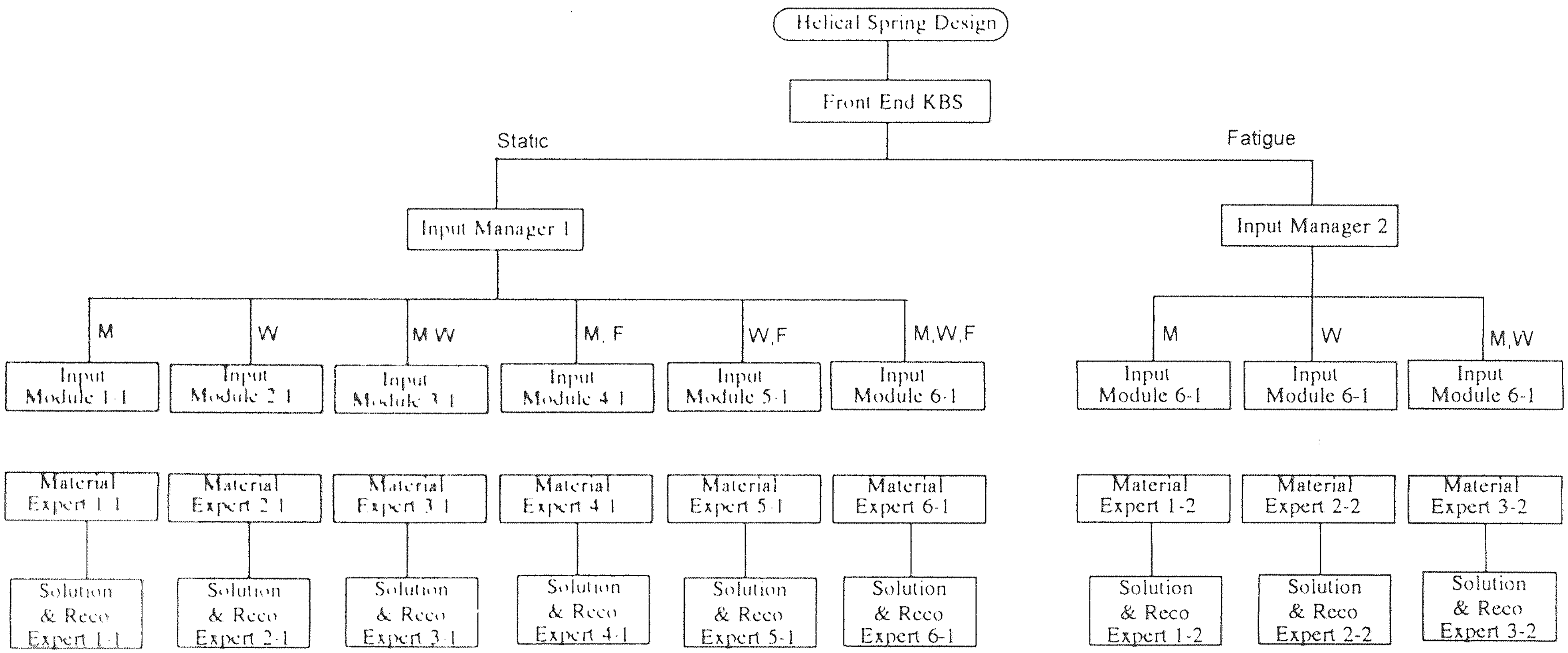
The VP/Expert shell, or software development tool, is used throughout this application. It is a rule-based expert system shell that runs on an IBM Personal Computer or compatible machines. A familiarity with this IBM-PC and its DOS operating system is assumed.

The software links with spreadsheet files and dBASE database files. Such linkage features are commonplace on modern shell development tools. They enable users to build knowledge-based systems that add value to already developed computer-based files and systems. This software is also capable of linking with executable files and batch files created by another software. These capabilities are utilized in retrieving database values and running executable files developed in C to calculate the various parameters involved in the design of springs.

1.5 Structure of the KBS

Once an area, such as the design of helical springs, is isolated, then the precise opportunity or problem in the area is targeted for the KBS application. The block diagram in Figure 2 shows the structure of the KBS for the design of helical springs. This block diagram indicates the path followed by the system based on the user information.

The Introductory KBS briefly describes the system developed and prompts the user for an answer. Based on his response, the system branches to the next knowledge-based system which is the Input Manager. The Input Manager queries the user with choices and depending on the user choice, the system branches or chains to the Input Module. In the input module the user is required to provide all the initial information that is required to proceed with the design calculations. These values provided by the user forms the foundation for the rest of the parameters in the spring problem. This module just accepts the values provided by the user and steps forward to the next module which is the Material Expert. The Material Expert module is chained only when the user is not able to choose or provide a particular material to be used for the spring. This KBS once again queries the user and based on the environment in which the spring is applied, the system prompts the best suited material. If the user provides a particular material to be used, then the Input module is chained directly to the Solution and Recommendation Expert where all calculation and checks are performed to make the final recommendation and display the results. In case the system is chained to the material expert, the system falls on track to the solution and recommendation expert after suggesting the material. The Solution and Recommendation Expert calls or integrates to another executable program developed in C language to calculate the values of the various parameters. These values that are received



Legend:

M Mean Coil Radius

W Wire Diameter

F Free Height

Figure 1 Structure of the Knowledge-Based System by the system are checked for various constraints and the final recommendation or values are displayed.

This seemingly mundane modeling forces an evaluation of each piece of the KBS puzzle. The evaluation often causes the modeling or design process to be repeated over several iterations to model a correct solution. Usually, after one or two passes a credible design solution is sure to be achieved.

CHAPTER 2

DESIGN CONSIDERATIONS

2.1 User Input

2.1.1 Spring Expert Input

The knowledge-based system developed is designed to give the user a wide range of choices in terms of the input (known quantities). The system is designed for two types of users: In the first case where the user has several constraints involved in the design and the second type of user is not knowledgeable in terms of the input. The design procedure is carried out based on the values provided by the user. The user might have several constraints in designing the spring. Some of the key constraints are the following:

- Wire Diameter
- Mean Coil Radius
- Free Height

The wire diameter becomes a constraint when the user holds particular wire sizes in stock to design a spring. When the spring is used round a follower rod the Mean Coil Radius is constrained e.g., a slow speed follower kept in contact with its cam by a helical compression spring. The free height becomes a constraint when the spring is positioned in a hole of a particular depth.

There may be instances where one or more of the values are constrained. Taking all these aspects into consideration the user is given a wide range of choices as given below:

- Wire diameter
- Mean Coil Radius
- Wire diameter and mean coil radius
- Mean coil radius and free height
- Wire diameter and free height

- Wire diameter, free height and mean coil radius

In addition to one of the above choices, the user is required to provide the following parameters: Type of spring - extension or compression, type of load - static or fatigue, maximum load acting on the spring, the required deflection, the choice of material, and the required factor of safety to proceed with the calculations. If the user is unable to decide on a particular material, the system consults a material expert where the user is required to provide more information about the environment in which the spring is used and based on his answers, the appropriate material is suggested. The default factor of safety is assumed to be 2 to begin with, but ultimately the final value is calculated based on the design calculations of the spring.

2.1.2 Material Expert Input

The material expert is a separate knowledge-based system which is developed to help the user in choosing in a material for the design The system recommends the right material depending on the environment in which the spring is used. The material expert advisor is linked to the main system in case the user is unsure about the right material for the application. This KBS is a separate query based system designed to aid the user in making the right choice of the material pertaining to the application.

The KBS recommends the appropriate material based on the following application environment:

- Elevated temperature
- Electrical conductivity
- Corrosive atmosphere
- Clock or motor spring
- High strength

Based on the above mentioned conditions, a material that suits the environment is suggested for an application

2.2 Database Limitations

The database of materials is created for most commonly used materials in design of springs. The commonly used materials are: Music wire, Inconel, Inconel `X', Spring brass, Chrome-silicon alloy steel, Monel, Clock spring steel, Flat spring steel, Oil-tempered wire, Beryllium copper, Phosphor bronze, Stainless steel, Carbon steel, and Hard drawn wire. The database contains the Ultimate tensile strength, Yield Strength, Maximum and the minimum wire sizes available, Endurance limit and the Rigidity Modulus. These values are called by VP/Expert based on the condition that the user specified material matches one of the materials in the database.

Data on material properties and the standard wire sizes are given in Table 1. The values of the Ultimate Strength, S_u in shear are averaged between the maximum and the minimum values. The Yield strength, S_y and the Endurance Limit, S_e of the material are calculated using the empirical relation in Equations 2.1 and 2.2.

$$S_y = 0.75\, S_u \qquad (2.1)$$

$$S_e = 0.5\, S_u \qquad (2.2)$$

2.3 Type of Loading

The types of loading to which the spring is subject is one of the most important factors to consider in designing various types of mechanical springs. The types of loading are divided into the following categories

- Static Loading - Normal temperature
- Static Loading - Elevated temperature
- Fatigue or Repeated Loading

The KBS developed accounts for the various types of loading as discussed below.

Table 1 Database of materials and their properties

MATERIAL	ULTIMATE STRENGTH lb/sq.in	YIELD STRENGTH lb/sq.in	MODULUS OF RIGIDITY , G lb/sq.in	ENDURANC E LIMIT lb/sq.in	MINIMUM DIAMETER in.	MAXIMUM DIAMETER in.
Berryllium Copper	180000	135000	70000000	90000	.00300	.5000
Carbon Steel	132000	100000	115000000	66000	.00300	.1250
Chrome Silicon Alloy Steel	180000	135000	70000000	90000	.03500	.3750
Clock Spring Steel	250000	187000	90000000	125000	.00300	.1250
Flat Spring Steel	220000	165000	115000000	110000	.00300	.1250
Inconel	180000	135000	110000000	90000	.00400	.5000
Inconel X	180000	135000	110000000	90000	.00400	.5630
Monel	150000	112000	95000000	75000	.02800	.5000
Music Wire	270000	203000	115000000	135000	.00400	.2500
Spring Brass	115000	86000	55000000	57500	.02800	.5000
Stainless Steel	100000	75000	110000000	50000	.00300	.1250
Phosphor Bronze	90000	67500	65000000	45000	.00300	.1880
Hard Drawn Wire	200000	150000	115000000	100000	.02800	.6250
Oil Tempered Wire	180000	135000	115000000	90000	.02000	.6250

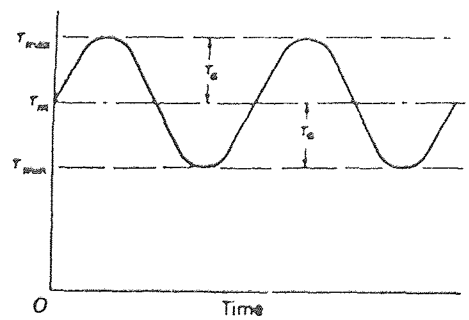


Figure 2 Fluctuating stresses in a helical spring (fatigue loading)

2.3.1 Static Loading - Normal Temperature

This category refers to springs subject to a steady load or one repeated, say, less than 100 to 1,000 times. In such cases the chief problem is usually to avoid excessive set or load loss. Thus, if a helical spring is compressed by a certain amount, the load may drop or relax with time; if the spring is loaded by a constant load, the spring may take a set or creep. In practical design, at normal temperatures if the peak stress in the spring is kept below the elastic limit of the material, trouble from set or relation will seldom occur. It is also a common practice to neglect the stress-concentration effects in calculating stresses.

Thus in this category of loading the spring is designed using formulas specified in the handbooks taking into account the fact that the calculations turn out to be lesser than the elastic limit of the corresponding material.

2.3.2 Static Loading - Elevated Temperature

At elevated temperatures, it is found that the effects of creep or relaxation become much more pronounced than at normal temperatures. This load loss or relaxation is also a function of time. For higher temperatures, materials such as Monel, Inconel, Inconel 'X', Stainless Steel, Music wire and Oil-tempered wire are generally used. Thus, the design procedure remains the same as normal temperature application, but the difference lies in the selection of the material based on the temperature in which the helical spring is used.

2.3.3 Fatigue or Repeated Loading

This category includes spring applications where the load does not remain constant but varies with time. Refer to Figure 2 for fluctuating stresses in helical springs during fatigue loading. During operation it is compressed periodically from a minimum shear stress τ_{min} to a maximum value τ_{max}. In many instances the number of cycles of required life may be small, say several thousands. But the valve spring in an automotive engine must sustain millions of cycles of operation without failure; so they must be designed for infinite life.

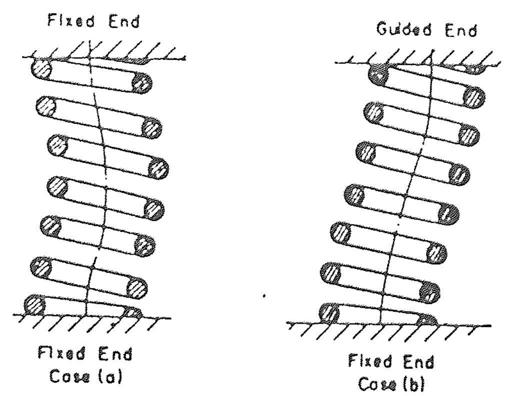


Figure 3 Buckling of springs with fixed or guided ends.

The allowable stress or stress range for helical springs subject to fatigue loading is considerably less than that for springs under static loading. It depends on many factors like: material, number of cycles, surface condition, etc. The number of cycles the spring must sustain directly influences the working stress of the spring. Based on its magnitude, the range stress acting on the spring is calculated according to equation.

The range stress is used as the working stress for the design of helical springs. The calculations are performed based on this range stress and the design parameters are calculated accordingly.

2.4 Eccentricity of Loading

The effect of an eccentric load is to increase the stress on one side of the spring diameter and decrease it on the other. As an approximation by Keysor [1], it may be assumed that where the spring index is fairly large the stress will be increased in the ratio $(1+e/R)$ as compared with the stress for purely axial loading.

2.5 Buckling or Instability of Compression Springs

If compression springs are made too long, instability may occur due to column action under load. In design it is necessary to guard against this by choosing the spring proportions in such a way that the working load will always be less than the critical buckling load. Figure 3 shows buckling of helical springs with fixed to guided ends. The results of an analysis show that the critical deflection δ_{cr} at which instability occurs depends on the ratio of H_0/D where H_0 is the free height of the spring. Curves for finding buckling deflection of helical spring are drawn and these values are tabulated as shown in Table 2 below. Thus for any spring with a given H_0/D, the ratio of δ_{cr}/H_0 may be found and from this, δ_{cr}. Multiplying δ_{cr} by the spring rate gives the theoretical instability or buckling load. This buckling load is compared with the maximum load acting on the

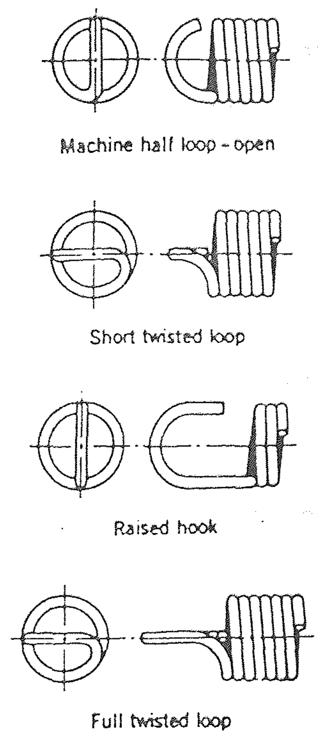


Figure 4 Type of Ends in an Extension Spring.

spring. When the critical load is greater than the maximum load, the spring does not buckle.

Two types of buckling or instability of helical compression springs frequently occur in practice. The first involves a spring with fixed ends and the second case where one end if the spring is fixed and the other end is guided. The latter case is common where springs are used as vibration isolators to support mounted equipment. The design is performed for the latter case where the ratio of δ_{cr}/H_0 changes drastically.

Table 2 Critical buckling deflection of helical springs

H_0/D	2.7-3	3-4	4-5	5-6	>6
δ_{cr}/H_0	0.5	0.28	0.18	0.1	0.05

2.6 Type of Ends

2.6.1 Extension Springs

Extension springs must necessarily have some means of transferring the load from the support to the body of the spring. Although this can be done with a threaded plug or swivel hook, both of these add to the cost of the finished product. So one of the following methods is usually employed:

- Machine half loop - open
- Short twisted loop
- Raised hook
- Full twisted loop

Figure 4 shows the various type of ends in an extension spring. In designing a spring with a hook end, the stress-concentration effect must be considered. By tests performed by Spring Research Association [12], the stress concentration is approximated to 15% (i.e.,

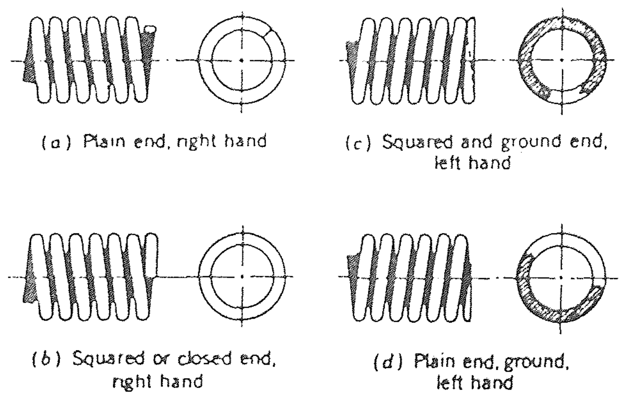


Figure 5 Type of Ends in a Compression Spring.

1.15) for most of the applications and this factor should be included by the user while providing the factor of safety. For example, if the user generally requires a factor of safety to be 2, he is required to key in a value of 2.3 (product of 2 and 1.15) for an extension spring.

2.6.2 Compression Springs

The type of ends should be specified as follows:

- Plain ends
- Plain ends, ground
- Square and ground or forged ends
- Square or closed ends not ground
- 2 1/2 turns set up

Figure 5 shows the various types of ends for a compression spring. The type of end used results in dead or inactive turns at each end of the spring, and these must be added to the number of active turns to obtain the total number of turns. It is customary in the design of springs to neglect the effects of eccentricity of loading due to end turns. It is also customary to neglect the effect of residual stresses caused by heat-treatment or over stressing. Instead, these two effects are usually accounted for an increase in the factor of safety.

CHAPTER 3

DESIGN OF HELICAL SPRINGS

3.1 Static Loading

The helical spring is designed according to the known quantities provided by the user. Several options are provided for the user in terms of the input and based on these known quantities, the design equations are modified to calculate the parameters for the helical spring. The options, as stated earlier, are as follows:

- Mean Coil Radius
- Wire Diameter
- Mean Coil Radius and Free Height
- Wire Diameter and Free Height
- Wire Diameter and Mean Coil Radius
- Wire Diameter, Mean Coil Radius, and Free Height

Apart from one of the above mentioned choices, the user is required to input the following values: Type of spring - extension or compression, maximum load acting on the spring, eccentricity in loading (if any), the maximum deflection, the type of ends and the material required. The design procedure adopted for each of the above cases is discussed below.

3.1.1 Mean Coil Radius known

In this case the mean coil radius is known along with the above mentioned parameters. The design steps are as follows

1 Calculate the wire diameter, d using the relation in Equation 3.1.

$$d^3 = \frac{16\,R\,k\,P}{\pi\,S_v} \qquad (3.1)$$

2 The calculated wire diameter is standardized using the standard wire stock.

3 Calculate the spring constant, K using Equation A.2

4 Calculate the active number of turns, n_a using Equation A.9.

5 Calculate the total number of turns, n_t.

6 Calculate the solid height, H using Equation A.3.

7 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4

8 Calculate the free height, H_0 using Equation A.8.

9 Calculate the spring index, C of the spring using Equation A.5.

10 Recalculate the Wahl correction factor, k using Equation A.7.

11 Calculate the critical load, P_{cr} using Equation A.19.

12 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

13 Check whether the calculated wire diameter lies between the allowable limits for the corresponding material.

14 Check whether the calculated mean coil radius (due to standardization of the wire diameter) is lesser than 5% of the desired value.

15 Check whether the critical load is greater than the maximum load (compression spring).

16 Calculate the effective length of the spring, L using Equation A.20.

Thus using the known values, the design parameters for the spring are evaluated.

3.1.2 Wire Diameter known

The wire diameter when known, along with the above mentioned parameters, the design procedure adopted is as given below. It is assumed that the user specifies the standard wire diameter in the suggested material.

1 Calculate the mean coil radius R using Equation 3.2.

$$R = \frac{S_u \pi d^3}{16 k P K_{sf}} \tag{3.2}$$

2 Calculate the spring constant, K using Equation A.2.

3 Calculate the active number of turns, n_a using Equation A.9.

4 Calculate the total number of turns, n_t.

5 Calculate the solid height, H using Equation A.3.

6 Calculate the displacement from zero to maximum load, δ_0.

7 Calculate the free height, H_0 using Equation A.8.

8 Calculate the spring index, C of the spring using Equation A.5.

9 Recalculate the Wahl correction factor, k using Equation A.7.

10 Calculate the critical load, P_{cr} using Equation A.19.

11 Check whether the calculated factor of safety is greater than 1.25 using Equation A.1 and A.6.

12 Check whether the critical load is greater than the maximum load (compression spring).

13 Calculate the effective length of the spring, L using Equation A.20.

Thus the various design parameters are calculated from the user values.

3.1.3 Mean Coil Radius and Free Height known

The mean coil radius and free height when know along with the required input quantities, the data becomes redundant because the design can be carried out even when the mean coil radius alone is known as seen earlier in Section 3.1.1. In this case the design is performed based on the mean coil radius value and the resulting free height from the calculations is displayed. The design steps is as follows:

1 Calculate the wire diameter, d using Equation 3.1.

2 The calculated wire diameter is standardized using the standard wire stock.

3 Calculate the spring constant, K using Equation A.2.

4 Calculate the active number of turns, n_a using Equation A.9.

5 Calculate the total number of turns, n_t.

6 Calculate the solid height, H using Equation A.3.

7 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

8 Recalculate the free height, H_0 using Equation A.8.

9 Calculate the spring index, C of the spring using Equation A.5.

10 Recalculate the Wahl correction factor, k using Equation A.7.

11 Calculate the critical load, P_{cr} using Equation A.19.

12 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

13 Check whether the calculated mean coil radius (due to standardization of the wire diameter) is lesser than 5% of the desired value.

14 Check whether the critical load is greater than the maximum load (compression springs).

15 Calculate the effective length of the spring, L using Equation A.20.

Thus the calculated free height may be closer to the value desired and the remaining parameters involved in the design are also estimated.

3.1.4 Wire Diameter and Mean Coil Radius known

In this case, the two important parameters in springs are known. This procedure avoids recalculation of values. The design procedure adopted is as follows:

1 Calculate the active number of turns, n_a using Equation 3.3.

$$n_a = \frac{G \delta d^4}{64 R^3 P} \tag{3.3}$$

2 Calculate the spring constant, K using Equation A.2.

3 Calculate the solid height, H using Equation A.3.

4 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

5 Calculate the free height, H_0 using Equation A.8.

6 Calculate the spring index, C of the spring using Equation A.5.

7 Recalculate the Wahl correction factor, k.

8 Calculate the critical load, P_{cr}.

9 Calculate the total number of turns, n_t.

10 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

11 Check whether the critical load is greater than the maximum load (compression springs).

12 Calculate the effective length of the spring, L using Equation A.20.

The design procedure is straightforward in this case and does not involve any recalculations or iterations. Thus the necessary parameters for spring design are calculated and displayed.

3.1.5 Wire Diameter and Free Height known

In this case the design is performed based on the desired free height. The remaining parameters are calculated according to the specified value of free height. The procedure adopted is as follows:

1 Calculate the solid height, H of the spring using Equation 3.4.

$$H = H_0 - \frac{P}{K} \quad \text{(Compression)} \tag{3.4}$$

$$H = H_0 + \frac{P}{K} \quad \text{(Extension)} \tag{3.5}$$

2 Calculate the active number of turns, n_a using Equation 3.6.

$$n_a = \frac{H}{d} \tag{3.6}$$

3 Calculate the mean coil radius, R using Equation 3.7.

$$R^3 = \frac{G\,\delta\,d^4}{64\,n_a\,P} \tag{3.7}$$

4 Calculate the spring index, C using Equation A.5.

5 Calculate the spring constant, K using Equation A.2.

6 Calculate the wahl correction factor, k using Equation A.7.

7 Calculate the solid height, H using Equation A.3.

8 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

9 Recalculate the free height, H_0 using Equation A.8.

10 Calculate the critical load, P_{cr} using Equation A.19.

11 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

12 Check whether the critical load is greater than the maximum load (compression springs).

13 Calculate the effective length of the spring, L using Equation A.20.

The free height calculated may not equal the desired value. In this case, the design is not started with the free height value because the calculation yields inconsistent values of wire diameter. Thus the free height is recalculated and modified based on the values of the remaining parameters.

3.1.6 Wire Diameter, Mean Coil Radius and Free Height known

In this case, the data is redundant and the design procedure does not involve any iterations and assumptions. The design is straight forward and the procedure adopted is as given below:

1 Calculate the solid height, H of the spring using Equation 3.4 - 3.5.

2 Calculate the active number of turns, n_a using Equation 3.6.

3 Calculate the spring index, C using Equation A.5.

4 Calculate the spring constant, K using Equation A.2.

5 Calculate the wahl correction factor, k using Equation A.7.

6 Calculate the solid height, H using Equation A.3.

7 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

8 Calculate the critical load, P_{cr} using Equation A.19.

9 Calculate the total number of turns, n_t.

10 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

11 Check whether the critical load is greater than the maximum load (compression springs).

13 Calculate the effective length of the spring, L using Equation A.20.

Thus the parameters required to design a spring are calculated and displayed.

3.2 Fatigue Loading

The allowable stress or stress range for helical springs subject to fatigue or repeated loading is considerably less than that for springs under static loading. In the case of fatigue loading cycle, the user provides the following information: Type of spring - compression or extension spring, maximum and the minimum load acting on the spring, maximum deflection required, number of cycles the spring should withstand, the factor of safety (if possible), type of ends required, and any recommendation on the material. In addition to the above mentioned values, the user is required to provide one of the or more of the following values:

- Mean Coil Radius
- Wire Diameter
- Mean Coil Radius and Wire Diameter

The allowable stress or the range stress is calculated using Equation 3.8.

$$S_r = \frac{S_u}{0.5\,N^{0.1003}} \qquad (3.8)$$

3.2.1 Mean Coil Radius known

In this case the mean coil radius is known along with the above mentioned parameters. The design steps are as follows:

1 Calculate the wire diameter, d using Equation 3.1.

2 The calculated wire diameter is standardized using the standard wire stock.

3 Calculate the Spring constant, K using Equation A.2.

4 Calculate the active number of turns, n_a using Equation A.9.

5 Calculate the total number of turns, n_t.

6 Calculate the solid height, H using Equation A.3.

7 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

8 Calculate the free height, H_0 using Equation A.8.

9 Calculate the spring index, C of the spring using Equation A.5.

10 Recalculate the Wahl correction factor, k using Equation A.7.

11 Calculate the critical load, P_{cr} using Equation A.19.

12 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

13 Check whether the calculated wire diameter lies between the allowable limits for the corresponding material.

14 Check whether the calculated mean coil radius (due to standardization of the wire diameter) is lesser than 5% of the desired value.

15 Check whether the critical load is greater than the maximum load (compression spring).

16 Calculate the effective length of the spring, L using Equation A.20.

Thus using the known values, the design parameters for the spring are evaluated.

3.2.2 Wire Diameter known

The wire diameter when known, along with the above mentioned parameters, the design procedure adopted is as given below. It is assumed that the user specifies the standard wire diameter in the suggested material.

1 Calculate the mean coil radius R using Equation 3.2.

2 Calculate the spring constant, K using Equation A.2.

3 Calculate the active number of turns, n_a using Equation A.9.

4 Calculate the total number of turns, n_t.

5 Calculate the solid height, H using Equation A.3.

6 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

7 Calculate the free height, H_0 using Equation A.8.

8 Calculate the spring index, C of the spring using Equation A.5.

9 Recalculate the Wahl correction factor, k using Equation A.7.

10 Calculate the critical load, P_{cr} using Equation A.19.

11 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

12 Check whether the critical load is greater than the maximum load (compression spring).

13 Calculate the effective length of the spring, L using Equation A.20.

Thus the various design parameters are calculated from the user values.

3.2.3 Mean Coil Radius and Wire Diameter known

In this case, the two important parameters in springs are known. This procedure avoids recalculation of values. The design procedure adopted is as follows:

1 Calculate the active number of turns, n_a using Equation 3.3.

2 Calculate the spring constant, K using Equation A.2.

3 Calculate the solid height, H using Equation A.3.

4 Calculate the displacement from zero to maximum load, δ_0 using Equation A.4.

5 Calculate the free height, H_0 using Equation A.8.

6 Calculate the spring index, C of the spring using Equation A.5.

7 Recalculate the Wahl correction factor, k using Equation A.7.

8 Calculate the critical load, P_{cr} using Equation A.19.

9 Calculate the total number of turns, n_t.

10 Check whether the calculated factor of safety is greater than 1.25 using Equations A.1 and A.6.

11 Check whether the critical load is greater than the maximum load (compression springs).

12 Calculate the effective length of the spring, L using Equation A.20.

The design procedure is straight forward in this case and does not involve any recalculations or iterations. Thus the necessary parameters for spring design are calculated and displayed.

Thus in this chapter the detailed step-by-step procedure of the design problem was discussed with the necessary conditions taken into account. The results are discussed in Chapter 4.

CHAPTER 4

DISCUSSION OF RESULTS

4.1 Static Loading Example

The system was tested for a set of input values and the output was recorded. The system was run for both static and fatigue load problems and the results obtained were satisfactory and they matched the results obtained by hand calculations. In this case, the load was given to be static and the input choice was the wire diameter. The input values and the output values are given below:

Input:

Spring Type	Extension
Material	Music Wire
Load Type	Static load at normal temperature
Load, P	75
Deflection, δ	1.45
Eccentricity, e	0
Wire diameter, d	0.125
Desired K_{sf}	2.35
Ends	Short twisted loop

Output:

Mean Coil Radius, R	0.254861
Number of active turns, n_a	61
Number of total turns, n_t	61.25
Free Height, H_0	9.425072
Displacement from zero to max, δ_0	1.74
Spring Index, C	4.0777

Wahl correction factor, k	1.3945
Calculated K_{sf}	1.685
Effective Length, L	98.451

4.2 Fatigue Loading Example

In the fatigue loading example the spring type was chosen to be compression and the choice of values was the wire diameter and the mean coil radius. The results obtained are as follows:

Input:

Spring type	Compression
Material	Inconel X
Load type	Fatigue
Maximum load	50
Minimum load	30
Number of cycles	40,000
Deflection, δ	1.25
Wire diameter, d	0.125
Mean coil radius, R	0.375
Desired, K_{sf}	1.45
Ends	Square or closed ends

Output:

Number of active turns, n_a	24
Total number of turns, n_t	25.5
Solid height, H	2.9839
Free height, H_0	6.7339
Spring Index, C	6
Wahl correction factor, k	1.25

Effective length, L 56.245

Displacement from zero to max load, δ_0 3.75

The above mentioned results are obtained by trial and error procedure with the choice of material as the values obtained should satisfy several constraints like: buckling problem, the calculated factor of safety should be satisfactory, and the calculated values of wire diameter should lie within the minimum and the maximum wire sizes available for the corresponding material. Thus it was seen that the values obtained from the system yielded satisfactory results.

CHAPTER 5

RECOMMENDATIONS AND CONCLUSIONS

1 In this system, the user is required to suggest a desired factor of safety value which will used to perform the calculations for the spring parameters. If the user fails to provide a value, a default value of 2 is assumed to proceed with the calculations. The suggested approach to this problem will be to create a separate knowledge-based system to help the user arrive at an exact value. Based on the application type for example, the spring may deserve a higher value of factor of safety . If the user is skeptical about the factor of safety, he can be directed to a factor of safety expert which can suggest the best suited value for the application. Some of the features the factor of safety expert may take into account are:

- Life of the spring
- Cost
- Application environment - impact, fatigue or static loads
- Spring type - extension or compression springs etc.

Taking each of the above mentioned factors into account, the material expert can suggest the best suited value for the application development.

2 One of the outstanding factors to fully specify a spring together with the recommended form is the tolerance that a specification should take. There are two classes of tolerances which should cover the majority of the applications: the first class is for springs where close agreement with the design is important and the second class where close agreement with the design requirements is not so important, and to be used where the functional requirements permit. Tolerances are virtually important on the load/length, spring rate and free length. The present system does not account for the tolerances on the various dimensions of the spring. But one of the suggestions is to

account for the tolerances on various dimensions like mean coil radius, wire diameter, spring rate and the free length. These tolerances are calculated either using tables or empirical relationships which can be obtained from the handbooks.

3 The results obtained in the solution and recommendation expert can be shipped to another KBS where the results are checked for consistency. If the values obtained are inconsistent, appropriate suggestions can be made to obtain satisfactory results. For example, if the desired factor of safety value is not obtained, pertinent suggestions like choosing a material with better mechanical properties can be made to obtain the desired value. Such recommendations can be made in the separate KBS that is developed.

In the design of helical springs where the load, deflection, allowable stress, and the material are specified, there are an infinite number of solutions. In this system, equations and the method of design are adopted from Avallone and Baumeister [8]. If, in addition to these requirements, the wire diameter, mean coil radius or free height or a combination of these is fixed, there is only one solution. In the design procedure adopted, when the user specifies an extension spring, the stress concentration factor due to the ends is accounted by the user himself when specifying a factor of safety. A value of 1.15 is a reasonable value for most of the applications. Because of time, I confined myself to the problem presented here. I hope that this application, will encourage others to continue the work in developing systems which have similar system hierarchy such as shafts, couplings, gears, etc. and several other machine components.

APPENDIX A

EQUATIONS FOR DESIGN OF HELICAL SPRINGS

$$S_v = \frac{16\,R\,k\,P}{\pi\,d^3} \qquad \text{(A.1)}$$

$$K = \frac{\Delta P}{\Delta \delta} \qquad \text{(A.2)}$$

$$H = n\,d \qquad \text{(A.3)}$$

$$\delta_0 = \frac{P_{max}}{K} \qquad \text{(A.4)}$$

$$C = \frac{2\,R}{d} \qquad \text{(A.5)}$$

$$K_{sf} = \frac{S_v\,\pi\,d^3}{16\,R\,k\,P} \qquad \text{(A.6)}$$

$$k = \frac{4\,C - 1}{4C - 4} + \frac{0.615}{C} \qquad \text{(A.7)}$$

$$H_0 = H + \delta_0 \qquad \text{(A.8)}$$

$$n_a = \frac{G\,\delta\,d^4}{64\,K\,R^3} \qquad \text{(A.9)}$$

$n_t = n_a + 0.5$ (Plain End) (A.10)

$= n_a + 1.0$ (Plain and ground) (A.11)

$= n_a + 1.5$ (Square end) (A.12)

$= n_a + 2$ (Square and ground) (A.13)

$= n_a + 5$ (2 1/2 turns setup) (A.14)

$= n_a + 0.5$ (Machine half loop) (A.15)

$= n_a + 0.5$ (Short twisted loop) (A.16)

$$= n_a + 0.25 \quad \text{(Raised hook)} \qquad (A.17)$$

$$= n_a + 0.25 \quad \text{(Full twisted loop)} \qquad (A.18)$$

$$P_{cr} = H_0 \, \delta_{cr} \qquad (A.19)$$

$$L = 2 \pi R \, n_a \qquad (A.20)$$

APPENDIX B

KNOWLEDGE-BASED SYSTEM FILES

```
!KBS to display the capabilities of the Design of Helical Springs
!and its limitations.
!Saved as MAINSPRG.KBS

        BKCOLOR = 0;
        RUNTIME;
        ENDOFF;

!------------------------- ACTIONS BLOCK ----------------------

ACTIONS

    CLS
    COLOR = 6
    DISPLAY "

            EXPERT SYSTEM FOR THE DESIGN OF HELICAL SPRINGS"
    COLOR = 11
    DISPLAY "

      The Expert System calculates the parameters for the design of

      Helical Springs. The user is given choices in terms of the known

      values. The calculations are performed based on these values

      and the final values are displayed. If the calculations turn out

      to be inconsistent with the design, the user is required to change

      certain values which will ultimately yield the right values."

    COLOR = 6
    DISPLAY "


              Press any key to proceed with the design~"

    CLS
    COLOR = 15
    FIND Type_load
    FIND Loading
;
!---------------------------- RULES BLOCK -----------------------------
```

```
Rule Load1
    IF Load_type = 1 OR
      Load_type = 2
    THEN Loading = Static
        SAVEFACTS Static
        CHAIN Spring;

Rule Load2
    IF Load_type = 3
    THEN Loading = Fatigue
        SAVEFACTS Fatigue
        CHAIN Fatigue;

Rule Type1
    IF Load_type = 1
    THEN Type_load = Static_load_at_normal_temperature;

Rule Type2
    IF Load_type = 2
    THEN Type_load = Static_load_at_elevated_temperature;

Rule Type3
    IF Load_type = 3
    THEN Type_load = Fatigue_or_repeated_load;

!------------------------- QUESTIONS BLOCK -----------------------

ASK Load_type: "What is the type of loading acting on the spring?

            1 = Static Load at Normal Temperature

            2 = Static Load at Elevated Temperature

            3 = Fatigue or Repeated Loading
";
CHOICES Load_type: 1,   2,   3;
```

```
!MAIN KBS TO DETERMINE THE CONSTRAINTS FROM THE USER AND CHAIN
!THE VARIOUS KNOWLEDGE BASES ACCORDINGLY
!Saved as Fatigue.KBS
!       EXECUTE;
        RUNTIME;
        BKCOLOR = 3;
!       ENDOFF;

!*********************** ACTIONS BLOCK ************************
ACTIONS

    COLOR = 20
    DISPLAY "
                        NOTE!!!!!!
"
    COLOR = 15
    DISPLAY "
        Please choose one of the following constraints/known values
          to proceed with the design. According to the choice, the
          corresponding values are to be fed through the keyboard.

                    Press any key to continue~"

    CLS
    COLOR = 0
    FIND Type
    FIND Constraint
    CLS
    FIND Link
;

!************************** RULES BLOCK ***********************
Rule 1
    IF Constraint = 1
    THEN Link = Spring1
        FIND Mean_rad
        SAVEFACTS Fatigue1
        CHAIN Fatigue1;

Rule 2
    IF Constraint = 2
    THEN Link = Spring2
        FIND Wire_dia
        SAVEFACTS Fatigue2
        CHAIN Fatigue2;

Rule 3
    IF Constraint = 3
    THEN Link = Spring3
        FIND Mean_rad
        FIND Wire_dia
        SAVEFACTS Fatigue3
        CHAIN Fatigue3;
```

```
!*********************** QUESTIONS BLOCK ************************
ASK Type: "What type of helical spring do you require?";
CHOICES Type: Compression,    Extension;

ASK Mean_rad: "What is the value of the mean coil radius? ";

ASK Wire_dia: "What is the wire diameter of the spring? ";

ASK Constraint: "Choose one of the following:
  1 = Load,deflection,material,safety factor & mean coil radius

  2 = Load,deflection,material,safety factor & wire diameter

  3 = Load,deflection,material,safety factor,wire diameter
       & mean coil radius
";
CHOICES Constraint: 1,  2,  3;
```

```
! Spring Design - Session 1 with an improved user interface
! Saved as Fatigue1.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK ======================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

            Press any key to continue.~
     "
   CLS
   COLOR = 15
   CLS
   FIND Load_max
   CLS
   FIND Load_min
   CLS
   FIND Fatigue_life
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!======================= RULES BLOCK ======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Cycle1
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Fatmat1;
```

```
Rule 2
   If End_type = 1
   Then Ends = Square_and_ground_ends;

Rule 3
   If End_type = 2
   Then Ends = Square_or_closed_ends;

Rule 4
   If End_type = 3
   Then Ends = Plain_ends;

Rule 5
   If End_type = 4
   Then Ends = Plain_ends_not_ground;

Rule 6
   If End_type = 5
   Then Ends = Two_&_half_turns;

Rule 7
   If Fac_safe = Yes
   Then Safety = Yes
      FIND Factor
   Else Safety = No
      Factor = (2);

Rule 8
   If Mat_choice = 1
   Then Material = Stainless_steel;

Rule 9
   If Mat_choice = 2
   Then Material = Monel;

Rule 10
   If Mat_choice = 3
   Then Material = Inconel;

Rule 11
   If Mat_choice = 4
   Then Material = Inconel_X;

Rule 12
   If Mat_choice = 5
   Then Material = Music_wire;

Rule 13
   If Mat_choice = 6
   Then Material = Carbon_steel;

Rule 14
   If Mat_choice = 7
   Then Material = Chrome_silicon_alloy_steel;
```

```
Rule 15
    If Mat_choice = 8
    Then Material = Flat_spring_steel;

Rule 16
    If Mat_choice = 9
    Then Material = Clock_spring_steel;

Rule 17
    If Mat_choice = 10
    Then Material = Spring_brass;

Rule 18
    If Mat_choice = 11
    Then Material = Beryllium_copper;

Rule 19
    If Mat_choice = 12
    Then Material = Phosphor_bronze;

Rule 20
    If Mat_choice = 13
    Then Material = Hard_drawn_wire;

Rule 21
    If Mat_choice = 14
    Then Material = Oil_tempered_wire;

Rule 22
    If End_type = 6
    Then Ends = Machine_half_loop;

Rule 23
    If End_type = 7
    Then Ends = Short_twisted_loop;

Rule 24
    If End_type = 8
    Then Ends = Raised_hook;

Rule 25
    If End_type = 9
    Then Ends = Full_twisted_loop;

!====================== QUESTIONS BLOCK =======================

ASK Load_max: "What is the maximum load acting on the spring? ";

ASK Load_min: "What is the minimum load acting on the spring? ";

ASK Fatigue_life: "What is the number of cycles it should withstand? ";

ASK Defln: "What is the maximum deflection that is required? ";
```

ASK Mat_sugg: "Do you have any recommendation for the spring material?

Note: If there are no suggestions, a material would be suggested according to the type of application environment

";

CHOICES Mat_sugg: Yes, No;

ASK Mat_choice: "What material do you wish to use?

The materials in the database are:

1. Stainless_steel 2. Monel
3. Inconel 4. Inconel_X
5. Music_wire 6. Carbon_steel
7. Chrome_silicon_alloy_steel
8. Flat_spring_steel 9. Clock_spring_steel
10. Spring_brass 11. Beryllium_copper
12. Phosphor_bronze 13. Hard_drawn_wire
14. Oil_tempered_wire

";

CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

1 = Square & ground or forged ends (Compression springs)
2 = Squared or closed ends not ground (Compression springs)
3 = Plain Ends (Compression springs)
4 = Plain Ends ground (Compression springs)
5 = 2 1/2 turns set up (Compression springs)
6 = Machine half loop (Extension springs)
7 = Short twisted loop (Extension springs)
8 = Raised hook (Extension springs)
9 = Full twisted loop (Extension springs)

";

CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?".

```
! Spring Design - Session 1 with an improved user interface
! Saved as Fatigue2.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK ======================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                    INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

               Press any key to continue.~
    "
   CLS
   COLOR = 15
   CLS
   FIND Load_max
   CLS
   FIND Load_min
   CLS
   FIND Fatigue_life
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!====================== RULES BLOCK ======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Cycle2
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Fatmat2;
```

```
Rule 2
   If End_type = 1
   Then Ends = Square_and_ground_ends;

Rule 3
   If End_type = 2
   Then Ends = Square_or_closed_ends;

Rule 4
   If End_type = 3
   Then Ends = Plain_ends;

Rule 5
   If End_type = 4
   Then Ends = Plain_ends_not_ground;

Rule 6
   If End_type = 5
   Then Ends = Two_&_half_turns;

Rule 7
   If Fac_safe = Yes
   Then Safety = Yes
      FIND Factor
   Else Safety = No
      Factor = (2);

Rule 8
   If Mat_choice = 1
   Then Material = Stainless_steel;

Rule 9
   If Mat_choice = 2
   Then Material = Monel;

Rule 10
   If Mat_choice = 3
   Then Material = Inconel;

Rule 11
   If Mat_choice = 4
   Then Material = Inconel_X;

Rule 12
   If Mat_choice = 5
   Then Material = Music_wire;

Rule 13
   If Mat_choice = 6
   Then Material = Carbon_steel;

Rule 14
   If Mat_choice = 7
   Then Material = Chrome_silicon_alloy_steel;
```

```
Rule 15
    If Mat_choice = 8
    Then Material = Flat_spring_steel;

Rule 16
    If Mat_choice = 9
    Then Material = Clock_spring_steel;

Rule 17
    If Mat_choice = 10
    Then Material = Spring_brass;

Rule 18
    If Mat_choice = 11
    Then Material = Beryllium_copper;

Rule 19
    If Mat_choice = 12
    Then Material = Phosphor_bronze;

Rule 20
    If Mat_choice = 13
    Then Material = Hard_drawn_wire;

Rule 21
    If Mat_choice = 14
    Then Material = Oil_tempered_wire;

Rule 22
    If End_type = 6
    Then Ends = Machine_half_loop;

Rule 23
    If End_type = 7
    Then Ends = Short_twisted_loop;

Rule 24
    If End_type = 8
    Then Ends = Raised_hook;

Rule 25
    If End_type = 9
    Then Ends = Full_twisted_loop;

!====================== QUESTIONS BLOCK ========================

ASK Load_max: "What is the maximum load acting on the spring? ";

ASK Load_min: "What is the minimum load acting on the spring? ";

ASK Fatigue_life: "What is the number of cycles it should withstand? ";

ASK Defln: "What is the maximum deflection that is required? ";
```

ASK Mat_sugg: "Do you have any recommendation for the spring material?

Note: If there are no suggestions, a material would be
suggested according to the type of application
environment

";
CHOICES Mat_sugg: Yes, No;

ASK Mat_choice: "What material do you wish to use?

The materials in the database are:

1. Stainless_steel 2. Monel
3. Inconel 4. Inconel_X
5. Music_wire 6. Carbon_steel
7. Chrome_silicon_alloy_steel
8. Flat_spring_steel 9. Clock_spring_steel
10. Spring_brass 11. Beryllium_copper
12. Phosphor_bronze 13. Hard_drawn_wire
14. Oil_tempered_wire

";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

1 = Square & ground or forged ends (Compression springs)
2 = Squared or closed ends not ground (Compression springs)
3 = Plain Ends (Compression springs)
4 = Plain Ends ground (Compression springs)
5 = 2 1/2 turns set up (Compression springs)
6 = Machine half loop (Extension springs)
7 = Short twisted loop (Extension springs)
8 = Raised hook (Extension springs)
9 = Full twisted loop (Extension springs)

";
CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";

```
! Spring Design - Session 1 with an improved user interface
! Saved as Fatigue3.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!==================== ACTIONS BLOCK =====================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                  INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

               Press any key to continue.~
     "
   CLS
   COLOR = 15
   CLS
   FIND Load_max
   CLS
   FIND Load_min
   CLS
   FIND Fatigue_life
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!====================== RULES BLOCK =====================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Cycle3
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Fatmat3;
```

```
Rule 2
    If End_type = 1
    Then Ends = Square_and_ground_ends;

Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 8
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 9
    If Mat_choice = 2
    Then Material = Monel;

Rule 10
    If Mat_choice = 3
    Then Material = Inconel;

Rule 11
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 12
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 13
    If Mat_choice = 6
    Then Material = Carbon_steel;

Rule 14
    If Mat_choice = 7
    Then Material = Chrome_silicon_alloy_steel;
```

Rule 15
If Mat_choice = 8
Then Material = Flat_spring_steel;

Rule 16
If Mat_choice = 9
Then Material = Clock_spring_steel;

Rule 17
If Mat_choice = 10
Then Material = Spring_brass;

Rule 18
If Mat_choice = 11
Then Material = Beryllium_copper;

Rule 19
If Mat_choice = 12
Then Material = Phosphor_bronze;

Rule 20
If Mat_choice = 13
Then Material = Hard_drawn_wire;

Rule 21
If Mat_choice = 14
Then Material = Oil_tempered_wire;

Rule 22
If End_type = 6
Then Ends = Machine_half_loop;

Rule 23
If End_type = 7
Then Ends = Short_twisted_loop;

Rule 24
If End_type = 8
Then Ends = Raised_hook;

Rule 25
If End_type = 9
Then Ends = Full_twisted_loop;

!====================== QUESTIONS BLOCK =======================

ASK Load_max: "What is the maximum load acting on the spring? ";

ASK Load_min: "What is the minimum load acting on the spring? ";

ASK Fatigue_life: "What is the number of cycles it should withstand? ";

ASK Defln: "What is the maximum deflection that is required? ";

```
ASK Mat_sugg: "Do you have any recommendation for the spring material?

            Note: If there are no suggestions, a material would be
                suggested according to the type of application
                environment
";
CHOICES Mat_sugg: Yes,  No;

ASK Mat_choice: "What material do you wish to use?

        The materials in the database are:

        1. Stainless_steel          2. Monel
        3. Inconel                  4. Inconel_X
        5. Music_wire               6. Carbon_steel
        7. Chrome_silicon_alloy_steel
        8. Flat_spring_steel        9. Clock_spring_steel
        10. Spring_brass            11. Beryllium_copper
        12. Phosphor_bronze         13. Hard_drawn_wire
        14. Oil_tempered_wire

";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

            1 = Square & ground or forged ends    (Compression springs)
            2 = Squared or closed ends not ground (Compression springs)
            3 = Plain Ends                  (Compression springs)
            4 = Plain Ends ground           (Compression springs)
            5 = 2 1/2 turns set up          (Compression springs)
            6 = Machine half loop           (Extension springs)
            7 = Short twisted loop          (Extension springs)
            8 = Raised hook                 (Extension springs)
            9 = Full twisted loop           (Extension springs)
";
 CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";
```

```
! KBS to help the user to choose a material according to the
! requirements
! Saved as Fatmatl1.KBS
   EXECUTE;
   BKCOLOR = 3;
   RUNTIME;
!  ENDOFF;

!********************** ACTIONS BLOCK ***********************
ACTIONS

   BCALL Delete1," "
LOADFACTS Matl
CLS
COLOR = 15
FIND Material
CLS

COLOR = 15
DISPLAY "







           The suggested material for the application is:"

   COLOR = 11
   DISPLAY "

                  =====> {Material}~"
   SAVEFACTS Design
   CHAIN Cycle1
;

!*********************** RULES BLOCK ***********************

Rule 1
   If Clock = Yes
   Then Material = Clock_spring_steel;

Rule 2
   If Application = 2
   Then Material = Beryllium_copper;

Rule 3
   If Application = 3
   Then Material = Spring_brass;

Rule 4
   If Application = 1
   Then Appn = Elevated
```

```
        FIND Temperature;

Rule 5
    If Temperature = 1
    Then Material = Music_wire;

Rule 6
    If Temperature = 2
    Then Material = Chrome_silicon_alloy_steel;

Rule 7
    If Temperature = 3
    Then Material = Inconel;

Rule 8
    If Temperature = 4
    Then Material = Inconel_X;

!*********************** QUESTIONS BLOCK ************************

ASK Application: "What is the type of application environment?

            1 = Elevated temperature
            2 = Corrosive environment
            3 = Conductive environment (electricity & heat)
";
CHOICES Application: 1,  2,   3;

ASK Temperature: "What is the temperature range in which the spring
is used?

            1 = Temperature of 250-350 F
            2 = Temperature upto 450 F
            3 = Temperature upto 700 F
            4 = Temperature upto 900 F
";
CHOICES Temperature:1,  2,  3,  4;

ASK Clock: "Is the spring used in clock or a motor?";
CHOICES Clock: Yes,     No;
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Cycle1.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!======================== ACTIONS BLOCK ========================
ACTIONS
        LOADFACTS Fatigue
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Fatigue1

        COLOR = 15
        DISPLAY "
                    The user defined values are:

    Material            =        {Material}
    Load Type           =        {Type_load}
    Maximum Load        =        {Load_max}
    Minimum Load        =        {Load_min}
    No. of cycles       =        {Fatigue_life}
    Deflection          =        {Defln}
    Mean Coil Radius    =        {Mean_rad}
    Factor of safety    =        {Factor}
    Ends                =        {Ends}"

        COLOR = 11
        DISPLAY "

                    Press any key to continue~"

        BCALL File," "
        GET Material=Materials,matl,ALL
    CLS
    COLOR = 14
        DISPLAY "
                        MATERIAL PROPERTIES

                Material:                {Materials}
                Ultimate Shear Stress:       {Ultimate}
                Yield Stress:              {Yield}
                Rigidity Modulus:            {Rigidity}
                Endurance limit:            {Enduran}
"
    COLOR = 10
    DISPLAY "
                    Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
```

```
     SHIP File1,Rigidity
     SHIP File1,Enduran
     SHIP File1,Load_max
     SHIP File1,Load_min
     SHIP File1,Fatigue_life
     SHIP File1,Defln
     SHIP File1,Mean_rad
     SHIP File1,Factor

 CALL Cycle1,""

    RECEIVE File2,wire_dia
    RECEIVE File2,turns
    RECEIVE File2,length
    RECEIVE File2,solidht
    RECEIVE File2,disp
    RECEIVE File2,freeht
    RECEIVE File2,index
    RECEIVE File2,wahl
    RECEIVE File2,safety
    RECEIVE File2,ratio_rad
    RECEIVE File2,critload
    RECEIVE File2,reco

 FIND Diameter
 FIND Radius
 FIND Load
 FIND Total

 CLS
 COLOR = 15
 DISPLAY "
                 Design of Helical Springs


    The design values for the specified spring are the following:

     Wire Diameter                    =   {wire_dia}
     No. of active turns              =   {turns}
     Total # of turns                 =   {Total}
     Solid height                     =   {solidht}
     Displacement from zero to max    =   {disp}
     Effective length                 =   {length}
     Free height                      =   {freeht}
     Spring Index                     =   {index}
     Wahl factor                      =   {wahl}
     Factor of Safety                 =   {safety}

COLOR = 11
 DISPLAY "

          Press any key to continue with the design ~"

 CLS
```

```
   COLOR = 14
   FIND Val
   WCLOSE 1
;
!======================== RULES BLOCK ========================
Rule Change1
      IF Change_val = 1
      THEN Val = Change1
         CHAIN Ftmatrl1;

Rule Change2
      IF Change_val = 2
      THEN Val = Change2
         CHAIN Fatigue;

Rule Change3
      IF Change_val = 3
      THEN Val = Change3
         CLS
         COLOR = 27
         DISPLAY "
                    THANK YOU!!!
"
         COLOR = 14
         DISPLAY "

                 for using the KBS for the
"
         COLOR = 31
         DISPLAY "
                 DESIGN OF HELICAL SPRINGS"

         COLOR = 11
         DISPLAY "

                 Developed by: GIRIDHAR RAJA
         ME Student, New Jersey Institute of Technology,NJ

               Under the guidance of Dr. Nouri Levy
        Associate Professor, Department of Mechanical Engineering
              New Jersey Institute of Technology, NJ.
~";

Rule Dia
   IF Material = (Materials) AND
     wire_dia > (Mindia) AND
     wire_dia < (Maxdia)
   THEN Diameter = Inrange
   ELSE Diameter = Outrange
      CLS
      COLOR = 11
      DISPLAY "

      For {Material},
```

```
  ===>        Minimum diameter = {Mindia}

  ===>        Maximum diameter = {Maxdia}

  ===>        Calculated wire diameter = {wire_dia}

          We see that the wire diameter does not lie within the

                  minimum and maximum values.

        Choose a different material with higher mechanical properties
"
        COLOR = 5
        DISPLAY "
                    Press any key to continue~"
        CHAIN Ftmatrl1;

Rule Ends1
    IF Ends = Square_and_ground_ends
    THEN Total = (turns + 2);

Rule Ends2
    IF Ends = Square_or_closed_ends
    THEN Total = (turns + 1.5);

Rule Ends3
    IF Ends = Plain_ends_not_ground
    THEN Total = (turns + 0.5);

Rule Ends4
    IF Ends = Plain_ends_ground
    THEN Total = (turns + 1);

Rule Ends5
    IF Ends = Two_&_half_turns
    THEN Total = (turns + 5);

Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.5)
    ELSE Total = (turns + 0.25);

Rule Radius
    IF ratio_rad <= 1.05 AND
      ratio_rad >= 0.95
    THEN Radius = OK
    ELSE Radius = Not_OK
        CLS
        COLOR = 11
        DISPLAY "

        For {Material},
```

```
        Due to standardization of the wire diameter, the value

      of Mean Coil Radius is greater than 5% of the required value.

              So, please choose a different material."

      COLOR = 15
      DISPLAY "

                    Press any key to continue~"
      CLS
      FIND Val;

Rule Load1
   IF Type = Compression AND
     critload < (Load_max) AND
     safety > 1.25
   THEN Load = Poor
      CLS
      COLOR = 11
      DISPLAY "

                     Design is very unsafe!!"
      COLOR = 15
      DISPLAY "

      For {Material},

              Critical Load      =        {critload}

              Maximum Load       =        {Load_max}

              Factor of Safety   =        {safety}

         The Critical load is lesser than the Maximum load

       acting on the spring. This causes buckling of the spring.

        Please increase the the Mean Coil Radius or try with a

                      different material."

      COLOR = 10
      DISPLAY "

                    Press any key to continue~";

Rule Load2
   IF Type = Compression AND
     critload < (Load_max) AND
     safety < 1.25
   THEN Load = Poor
      CLS
      COLOR = 11
```

```
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

                Critical Load       =       {critload}

                Maximum Load        =       {Load_max}

                Factor of safety    =       {safety}

         The Critical Load is lesser than the Maximum Load acting

            on the spring. This causes buckling of the spring.

             Please increase the Mean Coil Radius or try with

                     a different material"

        COLOR = 10
        DISPLAY "
                  Press any key to continue~";

Rule Load3
    IF Type = Compression AND
      critload > (Load_max) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                      Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Calculated factor of safety     =      {safety}

             The Factor of Safety is too low which makes the

                        design very unsafe.

         Please choose a material with better mechanical properties

             like Young's Modulus & Ultimate Shear Strength.

        COLOR = 10
        DISPLAY "
                  Press any key to continue~";
```

```
Rule Load4
   IF Type = Compresion AND
     critload > (Load_max) AND
     safety > 1.25
   THEN Load = Good
      CLS
      COLOR = 11
      DISPLAY "
                  Design is very safe!!"
      COLOR = 15
      DISPLAY "

      For {Material},

              Calculated factor of safety  =   {safety}

              Critical Load             =   {critload}

              Maximum Load               =   {Loadmax}


           The Critical Load is greater than the Maximum Load

            acting on the spring and the Factor of Safety

                   is also fairly high."
           COLOR = 10
           DISPLAY "
                Press any key to view the results.~";

Rule Load5
   IF Type = Extension AND
     safety < 1.25
   THEN Load = Poor
      CLS
      COLOR = 11
      DISPLAY "



                  Design is very unsafe!!"
      COLOR = 15
      DISPLAY "



              The calculated factor of safety is very low.

   Please choose a different material with better mechanical properties.



               Press any key to still view the results~";
```

```
Rule Load6
   IF Type = Extension AND
     safety > 1.25
   THEN Load = Good
      CLS
      COLOR = 11
      DISPLAY "


                 Design is very safe!!"

      COLOR = 15
      DISPLAY "


            The calculated factor of safety is OK.



             Press any key to view the results~";

!====================== QUESTIONS BLOCK ====================

ASK Change_val: "Do you wish to change any one of the following?

            1 = Material

            2 = User values

            3 = None
";
CHOICES Change_val: 1, 2, 3;
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Cycle2.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!======================== ACTIONS BLOCK ======================
ACTIONS
        LOADFACTS Fatigue
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Fatigue2

        COLOR = 15
        DISPLAY "
                    The user defined values are:

     Spring Type        =        {Type}
     Material           =        {Material}
     Load Type          =        {Type_load}
     Maximum Load       =        {Load_max}
     Minimum Load       =        {Load_min}
     No. of cycles      =        {Fatigue_life}
     Deflection         =        {Defln}
     Wire Diameter      =        {Wire_dia}
     Factor of safety   =        {Factor}
     Ends               =        {Ends}"

        COLOR = 11
        DISPLAY "

                    Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
     CLS
     COLOR = 14
        DISPLAY "
                    MATERIAL PROPERTIES

              Material:                 {Materials}
              Ultimate Shear Stress:    {Ultimate}
              Yield Stress:             {Yield}
              Rigidity Modulus:         {Rigidity}
              Endurance limit:          {Enduran}

     COLOR = 10
     DISPLAY "
                    Press any key to continue~"

        SHIP File1,Ultimate
```

```
      SHIP File1,Yield
      SHIP File1,Rigidity
      SHIP File1,Enduran
      SHIP File1,Load_max
      SHIP File1,Load_min
      SHIP File1,Fatigue_life
      SHIP File1,Defln
      SHIP File1,Wire_dia
      SHIP File1,Factor

   CALL Cycle2,""

      RECEIVE File2,mean_rad
      RECEIVE File2,turns
      RECEIVE File2,length
      RECEIVE File2,solidht
      RECEIVE File2,disp
      RECEIVE File2,freeht
      RECEIVE File2,index
      RECEIVE File2,wahl
      RECEIVE File2,safety
      RECEIVE File2,critload
      RECEIVE File2,reco

   FIND Load
   FIND Total

   CLS
   COLOR = 15
   DISPLAY "
                  Design of Helical Springs


      The design values for the specified spring are the following:

      Mean Coil Radius              =  {mean_rad}
      No. of active turns           =  {turns}
      Total # of turns              =  {Total}
      Solid height                  =  {solidht}
      Displacement from zero to max =  {disp}
      Effective length              =  {length}
      Free height                   =  {freeht}
      Spring Index                  =  {index}
      Wahl factor                   =  {wahl}
      Factor of Safety              =  {safety}
"
   COLOR = 11
   DISPLAY "

           Press any key to continue with the design ~"

   CLS
   COLOR = 14
   FIND Val
```

```
   WCLOSE 1
;
!========================= RULES BLOCK =======================
Rule Change1
       IF Change_val = 1
       THEN Val = Change1
          CHAIN Ftmatrl2;

Rule Change2
       IF Change_val = 2
       THEN Val = Change2
          CHAIN Fatigue;

Rule Change3
       IF Change_val = 3
       THEN Val = Change3
          CLS
          COLOR = 27
          DISPLAY "
                        THANK YOU!!!
"
          COLOR = 14
          DISPLAY "

                     for using the KBS for the
"
          COLOR = 31
          DISPLAY "
                     DESIGN OF HELICAL SPRINGS"

          COLOR = 11
          DISPLAY "

                    Developed by: GIRIDHAR RAJA
          ME Student, New Jersey Institute of Technology,NJ

                  Under the guidance of Dr. Nouri Levy
         Associate Professor, Department of Mechanical Engineering
              New Jersey Institute of Technology, NJ
~";

Rule Ends1
   IF Ends = Square_and_ground_ends
   THEN Total = (turns + 2);

Rule Ends2
   IF Ends = Square_or_closed_ends
   THEN Total = (turns + 1.5);

Rule Ends3
   IF Ends = Plain_ends_not_ground
   THEN Total = (turns + 0.5);
```

```
Rule Ends4
    IF Ends = Plain_ends_ground
    THEN Total = (turns + 1);

Rule Ends5
    IF Ends = Two_&_half_turns
    THEN Total = (turns + 5);

Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.5)
    ELSE Total = (turns + 0.25);

Rule Load1
    IF Type = Compression AND
      critload < (Load_max) AND
      safety > 1.25
    THEN Load = Poor
       CLS
       COLOR = 11
       DISPLAY "

                    Design is very unsafe!!"
       COLOR = 15
       DISPLAY "

       For {Material},

              Critical Load      =        {critload}

              Maximum Load       =        {Load_max}

              Factor of Safety   =        {safety}

          The Critical load is lesser than the Maximum load

        acting on the spring. This causes buckling of the spring.

         Please increase the the Mean Coil Radius or try with a

                      different material."

       COLOR = 10
       DISPLAY "

                   Press any key to continue~";

Rule Load2
    IF Type = Compression AND
      critload < (Load_max) AND
      safety < 1.25
    THEN Load = Poor
       CLS
```

```
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

              Critical Load      =        {critload}

              Maximum Load       =        {Load_max}

              Factor of safety   =        {safety}

         The Critical Load is lesser than the Maximum Load acting

           on the spring. This causes buckling of the spring.

            Please increase the Mean Coil Radius or try with

                   a different material"

        COLOR = 10
        DISPLAY "
                   Press any key to continue~";

Rule Load3
   IF Type = Compression AND
     critload > (Load_max) AND
     safety < 1.25
   THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Calculated factor of safety      =       {safety}

            The Factor of Safety is too low which makes the

                   design very unsafe.

         Please choose a material with better mechanical properties

            like Young's Modulus & Ultimate Shear Strength
"
        COLOR = 10
        DISPLAY "
                   Press any key to continue~";
```

```
Rule Load4
    IF Type = Compression AND
      critload > (Load_max) AND
      safety > 1.0
    THEN Load = Good
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very safe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

            Calculated factor of safety  =   {safety}

            Critical Load              =  {critload}

            Maximum Load                =   {Load_max}


          The Critical Load is greater than the Maximum Load

           acting on the spring and the Factor of Safety

                    is also fairly high."
          COLOR = 10
          DISPLAY "
              Press any key to view the results ~";

Rule Load4
    IF Type = Extension AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "



                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "




             The calculated factor of safety is very low

    Please choose a different material with better mechanical properties.



              Press any key to still view the results~";
```

```
Rule Load6
   IF Type = Extension AND
     safety > 1.25
   THEN Load = Good
      CLS
      COLOR = 11
      DISPLAY "



                  Design is very safe!!"
      COLOR = 15
      DISPLAY "





             The calculated factor of safety is OK.


             Press any key to view the results.~";

!====================== QUESTIONS BLOCK ====================

ASK Change_val: "Do you wish to change any one of the following?

             1 = Material

             2 = User values

             3 = None
";
CHOICES Change_val: 1,  2,  3;
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Cycle3.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!======================== ACTIONS BLOCK ========================
ACTIONS
        LOADFACTS Fatigue
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Fatigue3

        COLOR = 15
        DISPLAY "
                    The user defined values are:

     Spring Type        =       {Type}
     Material           =       {Material}
     Load Type          =       {Type_load}
     Maximum Load       =       {Load_max}
     Minimum Load       =       {Load_min}
     No. of cycles      =       {Fatigue_life}
     Deflection         =       {Defln}
     Wire Diameter      =       {Wire_dia}
     Mean Coil Radius   =       {Mean_rad}
     Factor of safety   =       {Factor}
     Ends               =       {Ends}"

        COLOR = 11
        DISPLAY "

                    Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
     CLS
     COLOR = 14
        DISPLAY "
                        MATERIAL PROPERTIES

                Material:                  {Materials}
                Ultimate Shear Stress:     {Ultimate}
                Yield Stress:              {Yield}
                Rigidity Modulus:          {Rigidity}
                Endurance limit:           {Enduran}
"
     COLOR = 10
     DISPLAY "
                    Press any key to continue~"
```

```
    SHIP File1.Ultimate
    SHIP File1,Yield
    SHIP File1,Rigidity
    SHIP File1,Enduran
    SHIP File1,Load_max
    SHIP File1,Load_min
    SHIP File1,Fatigue_life
    SHIP File1,Defln
    SHIP File1,Wire_dia
    SHIP File1,Mean_rad
    SHIP File1,Factor

 CALL Cycle3,""

   RECEIVE File2,turns
   RECEIVE File2,length
   RECEIVE File2,solidht
   RECEIVE File2,disp
   RECEIVE File2,freeht
   RECEIVE File2,index
   RECEIVE File2,wahl
   RECEIVE File2,safety
   RECEIVE File2,critload
   RECEIVE File2,reco

 FIND Load
 FIND Total

 CLS
 COLOR = 15
 DISPLAY "
                Design of Helical Springs


   The design values for the specified spring are the following:

    No. of active turns            =   {turns}
    Total # of turns               =   {Total}
    Solid height                   =   {solidht}
    Displacement from zero to max  =   {disp}
    Effective length               =   {length}
    Free height                    =   {freeht}
    Spring Index                   =   {index}
    Wahl factor                    =   {wahl}
    Factor of Safety               =   {safety}
"
 COLOR = 11
 DISPLAY "

          Press any key to continue with the design ~"

 CLS
 COLOR = 14
 FIND Val
```

```
   WCLOSE 1
;
!======================= RULES BLOCK ======================
Rule Change1
        IF Change_val = 1
        THEN Val = Change1
            CHAIN Ftmatrl3;

Rule Change2
        IF Change_val = 2
        THEN Val = Change2
            CHAIN Fatigue;

Rule Change3
        IF Change_val = 3
        THEN Val = Change3
            CLS
            COLOR = 27
            DISPLAY "
                           THANK YOU!!!
"
            COLOR = 14
            DISPLAY "

                      for using the KBS for the
"
            COLOR = 31
            DISPLAY "
                      DESIGN OF HELICAL SPRINGS"

            COLOR = 11
            DISPLAY "

                      Developed by: GIRIDHAR RAJA
            ME Student, New Jersey Institute of Technology,NJ

                    Under the guidance of Dr. Nouri Levy
          Associate Professor, Department of Mechanical Engineering
                 New Jersey Institute of Technology, NJ.
~";

Rule Ends1
    IF Ends = Square_and_ground_ends
    THEN Total = (turns + 2);

Rule Ends2
    IF Ends = Square_or_closed_ends
    THEN Total = (turns + 1.5);

Rule Ends3
    IF Ends = Plain_ends_not_ground
    THEN Total = (turns + 0.5);

Rule Ends4
```

```
    IF Ends = Plain_ends_ground
    THEN Total = (turns + 1);

Rule Ends5
    IF Ends = Two_&_half_turns
    THEN Total = (turns + 5);

Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.5)
    ELSE Total = (turns + 0.25);

Rule Load1
    IF Type = Compression AND
      critload < (Load_max) AND
      safety > 1.25
    THEN Load = Poor
       CLS
       COLOR = 11
       DISPLAY "

                    Design is very unsafe!!"
       COLOR = 15
       DISPLAY "

       For {Material},

              Critical Load      =       {critload}

              Maximum Load       =       {Load_max}

              Factor of Safety   =       {safety}

          The Critical load is lesser than the Maximum load

        acting on the spring. This causes buckling of the spring.

         Please increase the the Mean Coil Radius or try with a

                     different material "

       COLOR = 10
       DISPLAY "

                   Press any key to continue--".

Rule Load2
    IF Type = Compression AND
      critload < (Load_max) AND
      safety < 1.25
    THEN Load = Poor
       CLS
       COLOR = 11
```

```
        DISPLAY "
                  Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

                  Critical Load      =        {critload}

                  Maximum Load       =        {Load_max}

                  Factor of safety   =        {safety}

         The Critical Load is lesser than the Maximum Load acting

             on the spring. This causes buckling of the spring.

              Please increase the Mean Coil Radius or try with

                          a different material"

        COLOR = 10
        DISPLAY "
                      Press any key to continue~";

Rule Load3
    IF Type = Compression AND
      critload > (Load_max) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                   Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Calculated factor of safety          =        {safety}

              The Factor of Safety is too low which makes the

                          design very unsafe

         Please choose a material with better mechanical properties

              like Young's Modulus & Ultimate Shear Strength
"
        COLOR = 10
        DISPLAY "
                      Press any key to continue~";

Rule Load4
```

```
IF Type = Compression AND
  critload > (Load_max) AND
  safety > 1.25
THEN Load = Good
   CLS
   COLOR = 11
   DISPLAY "
               Design is very safe!!"
   COLOR = 15
   DISPLAY "

   For {Material},

        Calculated factor of safety  =   {safety}

        Critical Load             =  {critload}

        Maximum Load                =   {Loadmax}


      The Critical Load is greater than the Maximum Load

       acting on the spring and the Factor of Safety

               is also fairly high."
      COLOR = 10
      DISPLAY "
           Press any key to view the results.~";

Rule Load5
   IF Type = Extension AND
     safety < 1.25
   THEN Load = Poor
      CLS
      COLOR = 11
      DISPLAY "



               Design is very unsafe!!"
      COLOR = 15
      DISPLAY "



        The calculated factor of safety is very low

   Please choose a different material with better mechanical properties.



        Press any key to still view the results.~";
```

```
Rule Load6
   IF Type = Extension AND
     safety >= 1.25
   THEN Load = Good
      CLS
      COLOR = 11
      DISPLAY "



                   Design is very safe!!





               Press any key to view the results.~";

!====================== QUESTIONS BLOCK ====================

ASK Change_val: "Do you wish to change any one of the following?

              1 = Material

              2 = User values

              3 = None
";
CHOICES Change_val: 1,  2,  3;
```

```
!KBS to change Material
!Saved as Ftmatrl1.KBS

   EXECUTE;
   BKCOLOR = 0;
   RUNTIME;
   !ENDOFF;

!======================== ACTIONS BLOCK =====================
ACTIONS

   BCALL Delete3," "
   COLOR = 14
   FIND Material
   SAVEFACTS Design
   CHAIN Cycle1
;
!========================== RULES BLOCK ==========================
Rule 1
   IF Mat_choice = 1
   THEN Material = Stainless_steel;

Rule 2
   IF Mat_choice = 2
   THEN Material = Monel;

Rule 3
   IF Mat_choice = 3
   THEN Material = Inconel;

Rule 4
   IF Mat_choice = 4
   THEN Material = Inconel_X;

Rule 5
   IF Mat_choice = 5
   THEN Material = Music_wire;

Rule 6
   IF Mat_choice = 6
   THEN Material = Carbon_steel;

Rule 7
   IF Mat_choice = 7
   THEN Material = Chrome_silicon_alloy_steel;

Rule 8
   IF Mat_choice = 8
   THEN Material = Flat_spring_steel;

Rule 9
   IF Mat_choice = 9
   THEN Material = Clock_spring_steel;
```

```
Rule 10
    IF Mat_choice = 10
    THEN Material = Spring_brass;

Rule 11
    IF Mat_choice = 11
    THEN Material = Beryllium_copper;

Rule 12
    IF Mat_choice = 12
    THEN Material = Phosphor_bronze;

Rule 13
    IF Mat_choice = 13
    THEN Material = Hard_drawn_wire;

Rule 14
    IF Mat_choice = 14
    THEN Material = Oil_tempered_wire;

!====================== QUESTIONS BLOCK =====================

ASK Mat_choice: "What material do you wish to use?

    The materials in the database are:

    1. Stainless_steel              2. Monel
    3. Inconel                      4. Inconel_X
    5. Music_wire                   6. Carbon_steel
    7. Chrome_silicon_alloy_steel   8. Flat_spring_steel
    9. Clock_spring_steel           10. Spring_brass
    11. Beryllium_copper            12. Phosphor_bronze
    13. Hard_drawn_wire             14. Oil_tempered_wire

";
CHOICES Mat_choice: 1,2,3,4,5,6,7,8,9,10,11,12,13,14;
```

```
!MAIN KBS TO DETERMINE THE CONSTRAINTS FROM THE USER AND CHAIN
!THE VARIOUS KNOWLEDGE BASES ACCORDINGLY
!Saved as Spring.KBS
!       EXECUTE;
        RUNTIME;
        BKCOLOR = 3;
!       ENDOFF;

!*********************** ACTIONS BLOCK ************************
ACTIONS

   COLOR = 21
   DISPLAY "
                        NOTE!!!!!!
"
   COLOR = 15
   DISPLAY "
      Please choose one of the following constraints/known values
        to proceed with the design. According to the choice, the
        corresponding values are to be fed through the keyboard.

                  Press any key to continue~"

   CLS
   COLOR = 5
   FIND Type
   FIND Constraint
   CLS
   FIND Link
;

!************************** RULES BLOCK ***********************
Rule 1
   IF Constraint = 1
   THEN Link = Spring1
      FIND Mean_rad
      SAVEFACTS Sp1
      CHAIN Spring1;

Rule 2
   IF Constraint = 2
   THEN Link = Spring2
      FIND Wire_dia
      SAVEFACTS Sp2
      CHAIN Spring2;

Rule 3
   IF Constraint = 3
   THEN Link = Spring3
      FIND Mean_rad
      FIND Free_ht
      SAVEFACTS Sp3
      CHAIN Spring3;
```

```
Rule 4
   IF Constraint = 4
   THEN Link = Spring4
      FIND Wire_dia
      FIND Mean_rad
      SAVEFACTS Sp4
      CHAIN Spring4;

Rule 5
   IF Constraint = 5
   THEN Link = Spring5
      FIND Wire_dia
      FIND Free_ht
      SAVEFACTS Sp5
      CHAIN Spring5;

Rule 6
   IF Constraint = 6
   THEN Link = Spring6
      FIND Mean_rad
      FIND Wire_dia
      FIND Free_ht
      SAVEFACTS Sp6
      CHAIN Spring6;

!********************** QUESTIONS BLOCK ***********************
ASK Type: "What type of helical spring do you require?";
CHOICES Type: Compression,   Extension;

ASK Mean_rad: "What is the value of the mean coil radius? ";

ASK Wire_dia: "What is the wire diameter of the spring? ";

ASK Free_ht: "What is the free height of the spring that is
required? ";

ASK Constraint: "Choose one of the following:
 1 = Load,deflection,material,safety factor & mean coil radius

 2 = Load,deflection,material,safety factor & wire diameter

 3 = Load,deflection,material,safety factor,mean coil radius & free height
    Note: Free Height may turn out to be redundant

 4 = Load,deflection,material,safety factor,wire diameter
      & mean coil radius

 5 = Load,deflection,material,safety factor,wire diameter & free height

 6 = Load,deflection,material,safety factor,mean coil radius
      wire diameter & free height
";
CHOICES Constraint: 1, 2, 3, 4, 5, 6;
```

```
! Spring Design - Session 1 with an improved user interface
! Saved as Spring1.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!==================== ACTIONS BLOCK =====================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                    INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

               Press any key to continue.~
    "
   CLS
   COLOR = 15
   CLS
   FIND Load_static
   CLS
   FIND Val_ecc
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!===================== RULES BLOCK ======================
Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Design1
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Matl1;

Rule 2
   If End_type = 1
   Then Ends = Square_and_ground_ends;
```

```
Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Load_type = 2
    Then Type_load = Static_load_at_Elevated_Temperature
        FIND Load_val;

Rule 8
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 9
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 10
    If Mat_choice = 2
    Then Material = Monel;

Rule 11
    If Mat_choice = 3
    Then Material = Inconel;

Rule 12
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 13
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 14
    If Mat_choice = 6
    Then Material = Carbon_steel;

Rule 15
    If Mat_choice = 7
```

```
    Then Material = Chrome_silicon_alloy_steel;

Rule 16
    If Mat_choice = 8
    Then Material = Flat_spring_steel;

Rule 17
    If Mat_choice = 9
    Then Material = Clock_spring_steel;

Rule 18
    If Mat_choice = 10
    Then Material = Spring_brass;

Rule 19
    If Mat_choice = 11
    Then Material = Beryllium_copper;

Rule 20
    If Mat_choice = 12
    Then Material = Phosphor_bronze;

Rule 21
    If Mat_choice = 13
    Then Material = Hard_drawn_wire;

Rule 22
    If Mat_choice = 14
    Then Material = Oil_tempered_wire;

Rule 23
    If Ecc_val = Yes
    Then Val_ecc = Yes
        FIND Eccen
    Else Val_ecc = No
        Eccen = (0);

Rule 24
    If End_type = 6
    Then Ends = Machine_half_loop;

Rule 25
    If End_type = 7
    Then Ends = Short_twist_loop;

Rule 26
    If End_type = 8
    Then Ends = Raised_hook;

Rule 27
    If End_type = 9
    Then Ends = Full_twisted_loop;

!====================== QUESTIONS BLOCK ======================
```

ASK Load_static: "What is the maximum load acting on the spring? ";

ASK Defln: "What is the maximum deflection that is required? ";

ASK Mat_sugg: "Do you have any recommendation for the spring material?

Note: If there are no suggestions, a material would be suggested according to the type of application environment

";

CHOICES Mat_sugg: Yes, No;

ASK Mat_choice: "What material do you wish to use?

The materials in the database are:

1. Stainless_steel 2. Monel
3. Inconel 4. Inconel_X
5. Music_wire 6. Carbon_steel
7. Chrome_silicon_alloy_steel
8. Flat_spring_steel 9. Clock_spring_steel
10. Spring_brass 11. Beryllium_copper
12. Phosphor_bronze 13. Hard_drawn_wire
14. Oil_tempered_wire

";

CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

1 = Square & ground or forged ends (Compression spring)
2 = Squared or closed ends not ground (Compression spring)
3 = Plain Ends (Compression spring)
4 = Plain Ends ground (Compression spring)
5 = 2 1/2 turns set up (Compression spring)
6 = Machine half loop - open (Extension spring)
7 = Short twisted loop (Extension spring)
8 = Raised hook (Exctension spring)
9 = Full twisted loop (Extension spring)

";

CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";

ASK Ecc_val: "Is there an eccentricity in the loading?";
CHOICES Ecc_val: No, Yes;

ASK Eccen: "What is the value of the eccentricity?";

```
! Spring Design - Session 1 with an improved user interface
! Saved as Spring2.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK =====================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

            Press any key to continue.~
   "
   CLS
   COLOR = 15
   CLS
   FIND Load_static
   CLS
   FIND Val_ecc
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!====================== RULES BLOCK ======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Design2
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Matl2;

Rule 2
   If End_type = 1
```

```
    Then Ends = Square_and_ground_ends;

Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Load_type = 2
    Then Type_load = Static_load_at_Elevated_Temperature
        FIND Load_val;

Rule 8
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 9
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 10
    If Mat_choice = 2
    Then Material = Monel;

Rule 11
    If Mat_choice = 3
    Then Material = Inconel;

Rule 12
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 13
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 14
    If Mat_choice = 6
    Then Material = Carbon_steel;
```

```
Rule 15
    If Mat_choice = 7
    Then Material = Chrome_silicon_alloy_steel;

Rule 16
    If Mat_choice = 8
    Then Material = Flat_spring_steel;

Rule 17
    If Mat_choice = 9
    Then Material = Clock_spring_steel;

Rule 18
    If Mat_choice = 10
    Then Material = Spring_brass;

Rule 19
    If Mat_choice = 11
    Then Material = Beryllium_copper;

Rule 20
    If Mat_choice = 12
    Then Material = Phosphor_bronze;

Rule 21
    If Mat_choice = 13
    Then Material = Hard_drawn_wire;

Rule 22
    If Mat_choice = 14
    Then Material = Oil_tempered_wire;

Rule 23
    If Ecc_val = Yes
    Then Val_ecc = Yes
        FIND Eccen
    Else Val_ecc = No
        Eccen = (0);

Rule 24
    If End_type = 6
    Then Ends = Machine_half_loop;

Rule 25
    If End_type = 7
    Then Ends = Short_twisted_loop;

Rule 26
    If End_type = 8
    Then Ends = Raised_hook;

Rule 27
    If End_type = 9
    Then Ends = Full_twisted_loop;
```

```
!====================== QUESTIONS BLOCK ======================

ASK Load_static: "What is the maximum load acting on the spring? ";

ASK Defln: "What is the maximum deflection that is required? ";

ASK Mat_sugg: "Do you have any recommendation for the spring material?

          Note: If there are no suggestions, a material would be
             suggested according to the type of application
             environment
";
CHOICES Mat_sugg: Yes,  No;

ASK Mat_choice: "What material do you wish to use?

      The materials in the database are:

      1. Stainless_steel          2. Monel
      3. Inconel                  4. Inconel_X
      5. Music_wire               6. Carbon_steel
      7. Chrome_silicon_alloy_steel
      8. Flat_spring_steel        9. Clock_spring_steel
      10. Spring_brass            11. Beryllium_copper
      12. Phosphor_bronze         13. Hard_drawn_wire
      14. Oil_tempered_wire

";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

          1 = Square & ground or forged ends   (Compression spring)
          2 = Squared or closed ends not ground (Compression spring)
          3 = Plain Ends                 (Compression spring)
          4 = Plain Ends ground          (Compression spring)
          5 = 2 1/2 turns set up         (Compression spring)
          6 = Machine half loop          (Extension springs)
          7 = Short twisted loop         (Extension springs)
          8 = Raised hook                (Extension springs)
          9 = Full twisted loop          (Extension springs)
";
 CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";

ASK Ecc_val: "Is there an eccentricity in the loading?";
CHOICES Ecc_val: No,   Yes;

ASK Eccen: "What is the value of eccentricity?";
```

```
! Spring Design - Session 1 with an improved user interface
! Saved as Spring3.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK ======================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                    INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

                Press any key to continue.~
      "

   CLS
   COLOR = 15
   CLS
   FIND Load_static
   CLS
   FIND Val_ecc
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!======================= RULES BLOCK ======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Design3
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Matl3;
```

Rule 2
    If End_type = 1
    Then Ends = Square_and_ground_ends;

Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Load_type = 2
    Then Type_load = Static_load_at_Elevated_Temperature
        FIND Load_val;

Rule 8
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 9
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 10
    If Mat_choice = 2
    Then Material = Monel;

Rule 11
    If Mat_choice = 3
    Then Material = Inconel;

Rule 12
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 13
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 14
    If Mat_choice = 6

```
   Then Material = Carbon_steel;

Rule 15
   If Mat_choice = 7
   Then Material = Chrome_silicon_alloy_steel;

Rule 16
   If Mat_choice = 8
   Then Material = Flat_spring_steel;

Rule 17
   If Mat_choice = 9
   Then Material = Clock_spring_steel;

Rule 18
   If Mat_choice = 10
   Then Material = Spring_brass;

Rule 19
   If Mat_choice = 11
   Then Material = Beryllium_copper;

Rule 20
   If Mat_choice = 12
   Then Material = Phosphor_bronze;

Rule 21
   If Mat_choice = 13
   Then Material = Hard_drawn_wire;

Rule 22
   If Mat_choice = 14
   Then Material = Oil_tempered_wire;

Rule 23
   If Ecc_val = Yes
   Then Val_ecc = Yes
      FIND Eccen
   Else Val_ecc = No
      Eccen = (0);

Rule 24
   If End_type = 6
   Then Ends = Machine_half_loop;

Rule 25
   If End_type = 7
   Then Ends = Short_twisted_loop;

Rule 26
   If End_type = 8
   then Ends = Raised_hook;

Rule 27
```

```
   If End_type = 9
   Then Ends = Full_twisted_loop;

!====================== QUESTIONS BLOCK =======================

ASK Load_static: "What is the maximum load acting on the spring? ";

ASK Defln: "What is the maximum deflection that is required? ";

ASK Mat_sugg: "Do you have any recommendation for the spring material?

            Note: If there are no suggestions, a material would be
                suggested according to the type of application
                environment
";
CHOICES Mat_sugg: Yes,   No;

ASK Mat_choice: "What material do you wish to use?

        The materials in the database are:
        1. Stainless_steel          2. Monel
        3. Inconel                  4. Inconel_X
        5. Music_wire               6. Carbon_steel
        7. Chrome_silicon_alloy_steel
        8. Flat_spring_steel        9. Clock_spring_steel
        10. Spring_brass            11. Beryllium_copper
        12. Phosphor_bronze         13. Hard_drawn_wire
        14. Oil_tempered_wire

";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

            1 = Square & ground or forged ends   (Compression springs)
            2 = Squared or closed ends not ground (Compression springs)
            3 = Plain Ends                       (Compression springs)
            4 = Plain Ends ground                (Compression springs)
            5 = 2 1/2 turns set up               (Compression springs)
            6 = Machine half loop - open         (Extension springs)
            7 = Short twisted loop               (Extension springs)
            8 = Raised hook                      (Extension springs)
            9 = Full twisted loop                (Extension springs)
";
 CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?",

ASK Ecc_val: "Is there an eccentricity in the loading?";
CHOICES Ecc_val: Yes, No;
ASK Eccen: "What is the value of eccentricity?";
```

```
! Spring Design - Session 1 with an improved user interface
! Saved as Spring4.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK ======================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

              Press any key to continue.~
      "
   CLS
   COLOR = 15
   CLS
   FIND Load_static
   CLS
   FIND Val_ecc
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!======================= RULES BLOCK ======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Design4
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Matl4;

Rule 2
   If End_type = 1
```

```
    Then Ends = Square_and_ground_ends;

Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Load_type = 2
    Then Type_load = Static_load_at_Elevated_Temperature
        FIND Load_val;

Rule 8
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 9
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 10
    If Mat_choice = 2
    Then Material = Monel;

Rule 11
    If Mat_choice = 3
    Then Material = Inconel;

Rule 12
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 13
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 14
    If Mat_choice = 6
    Then Material = Carbon_steel;
```

```
Rule 15
    If Mat_choice = 7
    Then Material = Chrome_silicon_alloy_steel;

Rule 16
    If Mat_choice = 8
    Then Material = Flat_spring_steel;

Rule 17
    If Mat_choice = 9
    Then Material = Clock_spring_steel;

Rule 18
    If Mat_choice = 10
    Then Material = Spring_brass;

Rule 19
    If Mat_choice = 11
    Then Material = Beryllium_copper;

Rule 20
    If Mat_choice = 12
    Then Material = Phosphor_bronze;

Rule 21
    If Mat_choice = 13
    Then Material = Hard_drawn_wire;

Rule 22
    If Mat_choice = 14
    Then Material = Oil_tempered_wire;

Rule 23
    If Ecc_val = Yes
    Then Val_ecc = Yes
        FIND Eccen
    Else Val_ecc = No
        Eccen = (0);

Rule 24
    If End_type = 6
    Then Ends = Machine_half_loop;

Rule 25
    If End_type = 7
    Then Ends = Short_twisted_loop;

Rule 26
    If End_type = 8
    Then Ends = Raised_hook;

Rule 27
    If End_type = 9
    Then Ends = Full_twisted_loop;
```

```
!====================== QUESTIONS BLOCK =======================

ASK Load_static: "What is the maximum load acting on the spring? ";

ASK Defln: "What is the maximum deflection that is required? ";

ASK Mat_sugg: "Do you have any recommendation for the spring material?

            Note: If there are no suggestions, a material would be
                suggested according to the type of application
                environment
";
CHOICES Mat_sugg: Yes,   No;

ASK Mat_choice: "What material do you wish to use?

        The materials in the database are:
        1. Stainless_steel          2. Monel
        3. Inconel                  4. Inconel_X
        5. Music_wire               6. Carbon_steel
        7. Chrome_silicon_alloy_steel
        8. Flat_spring_steel        9. Clock_spring_steel
        10. Spring_brass            11. Beryllium_copper
        12. Phosphor_bronze         13. Hard_drawn_wire
        14. Oil_tempered_wire
";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

            1 = Square & ground or forged ends   (Compression springs)
            2 = Squared or closed ends not ground (Compression springs)
            3 = Plain Ends                  (Compression springs)
            4 = Plain Ends ground           (Compression springs)
            5 = 2 1/2 turns set up          (Compression springs)
            6 = Machine half loop - open    (Extension springs)
            7 = Short twisted loop          (Extension springs)
            8 = Raised hook                 (Extension springs)
            9 = Full twisted loop           (Extension springs)
";
CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?

        Note: If the choice is No, the default value is 2.";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";

ASK Ecc_val: "Is there an eccentricity in the loading?".
CHOICES Ecc_val: No,    Yes;

ASK Eccen: "What is the value of eccentricity?";
```

```
! Spring Design - Session 1 with an improved user interface
! Saved as Spring5.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK =====================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                    INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

                Press any key to continue.~
      "
   CLS
   COLOR = 15
   CLS
   FIND Load_static
   CLS
   FIND Val_ecc
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!======================= RULES BLOCK =======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Design5
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Matl5;

Rule 2
   If End_type = 1
```

```
    Then Ends = Square_and_ground_ends;

Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Load_type = 2
    Then Type_load = Static_load_at_Elevated_Temperature
        FIND Load_val;

Rule 8
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 9
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 10
    If Mat_choice = 2
    Then Material = Monel;

Rule 11
    If Mat_choice = 3
    Then Material = Inconel;

Rule 12
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 13
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 14
    If Mat_choice = 6
    Then Material = Carbon_steel;
```

```
Rule 15
   If Mat_choice = 7
   Then Material = Chrome_silicon_alloy_steel;

Rule 16
   If Mat_choice = 8
   Then Material = Flat_spring_steel;

Rule 17
   If Mat_choice = 9
   Then Material = Clock_spring_steel;

Rule 18
   If Mat_choice = 10
   Then Material = Spring_brass;

Rule 19
   If Mat_choice = 11
   Then Material = Beryllium_copper;

Rule 20
   If Mat_choice = 12
   Then Material = Phosphor_bronze;

Rule 21
   If Mat_choice = 13
   Then Material = Hard_drawn_wire;

Rule 22
   If Mat_choice = 14
   Then Material = Oil_tempered_wire;

Rule 23
   If Ecc_val = Yes
   Then Val_ecc = Yes
      FIND Eccen
   Else Val_ecc = No
      Eccen = (0);

Rule 24
   If End_type = 6
   Then Ends = Machine_half_loop;

Rule 25
   If End_type = 7
   Then Ends = Short_twisted_loop;

Rule 26
   If End_type = 8
   Then Ends = Raised_hook;

Rule 27
   If End_type = 9
   Then Ends = Full_twisted_loop;
```

```
!====================== QUESTIONS BLOCK =======================

ASK Load_static: "What is the maximum load acting on the spring? ";

ASK Defln: "What is the maximum deflection that is required? ";

ASK Mat_sugg: "Do you have any recommendation for the spring material?

            Note: If there are no suggestions, a material would be
                suggested according to the type of application
                environment
";
CHOICES Mat_sugg: Yes,   No;

ASK Mat_choice: "What material do you wish to use?

        The materials in the database are:

        1. Stainless_steel          2. Monel
        3. Inconel                  4. Inconel_X
        5. Music_wire               6. Carbon_steel
        7. Chrome_silicon_alloy_steel
        8. Flat_spring_steel        9. Clock_spring_steel
        10. Spring_brass            11. Beryllium_copper
        12. Phosphor_bronze         13. Hard_drawn_wire
        14. Oil_tempered_wire

";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

            1 = Square & ground or forged ends    (Compression springs)
            2 = Squared or closed ends not ground (Compression springs)
            3 = Plain Ends                  (Compression springs)
            4 = Plain Ends ground           (Compression springs)
            5 = 2 1/2 turns set up          (Compression springs)
            6 = Machine half loop           (Extension springs)
            7 = Short twisted loop          (Extension springs)
            8 = Raised hook                 (Extension springs)
            9 = Full twisted loop           (Extension springs)
";
 CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";

ASK Ecc_val: "Is there an eccentricity in the loading?";
CHOICES Ecc_val: No,    Yes;

ASK Eccen: "What is the value of eccentricity?";
```

```
! Spring Design - Session 1 with an improved user interface
! Saved as Spring6.KBS
   EXECUTE;
   BKCOLOR = 6;
   RUNTIME;
!  ENDOFF;

!===================== ACTIONS BLOCK =======================
ACTIONS
   BCALL Reset," "
   CLS
   COLOR = 20
   DISPLAY "

                    INSTRUCTIONS"

   COLOR = 15
   DISPLAY "
   Use the arrow keys to move the light bar to a desired answer
   choice and then press the ENTER key. For questions without
   answer choices, enter the required value.

               Press any key to continue.~
     "
   CLS
   COLOR = 15
   CLS
   FIND Load_static
   CLS
   FIND Val_ecc
   CLS
   FIND Defln
   CLS
   FIND Ends
   CLS
   FIND Safety
   CLS
   FIND Sugg_mat
   CLS
;
!======================= RULES BLOCK =======================

Rule 1
   If Mat_sugg = Yes
   Then Sugg_mat = Dummy
      FIND Material
      SAVEFACTS Desn
      CHAIN Design6
   Else Sugg_mat = Dummy
      SAVEFACTS Matl
      CHAIN Matl6;

Rule 2
   If End_type = 1
```

```
    Then Ends = Square_and_ground_ends;

Rule 3
    If End_type = 2
    Then Ends = Square_or_closed_ends;

Rule 4
    If End_type = 3
    Then Ends = Plain_ends;

Rule 5
    If End_type = 4
    Then Ends = Plain_ends_not_ground;

Rule 6
    If End_type = 5
    Then Ends = Two_&_half_turns;

Rule 7
    If Load_type = 2
    Then Type_load = Static_load_at_Elevated_Temperature
        FIND Load_val;

Rule 8
    If Fac_safe = Yes
    Then Safety = Yes
        FIND Factor
    Else Safety = No
        Factor = (2);

Rule 9
    If Mat_choice = 1
    Then Material = Stainless_steel;

Rule 10
    If Mat_choice = 2
    Then Material = Monel;

Rule 11
    If Mat_choice = 3
    Then Material = Inconel;

Rule 12
    If Mat_choice = 4
    Then Material = Inconel_X;

Rule 13
    If Mat_choice = 5
    Then Material = Music_wire;

Rule 14
    If Mat_choice = 6
    Then Material = Carbon_steel;
```

```
Rule 15
    If Mat_choice = 7
    Then Material = Chrome_silicon_alloy_steel;

Rule 16
    If Mat_choice = 8
    Then Material = Flat_spring_steel;

Rule 17
    If Mat_choice = 9
    Then Material = Clock_spring_steel;

Rule 18
    If Mat_choice = 10
    Then Material = Spring_brass;

Rule 19
    If Mat_choice = 11
    Then Material = Beryllium_copper;

Rule 20
    If Mat_choice = 12
    Then Material = Phosphor_bronze;

Rule 21
    If Mat_choice = 13
    Then Material = Hard_drawn_wire;

Rule 22
    If Mat_choice = 14
    Then Material = Oil_tempered_wire;

Rule 23
    If Ecc_val = Yes
    Then Val_ecc = Yes
        FIND Eccen
    Else Val_ecc = No
        Eccen = (0);

Rule 24
    If End_type = 6
    Then Ends = Machine_half_loop;

Rule 25
    If End_type = 7
    Then Ends = Short_twisted_loop;

Rule 26
    If End_type = 8
    Then Ends = Raised_hook;

Rule 27
    If End_type = 9
    Then Ends = Full_twisted_loop;
```

```
!====================== QUESTIONS BLOCK ========================

ASK Load_static: "What is the maximum load acting on the spring? ";

ASK Defln: "What is the maximum deflection that is required? ";

ASK Mat_sugg: "Do you have any recommendation for the spring material?

            Note: If there are no suggestions, a material would be
                suggested according to the type of application
                environment
";
CHOICES Mat_sugg: Yes,   No;

ASK Mat_choice: "What material do you wish to use?

        The materials in the database are:

        1. Stainless_steel          2. Monel
        3. Inconel                  4. Inconel_X
        5. Music_wire               6. Carbon_steel
        7. Chrome_silicon_alloy_steel
        8. Flat_spring_steel        9. Clock_spring_steel
        10. Spring_brass            11. Beryllium_copper
        12. Phosphor_bronze         13. Hard_drawn_wire
        14. Oil_tempered_wire

";
CHOICES Mat_choice:1,2,3,4,5,6,7,8,9,10,11,12,13,14;

ASK End_type: "What type of ends do you require for the desired spring?

            1 = Square & ground or forged ends    (Compression springs)
            2 = Squared or closed ends not ground (Compression springs)
            3 = Plain Ends                        (Compression springs)
            4 = Plain Ends ground                 (Compression springs)
            5 = 2 1/2 turns set up                (Compression springs)
            6 = Machine half loop                 (Extension springs)
            7 = Short twisted loop                (Extension springs)
            8 = Raised hook                       (Extension springs)
            9 = Full twisted loop                 (Extension springs)
";
 CHOICES End_type: 1,2,3,4,5,6,7,8,9;

ASK Fac_safe: "Do you want a particular factor of safety?";
CHOICES Fac_safe: Yes, No;

ASK Factor: "What is the factor of safety value?";

ASK Ecc_val: "Is there an eccentricity in the loading?";
CHOICES Ecc_val: No,    Yes;

ASK Eccen: "What is the value of the eccentricity?";
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Design1.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!========================= ACTIONS BLOCK =======================
ACTIONS
        LOADFACTS Static
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Sp1

        COLOR = 15
        DISPLAY "
                The user defined values are:

     Spring Type        =       {Type}
     Material           =       {Material}
     Load Type          =       {Type_load}
     Load               =       {Load_static}
     Deflection         =       {Defln}
     Eccentricity       =       {Eccen}
     Mean Coil Radius   =       {Mean_rad}
     Factor of safety   =       {Factor}
     Ends               =       {Ends}"

        COLOR = 11
        DISPLAY "

                Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl,ALL
    CLS
    COLOR = 14
        DISPLAY "
                  MATERIAL PROPERTIES

            Material:                 {Materials}
            Ultimate Shear Stress:      {Ultimate}
            Yield Stress:             {Yield}
            Rigidity Modulus:           {Rigidity}"

    COLOR = 10
    DISPLAY "
                Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
        SHIP File1,Rigidity
```

```
    SHIP File1,Load_static
    SHIP File1,Defln
    SHIP File1,Mean_rad
    SHIP File1,Eccen
    SHIP File1,Factor

  CALL Design1,""

    RECEIVE File2,wire_dia
    RECEIVE File2,turns
    RECEIVE File2,length
    RECEIVE File2,solidht
    RECEIVE File2,disp
    RECEIVE File2,freeht
    RECEIVE File2,index
    RECEIVE File2,wahl
    RECEIVE File2,safety
    RECEIVE File2,ratio_rad
    RECEIVE File2,crit_load
    RECEIVE File2,reco

  FIND Diameter
  FIND Radius
  FIND Load
  FIND Total
  CLS
  COLOR = 15
  DISPLAY "
                 Design of Helical Springs


    The design values for the specified spring are the following:

     Wire Diameter               =   {wire_dia}
     No. of active turns         =   {turns}
     Total # of turns            =   {Total}
     Solid height                =   {solidht}
     Displacement from zero to max    =   {disp}
     Effective length            =   {length}
     Free height                 =   {freeht}
     Spring Index                =   {index}
     Wahl factor                 =   {wahl}
     Factor of Safety            =   {safety}
"
  COLOR = 11
  DISPLAY "

          Press any key to continue with the design ~"

  CLS
  COLOR = 14
  FIND Val
  WCLOSE 1
```

```
!======================== RULES BLOCK =======================
Rule Change1
        IF Change_val = 1
        THEN Val = Change1
            CHAIN Matrl1;

Rule Change2
        IF Change_val = 2
        THEN Val = Change2
            CHAIN Spring;

Rule Change3
        IF Change_val = 3
        THEN Val = Change3
            CLS
            COLOR = 27
            DISPLAY "
                        THANK YOU!!!
"
            COLOR = 14
            DISPLAY "

                    for using the KBS for the
"
            COLOR = 31
            DISPLAY "
                    DESIGN OF HELICAL SPRINGS"

            COLOR = 11
            DISPLAY "

                    Developed by: GIRIDHAR RAJA
            ME Student, New Jersey Institute of Technology,NJ

                Under the guidance of Dr. Nouri Levy
          Associate Professor, Department of Mechanical Engineering
                New Jersey Institute of Technology, NJ.
~";

Rule Dia
    IF Material = (Materials) AND
      wire_dia > (Mindia) AND
      wire_dia < (Maxdia)
    THEN Diameter = Inrange
    ELSE Diameter = Outrange
        CLS
        COLOR = 11
        DISPLAY "

        For {Material},

  ===>      Minimum diameter = {Mindia}

  ===>      Maximum diameter = {Maxdia}
```

```
===>      Calculated wire diameter = {wire_dia}

          We see that the wire diameter does not lie within the

                    minimum and maximum values.

        Choose a different material with higher mechanical properties
"
        COLOR = 5
        DISPLAY "
                    Press any key to continue~"
        CHAIN Matrl1;

Rule Ends1
    IF Ends = Square_and_ground_ends
    THEN Total = (turns + 2);

Rule Ends2
    IF Ends = Square_or_closed_ends
    THEN Total = (turns + 1.5);

Rule Ends3
    IF Ends = Plain_ends_not_ground
    THEN Total = (turns + 0.5);

Rule Ends4
    IF Ends = Plain_ends_ground
    THEN Total = (turns + 1);

Rule Ends5
    IF Ends = Two_&_half_turns
    THEN Total = (turns + 5);

Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.25);

Rule Ends7
    IF Ends = Raised_hook OR
      Ends = Full_twisted_loop
    THEN Total = (turns + 0.5);

Rule Radius
    IF ratio_rad <= 1.05 AND
      ratio_rad >= 0.95
    THEN Radius = OK
    ELSE Radius = Not_OK
        CLS
        COLOR = 11
        DISPLAY "

                    RECOMMENDATION!!!
```

```
        For {Material},

          Due to standardization of the wire diameter, the value

        of Mean Coil Radius is greater than 10% of the required value.

                So, please choose a different material."

        COLOR = 15
        DISPLAY "

                    Press any key to continue~"

        CLS
        FIND Val;

Rule Load1
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety > 1.25
    THEN Load = Poor
        CLS
        COLOR = 15
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 11
        DISPLAY "

        For {Material},

            Critical Load       =       {crit_load}

            Maximum Load        =       {Load_static}

            The Critical Load is lesser than the Maximum Load.

                  This causes the spring to buckle.

        So, please increase the Mean Coil Radius or choose a material

                    with better Young's Modulus."

        COLOR = 10
        DISPLAY "
              Press any key to still view the results~";

Rule Load2
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 15
        DISPLAY "
```

```
                Design is very unsafe!!"
    COLOR = 11
    DISPLAY "

    For {Material},

    Critical Load              =       {crit_load}

    Maximum Load               =       {Load_static}

    Calculated Safety Factor   =       {safety}


      The Critical Load is lesser than the Maximum Load.

        The calculated factor of safety is also low.

   So, please increase the Mean Coil Radius or choose a different

          material with better Young's Modulus."

    COLOR = 10
    DISPLAY "
          Press any key to still view the results~";

Rule Load3
  IF Type = Compression AND
    crit_load > (Load_static) AND
    safety < 1.25
  THEN Load = Poor
    CLS
    COLOR = 11
    DISPLAY "
                Design is very unsafe!!"

    COLOR = 15
    DISPLAY "

    For {Material},

    Calculated factor of safety   =    {safety}

   The factor of safety is too low which makes the design unsafe.

  Please choose a different material with better mechanical properties."

    COLOR = 10
    DISPLAY "
          Press any key to still view the results~".

Rule Load4
  IF Compression = Type AND
    crit_load > (Load_static) AND
    safety > 1.25
```

```
    THEN Load = Good
       CLS
       COLOR = 11
       DISPLAY "
                    Design is safe!!"
       COLOR = 15
       DISPLAY "

       For {Material},

          The critical load is greater than the maximum load &

          the calculated factor of safety is also reasonable."

       COLOR = 10
       DISPLAY "
              Press any key to view the results.~";

Rule Load5
    IF Type = Extension AND
      safety > 1.25
    THEN Load = Good
       COLOR = 11
       DISPLAY "



                   Design is very safe!!"
       COLOR = 15
       DISPLAY "

            The calculated values for the design are OK


              Press any key to view the results~"
    ELSE Load = Poor
       COLOR = 11
       DISPLAY "
                   Design is very unsafe!!"
       COLOR = 15
       DISPLAY "

             The calculated factor of safety is low.

    Please choose a different material with better mechanical properties.



             Press any key to still view the results~".

!========================= QUESTIONS BLOCK =========================

ASK Change_val: "Do you wish to change any one of the following?
```

1 = Material

2 = User values

3 = None

";
CHOICES Change_val: 1,   2,   3;

```
! Spring Design - Module 3 - Design calculations
! Saved as Design2.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!       ENDOFF;

!========================= ACTIONS BLOCK ========================
ACTIONS
        LOADFACTS Static
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Sp2

        COLOR = 15
        DISPLAY "
                The user defined values are:

    Spring Type        =        {Type}
    Material           =        {Material}
    Load Type          =        {Type_load}
    Load               =        {Load_static}
    Deflection         =        {Defln}
    Eccentricity       =        {Eccen}
    Wire diameter      =        {Wire_dia}
    Factor of safety   =        {Factor}
    Ends               =        {Ends}"

        COLOR = 11
        DISPLAY "

                Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
    CLS
    COLOR = 14
        DISPLAY "
                    MATERIAL PROPERTIES

            Material:                  {Materials}
            Ultimate Shear Stress:     {Ultimate}
            Yield Stress:              {Yield}
            Rigidity Modulus:          {Rigidity}"

    COLOR = 10
    DISPLAY "
                Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
        SHIP File1,Rigidity
```

```
      SHIP File1,Load_static
      SHIP File1,Defln
      SHIP File1,Wire_dia
      SHIP File1,Eccen
      SHIP File1,Factor

   CALL Design2,""

      RECEIVE File2,mean_rad
      RECEIVE File2,turns
      RECEIVE File2,length
      RECEIVE File2,solidht
      RECEIVE File2,disp
      RECEIVE File2,freeht
      RECEIVE File2,index
      RECEIVE File2,wahl
      RECEIVE File2,safety
      RECEIVE File2,crit_load
      RECEIVE File2,reco

   FIND Load
   FIND Total
   CLS
   COLOR = 15
   DISPLAY "
                  Design of Helical Springs


      The design values for the specified spring are the following:

      Mean Coil Radius                = {mean_rad}
      No. of active turns             = {turns}
      Total # of turns                = {Total}
      Solid height                    = {solidht}
      Displacement from zero to max   = {disp}
      Effective length                = {length}
      Free height                     = {freeht}
      Spring Index                    = {index}
      Wahl factor                     = {wahl}
      Factor of Safety                = {safety}
"
   COLOR = 11
   DISPLAY "

            Press any key to continue with the design ~"

   CLS
   COLOR = 14
   FIND Val
   WCLOSE 1
;
!====================== RULES BLOCK ======================
Rule Change1
      IF Change_val = 1
```

```
      THEN Val = Change1
         CHAIN Matrl2;

Rule Change2
      IF Change_val = 2
      THEN Val = Change2
         CHAIN Spring;

Rule Change3
      IF Change_val = 3
      THEN Val = Change3
         CLS
         COLOR = 27
         DISPLAY "
                    THANK YOU!!!
"
         COLOR = 14
         DISPLAY "

                 for using the KBS for the
"
         COLOR = 31
         DISPLAY "
                 DESIGN OF HELICAL SPRINGS"

         COLOR = 11
         DISPLAY "

                Developed by: GIRIDHAR RAJA
         ME Student, New Jersey Institute of Technology,NJ

              Under the guidance of Dr. Nouri Levy
        Associate Professor, Department of Mechanical Engineering
             New Jersey Institute of Technology, NJ.
~";

Rule Ends1
   IF Ends = Square_and_ground_ends
   THEN Total = (turns + 2);

Rule Ends2
   IF Ends = Square_or_closed_ends
   THEN Total = (turns + 1.5);

Rule Ends3
   IF Ends = Plain_ends_not_ground
   THEN Total = (turns + 0.5);

Rule Ends4
   IF Ends = Plain_ends_ground
   THEN Total = (turns + 1);

Rule Ends5
   IF Ends = Two_&_half_turns
```

```
    THEN Total = (turns + 5);

Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.25)
    ELSE Total = (turns + 0.5);

Rule Load1
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety > 1.25
    THEN Load = Poor
        CLS
        COLOR = 15
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 11
        DISPLAY "

        For {Material},

        Critical Load       =       {crit_load}

        Maximum Load        =       {Load_static}

            The critical load is lesser than the maximum load.

              This causes the spring the spring to buckle.

       So, please increase the Mean Coil Radius or choose a material

                  with better Young's Modulus."
        COLOR = 10
        DISPLAY "
              Press any key to still view the results~";

Rule Load2
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 15
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 11
        DISPLAY "

        For {Material},

        Critical Load            =       {crit_load}

        Maximum Load             =       {Load_static}
```

```
    Calculated safety factor     =        {safety}

        The Critical load is lesser than the Maximum load.

          The calculated factor of safety is also low.

   So, please increase the Mean Coil Radius or choose a material

                with better Young's Modulus."
    COLOR = 10
    DISPLAY "
           Press any key to still view the results~";

Rule Load3
    IF Type = Compression AND
      crit_load > (Load_static) AND
      safety < 1.25
    THEN Load = Poor
       CLS
       COLOR = 15
       DISPLAY "
                    Design is very unsafe!!"
       COLOR = 11
       DISPLAY "

       For {Material},

           Calculated factor of safety   =    {safety}

      The factor of safety is too low which makes the design unsafe.

     So, please choose a material with better mechanical properties."
       COLOR = 10
       DISPLAY "
            Press any key to still view the results~";

Rule Load4
    IF Type = Compression AND
      crit_load > (Load_static) AND
      safety > 1.25
    THEN Load = Good
       CLS
       COLOR = 15
       DISPLAY "
                      Design is safe!!"
       COLOR = 11
       DISPLAY "

       For {Material},

          The critical load is greater than the maximum load &

          the calculated factor of safety is also reasonable."
       COLOR = 10
```

```
        DISPLAY "
                Press any key to view the results~";

Rule Load5
    IF Type = Extension AND
      safety > 1.25
    THEN Load = Good
        COLOR = 11
        DISPLAY "






                    Design is very safe!!"
        COLOR = 15
        DISPLAY "


                The calculations turn out to be OK"
    ELSE Load = Poor
        COLOR = 11
        DISPLAY "


                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "



              The factor of safety calculated is very low.

            Choose a material with better mechanical properties.




               Press any key to still view the results~".

!====================== QUESTIONS BLOCK =======================

ASK Change_val: "Do you wish to change any one of the following?

               1 = Material

               2 = User values

               3 = None
";
CHOICES Change_val: 1,  2,  3;
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Design3.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!====================== ACTIONS BLOCK ======================
ACTIONS
        LOADFACTS Static
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Sp3

        COLOR = 15
        DISPLAY "
                The user defined values are:

    Spring Type        =      {Type}
    Material          =      {Material}
    Load Type         =      {Type_load}
    Load              =      {Load_static}
    Eccentricity      =      {Eccen}
    Deflection        =      {Defln}
    Mean Coil Radius  =      {Mean_rad}
    Free Height       =      {Free_ht}
    Factor of safety  =      {Factor}
    Ends              =      {Ends}"

        COLOR = 11
        DISPLAY "

                Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
    CLS
    COLOR = 14
        DISPLAY "
                MATERIAL PROPERTIES

            Material:               {Materials}
            Ultimate Shear Stress:    {Ultimate}
            Yield Stress:           {Yield}
            Rigidity Modulus:         {Rigidity}"

    COLOR = 10
    DISPLAY "
                Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
```

```
    SHIP File1,Rigidity
    SHIP File1,Load_static
    SHIP File1,Defln
    SHIP File1,Mean_rad
    SHIP File1,Free_ht
    SHIP File1,Eccen
    SHIP File1,Factor

 CALL Design3,""

    RECEIVE File2,wire_dia
    RECEIVE File2,turns
    RECEIVE File2,length
    RECEIVE File2,solidht
    RECEIVE File2,freeht
    RECEIVE File2,disp
    RECEIVE File2,index
    RECEIVE File2,wahl
    RECEIVE File2,safety
    RECEIVE File2,ratio_rad
    RECEIVE File2,crit_load
    RECEIVE File2,reco

 Advance = (freeht/turns)

 FIND Diameter
 FIND Radius
 FIND Load
 FIND Total
 CLS
 COLOR = 15
 DISPLAY "
                 Design of Helical Springs


    The design values for the specified spring are the following:

     Wire Diameter                  =   {wire_dia}
     No. of active turns            =   {turns}
     Total # of turns               =   {Total}
     Solid height                   =   {solidht}
     Calculated Free Height         =   {freeht}
     Displacement from zero to max  =   {disp}
     Effective length               =   {length}
     Spring Index                   =   {index}
     Wahl factor                    =   {wahl}
     Factor of Safety               =   {safety}
     Advancement per turn           =   {Advance}
"
 COLOR = 11
 DISPLAY "

          Press any key to continue with the design ~"
```

```
   CLS
   COLOR = 14
   FIND Val
;
!========================= RULES BLOCK =======================
Rule Change1
       IF Change_val = 1
       THEN Val = Change1
           CHAIN Matrl3;

Rule Change2
       IF Change_val = 2
       THEN Val = Change2
           CHAIN Spring;

Rule Change3
       IF Change_val = 3
       THEN Val = Change3
           CLS
           COLOR = 27
           DISPLAY "
                         THANK YOU!!!
"
           COLOR = 14
           DISPLAY "

                    for using the KBS for the
"
           COLOR = 31
           DISPLAY "
                    DESIGN OF HELICAL SPRINGS"

           COLOR = 11
           DISPLAY "

                   Developed by: GIRIDHAR RAJA
           ME Student, New Jersey Institute of Technology,NJ

                 Under the guidance of Dr. Nouri Levy
          Associate Professor, Department of Mechanical Engineering
                New Jersey Institute of Technology, NJ.
~";

Rule Dia
   IF Material = (Materials) AND
     wire_dia > (Mindia) AND
     wire_dia < (Maxdia)
   THEN Diameter = Inrange
   ELSE Diameter = Outrange
      CLS
      COLOR = 11
      DISPLAY "

      For {Material},
```

```
===>      Minimum diameter = {Mindia}

===>      Maximum diameter = {Maxdia}

===>      Calculated wire diameter = {wire_dia}

          We see that the wire diameter does not lie within the
                minimum and maximum values.

        Choose a different material with higher mechanical properties
"
        COLOR = 5
        DISPLAY "
                  Press any key to continue~"
        CHAIN Matrl3;

Rule Ends1
   IF Ends = Square_and_ground_ends
   THEN Total = (turns + 2);

Rule Ends2
   IF Ends = Square_or_closed_ends
   THEN Total = (turns + 1.5);

Rule Ends3
   IF Ends = Plain_ends_not_ground
   THEN Total = (turns + 0.5);

Rule Ends4
   IF Ends = Plain_ends_ground
   THEN Total = (turns + 1);

Rule Ends5
   IF Ends = Two_&_half_turns
   THEN Total = (turns + 5);

Rule Ends6
   IF Ends = Machine_half_loop OR
     Ends = Short_twisted_loop
   THEN Total = (turns+0.5)
   ELSE Total = (turns+0.25);

Rule Radius
   IF ratio_rad <= 1.05 AND
     ratio_rad >= 0.95
   THEN Radius = OK
   ELSE Radius = Not_OK
      CLS
      COLOR = 11
      DISPLAY "

      For {Material},

        Due to standardization of the Wire Diameter, the value
```

```
          of the Mean Coil Radius is greater than 5% of the

                    required value.

           So, please change the material and try again."

       COLOR = 15
       DISPLAY "

                  Press any key to continue~";

Rule Load1
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety > 1.25
    THEN Load = Poor
       CLS
       COLOR = 15
       DISPLAY "
                    Design is very unsafe!!"
       COLOR = 11
       DISPLAY "

       For {Material},

           Critical Load      =       {crit_load}

           Maximum Load       =       {Load_static}

           The Critical load is lesser than the Maximum load.

                This causes the spring to buckle.

      So,please increase the Mean Coil Radius or choose a material

                 with better Young's Modulus."
       COLOR = 10
       DISPLAY "
               Press any key to view the results ~";

Rule Load2
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety < 1.25
    THEN Load = Poor
       CLS
       COLOR = 15
       DISPLAY "
                    Design is very unsafe!!"
       COLOR = 11
       DISPLAY "

       For {Material},
```

```
        Critical Load              =        {crit_load}

        Maximum Load                =        {Load_static}

        Calculated safety factor    =       {safety}

           The Critical load is lesser than the Maximum load.

             The calculated factor of safety is also low.

       So,please increase the Mean Coil Radius or choose a material

                    with better Young's Modulus."
        COLOR = 10
        DISPLAY "
                Press any key to view the results.~";

Rule Load3
    IF Type = Compression AND
      crit_load > (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 15
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 11
        DISPLAY "

        For {Material},

        Calculated safety factor    =       {safety}

     The factor of safety is too low which makes the design very unsafe.

       So, please choose a material with better mechanical properties."
        COLOR = 10
        DISPLAY "
                Press any key to view the results ~";

Rule Load4
    IF Type = Compression AND
      crit_load > (Load_static) AND
      safety > 1.25
    THEN Load = Good
        CLS
        COLOR = 15
        DISPLAY "
                    Design is safe!!"
        COLOR = 11
        DISPLAY "

        For {Material},
```

```
        The Critical load is greater than the Maximum load &

        the calculated factor of safety is also reasonable."
      COLOR = 10
      DISPLAY "
              Press any key to view the results.~";

Rule Load5
   IF Type = Extension AND
     safety > 1.25
   THEN Load = Good
      CLS
      COLOR = 11
      DISPLAY "




                 Design is very safe!!"
      COLOR = 15
      DISPLAY "

          The calculated factor of safety is reasonable.



              Press any key to view the results~";

Rule Load6
   IF Type = Extension AND
     safety < 1.25
   THEN Load = Poor
      CLS
      COLOR = 11
      DISPLAY "




                 Design is very unsafe!!"

      COLOR = 15
      DISPLAY "


          The calculated factor of safety is very low.

   Please choose a different material with better mechanical properties.





            Press any key to still view the results~";
```

```
!======================= QUESTIONS BLOCK ====================

ASK Change_val: "Do you wish to change any one of the following?

                1 = Material

                2 = User values

                3 = None
";
CHOICES Change_val: 1,   2,   3;
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Design4.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!======================== ACTIONS BLOCK ========================
ACTIONS
        LOADFACTS Static
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Sp4

        COLOR = 15
        DISPLAY "
                     The user defined values are:

    Spring Type          =        {Type}
    Material             =        {Material}
    Load Type            =        {Type_load}
    Load                 =        {Load_static}
    Deflection           =        {Defln}
    Eccentricity         =        {Eccen}
    Mean Coil Radius     =        {Mean_rad}
    Wire Diameter        =        {Wire_dia}
    Factor of safety     =        {Factor}
    Ends                 =        {Ends}"

        COLOR = 11
        DISPLAY "

                      Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
    CLS
    COLOR = 14
        DISPLAY "
                         MATERIAL PROPERTIES

                Material:                  {Materials}
                Ultimate Shear Stress:     {Ultimate}
                Yield Stress:              {Yield}
                Rigidity Modulus:          {Rigidity}"

    COLOR = 10
    DISPLAY "
                     Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
```

```
        SHIP File1,Rigidity
        SHIP File1,Load_static
        SHIP File1,Defln
        SHIP File1,Mean_rad
        SHIP File1,Wire_dia
        SHIP File1,Eccen
        SHIP File1,Factor

    CALL Design4,""

       RECEIVE File2,turns
       RECEIVE File2,length
       RECEIVE File2,solidht
       RECEIVE File2,disp
       RECEIVE File2,freeht
       RECEIVE File2,index
       RECEIVE File2,wahl
       RECEIVE File2,safety
       RECEIVE File2,crit_load
       RECEIVE File2,reco1

    FIND Load
    FIND Total
    CLS
    COLOR = 15
    DISPLAY "
                   Design of Helical Springs


       The design values for the specified spring are the following:

        No. of active turns           =   {turns}
        Total # of turns              =   {Total}
        Solid height                  =   {solidht}
        Displacement from zero to max    =   {disp}
        Effective length              =   {length}
        Free height                   =   {freeht}
        Spring Index                  =   {index}
        Wahl factor                   =   {wahl}
        Factor of Safety              =   {safety}
"
   COLOR = 11
    DISPLAY "

            Press any key to continue with the design ~"

    CLS
    COLOR = 14
    FIND Val
    WCLOSE 1
;
!======================== RULES BLOCK ======================
Rule Change1
        IF Change_val = 1
```

```
        THEN Val = Change1
           CHAIN Matrl4;

Rule Change2
        IF Change_val = 2
        THEN Val = Change2
           CHAIN Spring;

Rule Change3
        IF Change_val = 3
        THEN Val = Change3
           CLS
           COLOR = 27
           DISPLAY "
                       THANK YOU!!!
"
           COLOR = 14
           DISPLAY "

                   for using the KBS for the
"
           COLOR = 31
           DISPLAY "
                   DESIGN OF HELICAL SPRINGS"

           COLOR = 11
           DISPLAY "

                  Developed by: GIRIDHAR RAJA
           ME Student, New Jersey Institute of Technology,NJ

               Under the guidance of Dr. Nouri Levy
          Associate Professor, Department of Mechanical Engineering
                New Jersey Institute of Technology, NJ.
~";

Rule Ends1
    IF Ends = Square_and_ground_ends
    THEN Total = (turns + 2);

Rule Ends2
    IF Ends = Square_or_closed_ends
    THEN Total = (turns + 1.5);

Rule Ends3
    IF Ends = Plain_ends_not_ground
    THEN Total = (turns + 0.5);

Rule Ends4
    IF Ends = Plain_ends_ground
    THEN Total = (turns + 1);

Rule Ends5
    IF Ends = Two_&_half_turns
```

```
    THEN Total = (turns + 5);

Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.5)
    ELSE Total = (turns + 0.25);

Rule Load1
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety > 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

            Critical Load      =        {crit_load}

            Maximum Load       =        {Load_static}

            The Critical load is lesser than the Maximum load.

                  This causes the spring to buckle.

        So,please increase the Mean Coil Radius or choose a material

                    with better Young's Modulus."
        COLOR = 10
        DISPLAY "
                  Press any key to view the results~";

Rule Load2
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Critical Load              =        {crit_load}

        Maximum Load               =        {Load_static}
```

```
        Calculated safety factor    =       {safety}

          The Critical load is lesser than the Maximum load.

            The calculated factor of safety is also low.

        So,please increase the Mean Coil Radius or choose a material

                    with better Young's Modulus."
        COLOR = 10
        DISPLAY "
                  Press any key to view the results~";

Rule Load3
   IF Type = Compression AND
     crit_load > (Load_static) AND
     safety < 1.25
   THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                     Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Calculated safety factor    =       {safety}

   The calculated factor of safety is low which makes the design unsafe.

  So,please choose a different material with better mechanical properties."
        COLOR = 10
        DISPLAY "
                  Press any key to view the results~";

Rule Load4
   IF Type = Compression AND
     crit_load > (Load_static) AND
     safety > 1.25
   THEN Load = Good
        CLS
        COLOR = 11
        DISPLAY "
                       Design is safe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

          The critical load is greater than the maximum load &

          the calculated factor of safety is also reasonable."
```

```
        COLOR = 10
        DISPLAY "
                Press any key to view the results~";

Rule Load5
    IF Type = Extension AND
      safety > 1.25
    THEN Load = Good
        CLS
        COLOR = 11
        DISPLAY "



                  Design is very safe!!"
        COLOR = 15
        DISPLAY "



             The calculated factor of safety is reasonable.


                Press any key to view the results~";

Rule Load6
    IF Type = Extension AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "



                  Design is very unsafe!!"
        COLOR = 15
        DISPLAY "



             The calculated factor of safety is very low.

    Please choose a different material with better mechanical properties.



              Press any key to still view the results ~".

!====================== QUESTIONS BLOCK ===================

ASK Change_val: "Do you wish to change any one of the following?
```

1 = Material

2 = User values

3 = None

";

CHOICES Change_val: 1,   2,   3;

```
! Spring Design - Module 3 - Design calculations
! Saved as Design5.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!======================== ACTIONS BLOCK ========================
ACTIONS
        LOADFACTS Static
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Sp5

        COLOR = 15
        DISPLAY "
                    The user defined values are:

    Material          =       {Material}
    Load Type         =       {Type_load}
    Load              =       {Load_static}
    Deflection        =       {Defln}
    Eccentricity      =       {Eccen}
    Wire diameter     =       {Wire_dia}
    Free Height       =       {Free_ht}
    Factor of safety  =       {Factor}
    Ends              =       {Ends}"

        COLOR = 11
        DISPLAY "

                    Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
    CLS
    COLOR = 14
        DISPLAY "
                       MATERIAL PROPERTIES

              Material:                 {Materials}
              Ultimate Shear Stress:    {Ultimate}
              Yield Stress:             {Yield}
              Rigidity Modulus:         {Rigidity}"

    COLOR = 10
    DISPLAY "
                    Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
        SHIP File1,Rigidity
```

```
      SHIP File1,Load_static
      SHIP File1,Defln
      SHIP File1,Wire_dia
      SHIP File1,Free_ht
      SHIP File1,Eccen
      SHIP File1,Factor

   CALL Design5,""

      RECEIVE File2,mean_rad
      RECEIVE File2,turns
      RECEIVE File2,length
      RECEIVE File2,solidht
      RECEIVE File2,disp
      RECEIVE File2,index
      RECEIVE File2,wahl
      RECEIVE File2,safety
      RECEIVE File2,crit_load
      RECEIVE File2,reco

   FIND Load
   FIND Total
   CLS
   COLOR = 15
   DISPLAY "
                  Design of Helical Springs


      The design values for the specified spring are the following:

      Mean Coil Radius               =   {mean_rad}
      No. of active turns            =   {turns}
      Total # of turns               =   {Total}
      Solid height                   =   {solidht}
      Displacement from zero to max  =   {disp}
      Effective length               =   {length}
      Spring Index                   =   {index}
      Wahl factor                    =   {wahl}
      Factor of Safety               =   {safety}
"
   COLOR = 11
   DISPLAY "

          Press any key to continue with the design ~"

   CLS
   COLOR = 14
   FIND Val
   WCLOSE 1
;
!======================== RULES BLOCK =======================
Rule Change1
      IF Change_val = 1
      THEN Val = Change1
```

```
        CHAIN Matrl5;

Rule Change2
      IF Change_val = 2
      THEN Val = Change2
         CHAIN Spring;

Rule Change3
      IF Change_val = 3
      THEN Val = Change3
         CLS
         COLOR = 27
         DISPLAY "
                     THANK YOU!!!
"
         COLOR = 14
         DISPLAY "

                 for using the KBS for the
"
         COLOR = 31
         DISPLAY "
                 DESIGN OF HELICAL SPRINGS"

         COLOR = 11
         DISPLAY "

                Developed by: GIRIDHAR RAJA
         ME Student, New Jersey Institute of Technology,NJ

              Under the guidance of Dr. Nouri Levy
        Associate Professor, Department of Mechanical Engineering
             New Jersey Institute of Technology, NJ.
~";

Rule Ends1
   IF Ends = Square_and_ground_ends
   THEN Total = (turns + 2);

Rule Ends2
   IF Ends = Square_or_closed_ends
   THEN Total = (turns + 1.5);

Rule Ends3
   IF Ends = Plain_ends_not_ground
   THEN Total = (turns + 0.5);

Rule Ends4
   IF Ends = Plain_ends_ground
   THEN Total = (turns + 1);

Rule Ends5
   IF Ends = Two_&_half_turns
   THEN Total = (turns + 5);
```

```
Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.5)
    ELSE Total = (turns + 0.25);

Rule Load1
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety > 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"

        COLOR = 15
        DISPLAY "

        For {Material},

            Critical load       =       {crit_load}

            Maximum load        =       {Load_static}

            The Critical load is lesser than the Maximum load.

                This causes the spring to buckle.

        So, please increase the Mean Coil Radius or choose a material

                    with better Young's Modulus."

        COLOR = 10
        DISPLAY "
                Press any key to view the results~";

Rule Load2
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"

        COLOR = 15
        DISPLAY "

        For {Material},

        Critical load           =       {crit_load}
```

```
        Maximum load              =       {Load_static}

        Calculated safety factor     =       {safety}

           The Critical load is lesser than the Maximum load.

             The calculated factor of safety is also low.

      So, please increase the Mean Coil Radius or choose a material

                  with better Young's Modulus."

        COLOR = 10
        DISPLAY "
              Press any key to view the results~";

Rule Load3
   IF Type = Compression
     crit_load > (Load_static) AND
     safety < 1.25
   THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                 Design is very unsafe!!"

        COLOR = 15
        DISPLAY "

        For {Material},

        Calculated safety factor     =       {safety}

   The calculated factor of safety is low which makes the design unsafe.

     So, please choose a material with better mechanical properties."

        COLOR = 10
        DISPLAY "
              Press any key to view the results~";

Rule Load4
   IF Type = Compression AND
     crit_load > (Load_static) AND
     safety > 1.25
   THEN Load = Good
        CLS
        COLOR = 11
        DISPLAY "
                   Design is safe!!"

        COLOR = 15
        DISPLAY "
```

```
        For {Material},

          The Critical load is greater than the Maximum load &

          the calculated factor of safety is also reasonable."

        COLOR = 10
        DISPLAY "
                Press any key to view the results~";

Rule Load5
    IF Type = Extension AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "




                    Design is unsafe!!"
        COLOR = 15
        DISPLAY "




           The calculated factor of safety is very low.

    Please choose a different material with better mechanical properties.



               Please press any key to still view the results~";

Rule Load6
    IF Type = Extension AND
      safety > 1.25
    THEN Load = Good
        CLS
        COLOR = 11
        DISPLAY "




                   Design is very safe!!"
        COLOR = 15
        DISPLAY "



               The calculated factor of safety is OK.
```

```
          Press any key to view the results~";

!======================= QUESTIONS BLOCK ====================

ASK Change_val: "Do you wish to change any one of the following?

          1 = Material

          2 = User values

          3 = None
";
CHOICES Change_val: 1,  2,  3;
```

```
! Spring Design - Module 3 - Design calculations
! Saved as Design6.KBS

        EXECUTE;
        BKCOLOR = 0;
        RUNTIME;
!        ENDOFF;

!======================== ACTIONS BLOCK ========================
ACTIONS
        LOADFACTS Static
        LOADFACTS Matl
        LOADFACTS Desn
        LOADFACTS Design
        LOADFACTS Sp6

        COLOR = 15
        DISPLAY "
                    The user defined values are:

     Material            =       {Material}
     Load Type           =       {Type_load}
     Load                =       {Load_static}
     Deflection          =       {Defln}
     Eccentricity        =       {Eccen}
     Wire diameter       =       {Wire_dia}
     Mean Coil Radius    =       {Mean_rad}
     Free Height         =       {Free_ht}
     Factor of safety    =       {Factor}
     Ends                =       {Ends}"

        COLOR = 11
        DISPLAY "

                    Press any key to continue~"

        BCALL File," "
        GET Material=Materials.matl.ALL
    CLS
    COLOR = 14
        DISPLAY "
                       MATERIAL PROPERTIES

              Material:                  {Materials}
              Ultimate Shear Stress:       {Ultimate}
              Yield Stress:              {Yield}
              Rigidity Modulus:            {Rigidity}"

    COLOR = 10
    DISPLAY "
                    Press any key to continue~"

        SHIP File1,Ultimate
        SHIP File1,Yield
```

```
      SHIP File1,Rigidity
      SHIP File1,Load_static
      SHIP File1,Defln
      SHIP File1,Wire_dia
      SHIP File1,Mean_rad
      SHIP File1,Free_ht
      SHIP File1,Eccen
      SHIP File1,Factor

   CALL Design6,""

      RECEIVE File2,turns
      RECEIVE File2,length
      RECEIVE File2,solidht
      RECEIVE File2,disp
      RECEIVE File2,index
      RECEIVE File2,wahl
      RECEIVE File2,safety
      RECEIVE File2,crit_load
      RECEIVE File2,reco

   FIND Load
   FIND Total
   CLS
   COLOR = 15
   DISPLAY "
                  Design of Helical Springs


      The design values for the specified spring are the following:

      No. of active turns              =   {turns}
      Total # of turns                 =   {Total}
      Solid height                     =   {solidht}
      Displacement from zero to max    =   {disp}
      Effective length                 =   {length}
      Spring Index                     =   {index}
      Wahl factor                      =   {wahl}
      Factor of Safety                 =   {safety}
"
   COLOR = 11
   DISPLAY "

            Press any key to continue with the design ~"

   CLS
   COLOR = 14
   FIND Val
   WCLOSE 1
;
!======================== RULES BLOCK =======================
Rule Change1
      IF Change_val = 1
      THEN Val = Change1
```

```
            CHAIN Matrl6;

Rule Change2
        IF Change_val = 2
        THEN Val = Change2
            CHAIN Spring;

Rule Change3
        IF Change_val = 3
        THEN Val = Change3
            CLS
            COLOR = 27
            DISPLAY "
                           THANK YOU!!!
"
            COLOR = 14
            DISPLAY "

                     for using the KBS for the
"
            COLOR = 31
            DISPLAY "
                     DESIGN OF HELICAL SPRINGS"

            COLOR = 11
            DISPLAY "

                    Developed by: GIRIDHAR RAJA
            ME Student, New Jersey Institute of Technology,NJ

                 Under the guidance of Dr. Nouri Levy
           Associate Professor, Department of Mechanical Engineering
                New Jersey Institute of Technology, NJ.
~";

Rule Ends1
    IF Ends = Square_and_ground_ends
    THEN Total = (turns + 2);

Rule Ends2
    IF Ends = Square_or_closed_ends
    THEN Total = (turns + 1.5);

Rule Ends3
    IF Ends = Plain_ends_not_ground
    THEN Total = (turns + 0.5);

Rule Ends4
    IF Ends = Plain_ends_ground
    THEN Total = (turns + 1);

Rule Ends5
    IF Ends = Two_&_half_turns
    THEN Total = (turns + 5);
```

```
Rule Ends6
    IF Ends = Machine_half_loop OR
      Ends = Short_twisted_loop
    THEN Total = (turns + 0.5)
    ELSE Total = (turns + 0.25);

Rule Load1
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety > 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

            Critical Load       =       {crit_load}

            Maximum Load        =       {Load_static}

            The Critical load is lesser than the Maximum load.

                  This causes the spring to buckle.

       So, please increase the Mean Coil Radius or choose a material

                    with better Young's Modulus."
        COLOR = 10
        DISPLAY "

                  Press any key to view the results~";

Rule Load2
    IF Type = Compression AND
      crit_load < (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                    Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Critical Load           =       {crit_load}

        Maximum Load                =       {Load_static}
```

```
        Calculated safety factor    =       {safety}

            The Critical load is lesser than the Maximum load.

              The calculated factor of safety is also low.

        So, please increase the Mean Coil Radius or choose a material

                   with better Young's Modulus."
        COLOR = 10
        DISPLAY "

                 Press any key to view the results~";

Rule Load3
    IF Type = Compression AND
      crit_load > (Load_static) AND
      safety < 1.25
    THEN Load = Poor
        CLS
        COLOR = 11
        DISPLAY "
                      Design is very unsafe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

        Calculated safety factor    =       {safety}

    The calculated factor of safety is low which makes the design unsafe.

      So, please choose a material with better mechanical properties."
        COLOR = 10
        DISPLAY "

                 Press any key to view the results~";

Rule Load4
    IF Type = Compression AND
      crit_load > (Load_static) AND
      safety > 1.25
    THEN Load = Good
        CLS
        COLOR = 11
        DISPLAY "
                        Design is safe!!"
        COLOR = 15
        DISPLAY "

        For {Material},

           The Critical load is greater than the Maximum load &
```

```
        The calculated factor of safety is also reasonable."
      COLOR = 10
      DISPLAY "

              Press any key to view the results~";

Rule Load5
   IF Type = Extension AND
     safety < 1.25
   THEN Load = Poor
      CLS
      COLOR = 11
      DISPLAY "


                 Design is very unsafe!!"
      COLOR = 15
      DISPLAY "



           The calculated factor of safety is very low.

   Please choose a different material with better mechanical properties.


            Press any key to still view the results~";

Rule Load6
   IF Type = Extension AND
     safety > 1.25
   THEN Load = Good
      CLS
      COLOR = 11
      DISPLAY "



                 Design is very safe!!"
      COLOR = 15
      DISPLAY "



            The calculated factor of safety is OK




             Press any key to view the results~".

!====================== QUESTIONS BLOCK ===================

ASK Change_val: "Do you wish to change any one of the following?
```

1 = Material

2 = User values

3 = None

";

CHOICES Change_val: 1, 2, 3;

```
! KBS to help the user to choose a material according to the
! requirements
! Saved as Matl1.KBS
   EXECUTE;
   BKCOLOR = 3;
   RUNTIME;
!   ENDOFF;

!********************** ACTIONS BLOCK ***********************
ACTIONS

   BCALL Delete1," "
LOADFACTS Matl
CLS
COLOR = 15
FIND Temp
FIND Material
CLS

COLOR = 15
DISPLAY "






            The suggested material for the application is:"

   COLOR = 4
   DISPLAY "

                   =====> {Material}~"
   SAVEFACTS Design
   CHAIN Design1;

!************************** RULES BLOCK ***************************

Rule 1
   If Clock = Yes
   Then Material = Clock_spring_steel,

Rule 2
   If Corrosion = Yes AND
     Conduct = Yes AND
     Strength = Yes
   Then Material = Beryllium_copper;

Rule 3
   If Corrosion = Yes AND
     Conduct = Yes
   Then Material = Phosphor_bronze;
```

```
Rule 4
   If Conduct = Yes
   Then Material = Spring_brass;

Rule 5
   If Corrosion = Yes AND
     Temp = Yes AND
     Temperature = 2
   Then Material = Monel;

Rule 6
   If Temperature = 1
   Then Material = Music_wire;

Rule 7
   If Temperature = 2
   Then Material = Chrome_silicon_alloy_steel;

Rule 8
   If Temperature = 3
   Then Material = Inconel;

Rule 9
   If Temperature = 4
   Then Material = Inconel_X;

Rule 10
   If Corrosion = Yes
   Then Material = Stainless_steel;

!*********************** QUESTIONS BLOCK ************************
ASK Temp: "Is the spring used in an elevated temperature environment?";
CHOICES Temp: No,  Yes;

ASK Temperature: "What is the temperature range in which the spring
is used?
          1 = Temperature of 250-350 F
          2 = Temperature upto 450 F
          3 = Temperature upto 700 F
          4 = Temperature upto 900 F
";
CHOICES Temperature:1,  2,  3,  4;

ASK Clock: "Is the spring used in clock or a motor?";
CHOICES Clock: Yes,     No;

ASK Corrosion: "Do you require good corrosion resistance?";
CHOICES Corrosion:  Yes, No;

ASK Conduct: "Do you require good conductive properties?";
CHOICES Conduct:    Yes,     No;

ASK Strength: "Is high strength a requiremnt for the material?";
CHOICES Strength:   Yes, No;
```

```
!KBS to change Material
!Saved as Matrl1.KBS

   EXECUTE;
   BKCOLOR = 0;
   RUNTIME;
   !ENDOFF;

!======================== ACTIONS BLOCK =====================
ACTIONS

   BCALL Delete3," "
   COLOR = 14
   FIND Material
   SAVEFACTS Design
   CHAIN Design1
;
!=========================== RULES BLOCK =========================
Rule 1
   IF Mat_choice = 1
   THEN Material = Stainless_steel;

Rule 2
   IF Mat_choice = 2
   THEN Material = Monel;

Rule 3
   IF Mat_choice = 3
   THEN Material = Inconel;

Rule 4
   IF Mat_choice = 4
   THEN Material = Inconel_X;

Rule 5
   IF Mat_choice = 5
   THEN Material = Music_wire;

Rule 6
   IF Mat_choice = 6
   THEN Material = Carbon_steel;

Rule 7
   IF Mat_choice = 7
   THEN Material = Chrome_silicon_alloy_steel;

Rule 8
   IF Mat_choice = 8
   THEN Material = Flat_spring_steel;

Rule 9
   IF Mat_choice = 9
   THEN Material = Clock_spring_steel;
```

```
Rule 10
    IF Mat_choice = 10
    THEN Material = Spring_brass;

Rule 11
    IF Mat_choice = 11
    THEN Material = Beryllium_copper;

Rule 12
    IF Mat_choice = 12
    THEN Material = Phosphor_bronze;

Rule 13
    IF Mat_choice = 13
    THEN Material = Hard_drawn_wire;

Rule 14
    IF Mat_choice = 14
    THEN Material = Oil_tempered_wire;

!====================== QUESTIONS BLOCK =====================

ASK Mat_choice: "What material do you wish to use?

    The materials in the database are:

    1. Stainless_steel              2. Monel
    3. Inconel                      4. Inconel_X
    5. Music_wire                   6. Carbon_steel
    7. Chrome_silicon_alloy_steel   8. Flat_spring_steel
    9. Clock_spring_steel           10. Spring_brass
    11. Beryllium_copper            12. Phosphor_bronze
    13. Hard_drawn_wire             14. Oil_tempered_wire

";
CHOICES Mat_choice: 1,2,3,4,5,6,7,8,9,10,11,12,13,14;
```

APPENDIX C

PROGRAM CODES FOR CALCULATING THE PARAMETERS

```
/* DESIGN OF HELICAL SPRINGS         -         Saved as Cycle1.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi              3.141592654

main()
 {
        FILE     *ptr1,*ptr2,*ptr3;

        float    solidht,ultimate,yield,wdefln,factor,rigidity,eccen;
        int      i=0;
        int      act_turns;
        float    user_values[10];
        float    input,k,work_str,mean_rad,rate,safe_str,disp,mean_rad1;
        float    turns,turns1,turns2,freeht,index,safety,work_str1;
        float    wahl_fac1,wahl_fac2,wahl_fac,length,ratio_rad,ratio_str;
        double   wire,wire_dia,wire1,wire2.
        float    dia [ ] = { 0.004,0.005,0.006,0.007,0.008,0.009,0.010,
                 0.012,0.014,0.016,0.018,0.02,0.022,0.024,0.026,0.028,
                 0.032,0.035,0.040,0.047,0.054,0.063,0.072,0.080,0.091,
                 0.105,0.12,0.135,0.162,0.177,0.192,0.207,0.225,0.250,
                 0.312,0.375,0.437,0.5,0.562,0.625}.
        float    crit_def,crit_load,ratio,defln,enduran,loadmax,loadmin,life;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                 printf("Error opening file FILE1\n");
                 exit(-1);
        }
```

```
ptr3=fopen("output.bat","r");
ptr2=fopen("file2","w");


clrscr();

for (i=0;i<10;i++)
{
user_values[i] = 0.0;
}

i=0;
while ( fscanf(ptr1,"%f",&input) != EOF)
{
        user_values[i] = input;
        i++;
}

if (user_values[9] == 0)
{
        user_values[9] = default1;
}

ultimate = user_values[0];
yield = user_values[1];
rigidity = user_values[2];
enduran = user_values[3];
loadmax = user_values[4];
loadmin = user_values[5];
life = user_values[6];
wdefln = user_values[7];
mean_rad = user_values[8];
factor = user_values[9];

fclose(ptr1);

k = default2;
defln = wdefln * 1.2;
work_str = ultimate/(pow(life/1000.0,0.100343331)*factor);
wire = (16.0*k*loadmax*mean_rad)/(work_str*pi);
wire_dia = pow(wire,0.333333333);

/* Standardize the wire diameter */
for (i=0;i<50;i++)
{       if (wire_dia <= dia[i])
          {
           wire_dia = dia[i];
           break;
          }
}

printf("\n\rNew Wire dia: %f",wire_dia);

mean_rad1 = (work_str*pi*pow(wire_dia,3.0))/(16.0*k*loadmax);
```

```
        rate = (loadmax - loadmin)/defln;
        turns1 = rigidity*defln*pow(wire_dia,4.0);
        turns2 = 64.0*pow(mean_rad,3.0)*loadmax;
        turns = turns1/turns2;
        act_turns = (int)(turns+0.5);
        length = 2.0*pi*mean_rad*turns;
        solidht = turns*wire_dia;
        disp = loadmax/rate;
        freeht = solidht+disp;
        ratio = (freeht/2.0*mean_rad);
        index = 2.0*mean_rad/wire_dia;
        wahl_fac1 = ((4*index)-1)/((4*index)-4);
        wahl_fac2 = 0.615/index;
        wahl_fac = wahl_fac1+wahl_fac2;
        printf("\n\rWahl factor = %f",wahl_fac);
        ratio_rad = mean_rad1/mean_rad;
        printf("\n\rRadius ratio = %f",ratio_rad);

        safe_str = 0.577*yield;
        safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*loadmax);

/*      Write to a txt file saved as file2      */
        fprintf(ptr2,"%f\n",wire_dia);
        fprintf(ptr2,"%d\n",act_turns);
        fprintf(ptr2,"%f\n",length);
        fprintf(ptr2,"%f\n",solidht);
        fprintf(ptr2,"%f\n",disp);
        fprintf(ptr2,"%f\n",freeht);
        fprintf(ptr2,"%f\n",index);
        fprintf(ptr2,"%f\n",wahl_fac);
        fprintf(ptr2,"%f\n",safety);
        fprintf(ptr2,"%f\n",ratio_rad);

        if ( ratio <= 2.5 )
                { crit_def = 1.0 * freeht;}

        else if    ( ratio > 2.5 && ratio <= 3.0)
                { crit_def = 0.5 * freeht;}

        else if ( ratio > 3.0 && ratio <= 4.0)
                { crit_def = 0.3 * freeht;}

        else if ( ratio > 4.0 && ratio <= 5.5)
                { crit_def = 0.1 * freeht;}

        else if ( ratio > 5.5 )
                { crit_def = 0.05 * freeht;}

        crit_load = crit_def * rate;

        fprintf(ptr2,"%f\n",crit_load);

        fclose(ptr2);
}
```

```
/* DESIGN OF HELICAL SPRINGS      -      Saved as Cycle2.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi             3.141592654

main()
 {
        FILE    *ptr1,*ptr2,*ptr3;

        float   solidht,ultimate,yield,wdefln,factor,rigidity,eccen;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp,mean_rad1;
        float   turns,turns1,turns2,freeht,index,safety,work_str1;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_rad,ratio_str;
        double  wire,wire_dia,wire1,wire2;
        float   crit_def,crit_load,ratio,defln,enduran,loadmax,loadmin,life;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");


        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
        }

        i=0;
        while ( fscanf(ptr1,"%f",&input) != EOF)
        {
```

```
            user_values[i] = input;
            i++;
    }

    if (user_values[9] == 0)
    {
            user_values[9] = default1;
    }

    ultimate = user_values[0];
    yield = user_values[1];
    rigidity = user_values[2];
    enduran = user_values[3];
    loadmax = user_values[4];
    loadmin = user_values[5];
    life = user_values[6];
    wdefln = user_values[7];
    wire_dia = user_values[8];
    factor = user_values[9];

    fclose(ptr1);

    k = default2;
    defln = wdefln * 1.2;
    work_str = ultimate/(pow(life/1000,0.100343331)*factor);
    mean_rad = (work_str*pi*pow(wire_dia,3.0))/(16.0*k*loadmax);
    rate = (loadmax - loadmin)/defln;
    turns1 = rigidity*defln*pow(wire_dia,4.0);
    turns2 = 64.0*pow(mean_rad,3.0)*loadmax;
    turns = turns1/turns2;
    act_turns = (int)(turns+0.5);

    length = 2.0*pi*mean_rad*turns;
    solidht = turns*wire_dia;
    disp = loadmax/rate;
    freeht = solidht+disp;
    ratio = (freeht/2.0*mean_rad);
    index = 2.0*mean_rad/wire_dia;
    wahl_fac1 = ((4*index)-1)/((4*index)-4);
    wahl_fac2 = 0.615/index;
    wahl_fac = wahl_fac1+wahl_fac2.
    printf("\n\rWahl factor = %f",wahl_fac).
    ratio_rad = mean_rad1/mean_rad;

    safe_str = 0.577*yield;
    safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*loadmax);

/*  Write to a txt file saved as file2      */
    fprintf(ptr2,"%f\n",mean_rad);
    fprintf(ptr2,"%d\n",act_turns);
    fprintf(ptr2,"%f\n",length);
    fprintf(ptr2,"%f\n",solidht);
    fprintf(ptr2,"%f\n",disp);
    fprintf(ptr2,"%f\n",freeht);
```

```
	fprintf(ptr2,"%f\n",index);
	fprintf(ptr2,"%f\n",wahl_fac);
	fprintf(ptr2,"%f\n",safety);

	if ( ratio <= 2.5 )
		{ crit_def = 1.0 * freeht;}

	else if	( ratio > 2.5 && ratio <= 3.0)
		{ crit_def = 0.5 * freeht;}

	else if ( ratio > 3.0 && ratio <= 4.0)
		{ crit_def = 0.3 * freeht;}

	else if ( ratio > 4.0 && ratio <= 5.5)
		{ crit_def = 0.1 * freeht;}

	else if ( ratio > 5.5 )
		{ crit_def = 0.05 * freeht;}

	crit_load = crit_def * rate;

	fprintf(ptr2,"%f\n",crit_load);

	fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS        -        Saved as Cycle3.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi              3.141592654

main()
{
        FILE    *ptr1,*ptr2,*ptr3;

        float   solidht,ultimate,yield,wdefln,factor,rigidity,eccen;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp,mean_rad1;
        float   turns,turns1,turns2,freeht,index,safety,work_str1;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_rad,ratio_str;
        double  wire,wire_dia,wire1,wire2;
        float   crit_def,crit_load,ratio,defln,enduran,loadmax,loadmin,life;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");


        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
        }

        i=0;
        while ( fscanf(ptr1,"%f",&input) != EOF)
        {
```

```
            user_values[i] = input;
            i++;
}

if (user_values[10] == 0)
{
            user_values[10] = default1;
}

ultimate = user_values[0];
yield = user_values[1];
rigidity = user_values[2];
enduran = user_values[3];
loadmax = user_values[4];
loadmin = user_values[5];
life = user_values[6];
wdefln = user_values[7];
wire_dia = user_values[8];
mean_rad = user_values[9];
factor = user_values[10];

fclose(ptr1);

k = default2;
defln = wdefln * 1.2;
work_str = ultimate/(pow(life/1000,0.100343331)*factor);
printf("\n\rWorking stress = %f",work_str);
index = 2.0*mean_rad / wire_dia;

rate = (loadmax - loadmin)/defln;
turns1 = rigidity*defln*pow(wire_dia,4.0);
turns2 = 64.0*pow(mean_rad,3.0)*loadmax;
turns = turns1/turns2;
act_turns = (int)(turns+0.5);
printf("\nTurns = %d",act_turns);

length = 2.0*pi*mean_rad*turns;
solidht = turns*wire_dia;
disp = loadmax/rate;
freeht = solidht+disp;
ratio = (freeht/2.0*mean_rad);
printf("\n\n\rSolid Height: %12.2f",solidht);
printf("\nLength = %f",length);

wahl_fac1 = ((4*index)-1)/((4*index)-4);
wahl_fac2 = 0.615/index;
wahl_fac = wahl_fac1+wahl_fac2;
printf("\n\rWahl factor = %f",wahl_fac);

safe_str = 0.577*yield;
safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*loadmax);
printf("\n\rSafety = %f",safety);
printf("\n\rIndex  = %f",index);
printf("\n\rSafe stress = %f",safe_str);
```

```
/*      Write to a txt file saved as file2      */
        fprintf(ptr2,"%d\n",act_turns);
        fprintf(ptr2,"%f\n",length);
        fprintf(ptr2,"%f\n",solidht);
        fprintf(ptr2,"%f\n",disp);
        fprintf(ptr2,"%f\n",freeht);
        fprintf(ptr2,"%f\n",index);
        fprintf(ptr2,"%f\n",wahl_fac);
        fprintf(ptr2,"%f\n",safety);

        if ( ratio <= 2.5 )
                { crit_def = 1.0 * freeht;}

        else if    ( ratio > 2.5 && ratio <= 3.0)
                { crit_def = 0.5 * freeht;}

        else if ( ratio > 3.0 && ratio <= 4.0)
                { crit_def = 0.3 * freeht;}

        else if ( ratio > 4.0 && ratio <= 5.5)
                { crit_def = 0.1 * freeht;}

        else if ( ratio > 5.5 )
                { crit_def = 0.05 * freeht;}

        crit_load = crit_def * rate;
        printf("\n\rRatio of l/D = %f",ratio);
        printf("\n\rCrit defln = %f",crit_def);
        printf("\n\rCritical load = %f",crit_load);

        fprintf(ptr2,"%f\n",crit_load);

        fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS        -        Saved as Design1.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi              3.141592654

main()
{
        FILE    *ptr1,*ptr2,*ptr3;

        float   load,solidht,ultimate,yield,wdefln,factor,rigidity,eccen;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp,mean_rad1;
        float   turns,turns1,turns2,frecht,index,safety,work_str1;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_rad,ratio_str;
        double  wire,wire_dia,wire1,wire2;
        float   dia [ ] = { 0.004,0.005,0.006,0.007,0.008,0.009,0.010,
                0.012,0.014,0.016,0.018,0.02,0.022,0.024,0.026,0.028,
                0.032,0.035,0.040,0.047,0.054,0.063,0.072,0.080,0.091,
                0.105,0.12,0.135,0.162,0.177,0.192,0.207,0.225,0.250,
                0.312,0.375,0.437,0.5,0.562,0.625};
        float   crit_def,crit_load,ratio,defln;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");


        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
```

```
}

i=0;
while ( fscanf(ptr1,"%f",&input) != EOF)
{
            user_values[i] = input;
            i++;
}

if (user_values[7] == 0)
{
            user_values[7] = default1;
}

ultimate = user_values[0];
yield = user_values[1];
rigidity = user_values[2];
load = user_values[3];
wdefln = user_values[4];
mean_rad = user_values[5];
eccen = user_values[6];
factor = user_values[7];

fclose(ptr1);

k = default2;
defln = wdefln * 1.2;
work_str = (yield*0.577)*(1+(eccen/mean_rad))/factor;
wire = (16.0*k*load*mean_rad)/(work_str*pi);
wire_dia = pow(wire,0.333333333);
printf("The value of Wire =%f\n",wire);
printf("Wire dia = %f\n",wire_dia);

/* Standardize the wire diameter */
for (i=0;i<50;i++)
{       if (wire_dia <= dia[i])
            {
              wire_dia = dia[i];
              break;
            }
}

printf("\n\rNew Wire dia: %f",wire_dia);

mean_rad1 = (work_str*pi*pow(wire_dia,3.0))/(16.0*k*load);
rate = load/defln;
turns1 = rigidity*defln*pow(wire_dia,4.0);
turns2 = 64.0*pow(mean_rad,3.0)*load;
turns = turns1/turns2;
act_turns = (int)(turns+0.5);
printf("\nTurns = %d",act_turns);
printf("\nMean rad = %f",mean_rad);

length = 2.0*pi*mean_rad*turns;
```

```
        solidht = turns*wire_dia;
        disp = load/rate;
        freeht = solidht+disp;
        ratio = (freeht/2.0*mean_rad);

        index = 2.0*mean_rad/wire_dia;
        wahl_fac1 = ((4*index)-1)/((4*index)-4);
        wahl_fac2 = 0.615/index;
        wahl_fac = wahl_fac1+wahl_fac2;
        printf("\n\rWahl factor = %f",wahl_fac);
        ratio_rad = mean_rad1/mean_rad;
        printf("\n\rRadius ratio = %f",ratio_rad);

        safe_str = 0.577*yield;
        safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*load);

/*      Write to a txt file saved as file2      */
        fprintf(ptr2,"%f\n",wire_dia);
        fprintf(ptr2,"%d\n",act_turns);
        fprintf(ptr2,"%f\n",length);
        fprintf(ptr2,"%f\n",solidht);
        fprintf(ptr2,"%f\n",disp);
        fprintf(ptr2,"%f\n",freeht);
        fprintf(ptr2,"%f\n",index);
        fprintf(ptr2,"%f\n",wahl_fac);
        fprintf(ptr2,"%f\n",safety);
        fprintf(ptr2,"%f\n",ratio_rad);

        if ( ratio <= 2.5 )
                { crit_def = 1.0 * freeht;}

        else if   ( ratio > 2.5 && ratio <= 3.0)
                { crit_def = 0.5 * freeht;}

        else if ( ratio > 3.0 && ratio <= 4.0)
                { crit_def = 0.3 * freeht;}

        else if ( ratio > 4.0 && ratio <= 5.5)
                { crit_def = 0.1 * freeht;}

        else if ( ratio > 5.5 )
                { crit_def = 0.05 * freeht;}

        crit_load = crit_def * rate;
        printf("\n\rRatio of l/D = %f",ratio);
        printf("\n\rCrit defln = %f",crit_def);
        printf("\n\rCritical load = %f",crit_load);

        fprintf(ptr2,"%f\n",crit_load);

        fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS        -        Saved as Design2.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi             3.141592654

main()
{
        FILE    *ptr1,*ptr2,*ptr3;

        float   load,solidht,ultimate,yield,wdefln,factor,rigidity;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp;
        float   turns,turns1,turns2,frecht,index,safety;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_str;
        double  wire_dia,defln;
        float   crit_def,crit_load,ratio,eccen;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");
        fclose(ptr3);
        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
        }

        i=0;
        while ( fscanf(ptr1,"%f",&input) != EOF)
        {
                user_values[i] = input;
```

```
		i++;
	}

	if (user_values[7] == 0)
	{
		user_values[7] = default1;
	}

	ultimate = user_values[0];
	yield = user_values[1];
	rigidity = user_values[2];
	load = user_values[3];
	wdefln = user_values[4];
	wire_dia = user_values[5];
	eccen = user_values[6];
	factor = user_values[7];

	fclose(ptr1);

	defln = wdefln * 1.2;
	printf("\nDeflection = %4.2f",defln);
	k = default2;
	work_str = (yield*0.577)*(1+(eccen/0.25))/factor;
	mean_rad = (work_str*pi*pow(wire_dia,3.0))/(16.0*k*load);
	rate = load/defln;
	turns1 = rigidity*defln*pow(wire_dia,4.0);
	turns2 = 64.0*pow(mean_rad,3.0)*load;
	turns = turns1/turns2;
	act_turns = (int)(turns+0.5);
	length = 2.0*pi*mean_rad*turns;
	solidht = turns*wire_dia;
	disp = load/rate;
	freeht = solidht+disp;
	ratio = (freeht/2.0*mean_rad);
	printf("\n\n\rSolid Height: %12.2f Free ht = %12.2f",solidht,freeht);
	printf("\nLength = %f",length);

	index = 2.0*mean_rad/wire_dia;
	wahl_fac1 = ((4*index)-1)/((4*index)-4);
	wahl_fac2 = 0.615/index;
	wahl_fac = wahl_fac1+wahl_fac2;
	printf("\n\rWahl factor = %f",wahl_fac);

	safe_str = 0.577*yield;
	safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*load);
	printf("\n\rSafety = %f",safety);
	printf("\n\rIndex  = %f",index);
	printf("\n\rSafe stress = %f",safe_str);

/*	Write to a txt file saved as file2	*/
	fprintf(ptr2,"%f\n",wire_dia);
	fprintf(ptr2,"%d\n",act_turns);
	fprintf(ptr2,"%f\n",length);
	fprintf(ptr2,"%f\n",solidht);
```

```
        fprintf(ptr2,"%f\n",disp);
        fprintf(ptr2,"%f\n",freeht);
        fprintf(ptr2,"%f\n",index);
        fprintf(ptr2,"%f\n",wahl_fac);
        fprintf(ptr2,"%f\n",safety);

        if ( ratio <= 2.5 )
                { crit_def = 1.0 * freeht;}

        else if   ( ratio > 2.5 && ratio <= 3.0)
                { crit_def = 0.5 * freeht;}

        else if ( ratio > 3.0 && ratio <= 4.0)
                { crit_def = 0.3 * freeht;}

        else if ( ratio > 4.0 && ratio <= 5.5)
                { crit_def = 0.1 * freeht;}

        else if ( ratio > 5.5 )
                { crit_def = 0.05 * freeht;}

        crit_load = crit_def * rate;
        printf("\n\rRatio of l/D = %f",ratio);
        printf("\n\rCrit defln = %f",crit_def);
        printf("\n\rCritical load = %f",crit_load);

        fprintf(ptr2,"%f\n",crit_load);

        fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS        -        Saved as Design3.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi             3.141592654

main()
{
        FILE     *ptr1,*ptr2,*ptr3;

        float    load,solidht,ultimate,yield,wdefln,factor,rigidity;
        int      i=0;
        int      act_turns;
        float    user_values[10];
        float    input,k,work_str,mean_rad,rate,safe_str,disp,mean_rad1;
        float    turns,turns1,turns2,freeht,index,safety,work_str1;
        float    wahl_fac1,wahl_fac2,wahl_fac,length,ratio_rad,ratio_str;
        double   wire,wire_dia,wire1,wire2;
        float    dia [ ] = { 0.004,0.005,0.006,0.007,0.008,0.009,0.010,
                 0.012,0.014,0.016,0.018,0.02,0.022,0.024,0.026,0.028,
                 0.032,0.035,0.040,0.047,0.054,0.063,0.072,0.080,0.091,
                 0.105,0.12,0.135,0.162,0.177,0.192,0.207,0.225,0.250,
                 0.312,0.375,0.437,0.5,0.562,0.625};
        float    crit_def,crit_load,ratio,eccen,defln;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");


        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
```

```
}

i=0;
while ( fscanf(ptr1,"%f",&input) != EOF)
{
        user_values[i] = input;
        i++;
}

if (user_values[8] == 0)
{
        user_values[8] = default1;
}

ultimate = user_values[0];
yield = user_values[1];
rigidity = user_values[2];
load = user_values[3];
wdefln = user_values[4];
mean_rad = user_values[5];
freeht = user_values[6];
eccen = user_values[7];
factor = user_values[8];

fclose(ptr1);

defln = wdefln * 1.2;
k = default2;
work_str = (0.577*yield)*(1+(eccen/mean_rad))/factor;
rate = load/defln;
wire1 = 16*mean_rad*k*load;
wire2 = pi*work_str;
wire_dia = pow(wire1/wire2,0.33333333);
turns1 = pow(wire_dia,4.0)*rigidity*defln;
turns2 = 64*pow(mean_rad,3.0)*load;
turns = turns1/turns2;
act_turns = (int) (turns);
printf("\n\rThe wire diameter = %f",wire_dia);

/* Standardize the wire diameter */
for (i=0;i<50;i++)
{     if (wire_dia <= dia[i])
        {
         wire_dia = dia[i];
         break;
        }
}

printf("\n\rNew Wire dia: %f",wire_dia);

mean_rad1 = (work_str*pi*pow(wire_dia,3.0))/(16.0*k*load);
printf("\nTurns = %d",act_turns);
printf("\nMean rad = %f",mean_rad1);
```

```
        length = 2.0*pi*mean_rad*act_turns;
        solidht = act_turns*wire_dia;
        disp = load/rate;
        freeht = solidht+disp;
        printf("\n\rFree height = %f",freeht);
        ratio = (freeht/2.0*mean_rad);
        printf("\n\rSolid Height:   %12.2f",solidht);
        printf("\nLength = %f",length);

        index = 2.0*mean_rad/wire_dia;
        wahl_fac1 = ((4*index)-1)/((4*index)-4);
        wahl_fac2 = 0.615/index;
        wahl_fac = wahl_fac1+wahl_fac2;
        printf("\n\rWahl factor = %f",wahl_fac);
        ratio_rad = mean_rad1/mean_rad;
        printf("\n\rRadius ratio = %f",ratio_rad);

        safe_str = 0.577*yield;
        safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*load);

/*      Write to a txt file saved as file2      */
        fprintf(ptr2,"%f\n",wire_dia);
        fprintf(ptr2,"%d\n",act_turns);
        fprintf(ptr2,"%f\n",length);
        fprintf(ptr2,"%f\n",solidht);
        fprintf(ptr2,"%f\n",freeht);
        fprintf(ptr2,"%f\n",disp);
        fprintf(ptr2,"%f\n",index);
        fprintf(ptr2,"%f\n",wahl_fac);
        fprintf(ptr2,"%f\n",safety);
        fprintf(ptr2,"%f\n",ratio_rad);

        if ( ratio <= 2.5 )
                { crit_def = 1.0 * freeht;}

        else if   ( ratio > 2.5 && ratio <= 3.0)
                { crit_def = 0.5 * freeht;}

        else if ( ratio > 3.0 && ratio <= 4.0)
                { crit_def = 0.3 * freeht;}

        else if ( ratio > 4.0 && ratio <= 5.5)
                { crit_def = 0.1 * freeht;}

        else if ( ratio > 5.5 )
                { crit_def = 0.05 * freeht;}

        crit_load = crit_def * rate;

        fprintf(ptr2,"%f\n",crit_load);

        fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS          -          Saved as Design4.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi              3.141592654

main()
{
        FILE    *ptr1,*ptr2,*ptr3;

        float   load,solidht,ultimate,yield,wdefln,factor,rigidity;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp,mean_rad1;
        float   turns,turns1,turns2,frecht,index,safety,work_str1;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_rad,ratio_str;
        double  wire_dia;
        float   crit_def,crit_load,ratio,eccen,defln;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");


        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
        }

        i=0;
        while ( fscanf(ptr1,"%f",&input) != EOF)
        {
```

```
            user_values[i] = input;
            i++;
    }

    if (user_values[8] == 0)
    {
            user_values[8] = default1;
    }

    ultimate = user_values[0];
    yield = user_values[1];
    rigidity = user_values[2];
    load = user_values[3];
    wdefln = user_values[4];
    mean_rad = user_values[5];
    wire_dia = user_values[6];
    eccen = user_values[7];
    factor = user_values[8];

    fclose(ptr1);

    defln = wdefln * 1.2;
    k = default2;
    rate = load/defln;
    turns1 = pow(wire_dia,4.0)*rigidity*defln;
    turns2 = 64*pow(mean_rad,3.0)*load;
    turns = turns1/turns2;
    act_turns = (int) (turns);
    work_str = (yield*0.577)*(1+(eccen/mean_rad))/factor;
    printf("\nTurns = %d",act_turns);

    length = 2.0*pi*mean_rad*act_turns;
    solidht = act_turns*wire_dia;
    disp = load/rate;
    solidht = turns * wire_dia;
    freeht = solidht + disp;
    ratio = (freeht/2.0*mean_rad);
    printf("\n\n\rSolid Height: %12.2f",solidht);
    printf("\nLength = %f",length);

    index = 2.0*mean_rad/wire_dia;
    wahl_fac1 = ((4*index)-1)/((4*index)-4);
    wahl_fac2 = 0.615/index;
    wahl_fac = wahl_fac1+wahl_fac2;
    printf("\n\rWahl factor = %f",wahl_fac);

    safe_str = 0.577*yield;
    safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*load);
    printf("\n\rSafety = %f",safety);
    printf("\n\rIndex  = %f",index);
    printf("\n\rSafe stress = %f",safe_str);

/*  Write to a txt file saved as file2      */
    fprintf(ptr2,"%d\n",act_turns);
```

```
fprintf(ptr2,"%f\n",length);
fprintf(ptr2,"%f\n",solidht);
fprintf(ptr2,"%f\n",disp);
fprintf(ptr2,"%f\n",freeht);
fprintf(ptr2,"%f\n",index);
fprintf(ptr2,"%f\n",wahl_fac);
fprintf(ptr2,"%f\n",safety);

if ( ratio <= 2.5 )
        { crit_def = 1.0 * freeht;}

else if    ( ratio > 2.5 && ratio <= 3.0)
        { crit_def = 0.5 * freeht;}

else if ( ratio > 3.0 && ratio <= 4.0)
        { crit_def = 0.3 * freeht;}

else if ( ratio > 4.0 && ratio <= 5.5)
        { crit_def = 0.1 * freeht;}

else if ( ratio > 5.5 )
        { crit_def = 0.05 * freeht;}

crit_load = crit_def * rate;
printf("\n\rRatio of l/D = %f",ratio);
printf("\n\rCrit defln = %f",crit_def);
printf("\n\rCritical load = %f",crit_load);

fprintf(ptr2,"%f\n",crit_load);

fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS    -    Saved as Design5.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi        3.141592654

main()
{
        FILE    *ptr1,*ptr2,*ptr3;

        float   load,solidht,ultimate,yield,wdefln,factor,rigidity;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp;
        float   turns,turns1,turns2,freeht,index,safety;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_str;
        double  wire_dia,defln;
        float   crit_def,crit_load,ratio,eccen,rad1,rad2,rad;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");
        fclose(ptr3);
        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
        }

        i=0;
        while ( fscanf(ptr1,"%f",&input) != EOF)
        {
                user_values[i] = input;
```

```
            i++;
        }

        if (user_values[8] == 0)
        {
            user_values[8] = default1;
        }

        ultimate = user_values[0];
        yield = user_values[1];
        rigidity = user_values[2];
        load = user_values[3];
        wdefln = user_values[4];
        wire_dia = user_values[5];
        freeht = user_values[6];
        eccen = user_values[7];
        factor = user_values[8];

        fclose(ptr1);

        defln = wdefln * 1.2;
        printf("\nDeflection = %4.2f",defln);
        k = default2;
        rate = load/defln;
        solidht = freeht - (load/rate);
        act_turns = (int) (solidht/wire_dia);
        printf("\n\rTurns = %d",act_turns);
        rad1 = pow(wire_dia,3.0)*rigidity*defln;
        rad2 = 64*act_turns*load;
        mean_rad = pow(rad1/rad2,0.3333333);
        printf("\n\rRad = %8.2f",mean_rad);
        length = 2.0*mean_rad*act_turns;
        disp = load/rate;
        ratio = (freeht/2.0*mean_rad);
        printf("\n\rDisp = %8.2f",disp);
        printf("\n\n\rSolid Height: %12.2f Free ht = %12.2f",solidht,freeht);
        printf("\nLength = %f",length);

        index = 2.0*mean_rad/wire_dia;
        wahl_fac1 = ((4*index)-1)/((4*index)-4);
        wahl_fac2 = 0.615/index;
        wahl_fac = wahl_fac1+wahl_fac2;
        printf("\n\rWahl factor = %f",wahl_fac);

        safe_str = 0.577*yield;
        safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*load);
        printf("\n\rSafety = %f",safety);
        printf("\n\rIndex  = %f",index);
        printf("\n\rSafe stress = %f",safe_str);

/*      Write to a txt file saved as file2      */
        fprintf(ptr2,"%f\n",mean_rad);
        fprintf(ptr2,"%d\n",act_turns);
        fprintf(ptr2,"%f\n",length);
```

```
        fprintf(ptr2,"%f\n",solidht);
        fprintf(ptr2,"%f\n",disp);
        fprintf(ptr2,"%f\n",index);
        fprintf(ptr2,"%f\n",wahl_fac);
        fprintf(ptr2,"%f\n",safety);

        if ( ratio <= 2.5 )
                { crit_def = 1.0 * freeht;}

        else if    ( ratio > 2.5 && ratio <= 3.0)
                { crit_def = 0.5 * freeht;}

        else if ( ratio > 3.0 && ratio <= 4.0)
                { crit_def = 0.3 * freeht;}

        else if ( ratio > 4.0 && ratio <= 5.5)
                { crit_def = 0.1 * freeht;}

        else if ( ratio > 5.5 )
                { crit_def = 0.05 * freeht;}

        crit_load = crit_def * rate;
        printf("\n\rRatio of l/D = %f",ratio);
        printf("\n\rCrit defln = %f",crit_def);
        printf("\n\rCritical load = %f",crit_load);

        fprintf(ptr2,"%f\n",crit_load);

        fclose(ptr2);

}
```

```
/* DESIGN OF HELICAL SPRINGS        -        Saved as Design6.C

The following program calculates the various values
for the spring design.
The user values are read from a text file created by VP/Expert
and the calculations are made. The output values are stored in
a text file which can be retrieved in VP/Expert */

#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include<math.h>
#include<ctype.h>
#include<float.h>

#define default1  2.0
#define default2  1.0
#define pi              3.141592654

main()
{
        FILE    *ptr1,*ptr2,*ptr3;

        float   load,solidht,ultimate,yield,wdefln,factor,rigidity;
        int     i=0;
        int     act_turns;
        float   user_values[10];
        float   input,k,work_str,mean_rad,rate,safe_str,disp;
        float   turns,turns1,turns2,freeht,index,safety;
        float   wahl_fac1,wahl_fac2,wahl_fac,length,ratio_str;
        double  wire_dia,defln;
        float   crit_def,crit_load,ratio,eccen,rad1,rad2,rad;


        if ((ptr1=fopen("file1","r")) == NULL)
        {
                printf("Error opening file FILE1\n");
                exit(-1);
        }

        ptr3=fopen("output.bat","r");
        ptr2=fopen("file2","w");
        fclose(ptr3);
        clrscr();

        for (i=0;i<10;i++)
        {
        user_values[i] = 0.0;
        }

        i=0;
        while ( fscanf(ptr1,"%f",&input) != EOF)
        {
                user_values[i] = input;
```

```
        i++;
    }

    if (user_values[9] == 0)
    {
        user_values[9] = default1;
    }

    ultimate = user_values[0];
    yield = user_values[1];
    rigidity = user_values[2];
    load = user_values[3];
    wdefln = user_values[4];
    wire_dia = user_values[5];
    mean_rad = user_values[6];
    freeht = user_values[7];
    eccen = user_values[8];
    factor = user_values[9];

    fclose(ptr1);

    defln = wdefln * 1.2;
    printf("\nDeflection = %4.2f",defln);
    k = default2;
    rate = load/defln;
    solidht = freeht - (load/rate);
    act_turns = (int) (solidht/wire_dia);
    printf("\n\rTurns = %d",act_turns);
    length = 2.0*mean_rad*act_turns;
    disp = load/rate;
    ratio = (freeht/2.0*mean_rad);
    printf("\n\rDisp = %8.2f",disp);
    printf("\n\n\rSolid Height: %12.2f Free ht = %12.2f",solidht,freeht);
    printf("\nLength = %f",length);

    index = 2.0*mean_rad/wire_dia;
    wahl_fac1 = ((4*index)-1)/((4*index)-4);
    wahl_fac2 = 0.615/index;
    wahl_fac = wahl_fac1+wahl_fac2;
    printf("\n\rWahl factor = %f",wahl_fac);

    safe_str = 0.577*yield;
    safety = (safe_str*3.1415*pow(wire_dia,3))/(16.0*mean_rad*wahl_fac*load);
    printf("\n\rSafety = %f",safety);
    printf("\n\rIndex  = %f",index);
    printf("\n\rSafe stress = %f",safe_str);

/*  Write to a txt file saved as file2      */
    fprintf(ptr2,"%d\n",act_turns);
    fprintf(ptr2,"%f\n",length);
    fprintf(ptr2,"%f\n",solidht);
    fprintf(ptr2,"%f\n",disp);
    fprintf(ptr2,"%f\n",index);
    fprintf(ptr2,"%f\n",wahl_fac);
```

```
	fprintf(ptr2,"%f\n",safety);

	if ( ratio <= 2.5 )
		{ crit_def = 1.0 * freeht;}

	else if    ( ratio > 2.5 && ratio <= 3.0)
		{ crit_def = 0.5 * freeht;}

	else if ( ratio > 3.0 && ratio <= 4.0)
		{ crit_def = 0.3 * freeht;}

	else if ( ratio > 4.0 && ratio <= 5.5)
		{ crit_def = 0.1 * freeht;}

	else if ( ratio > 5.5 )
		{ crit_def = 0.05 * freeht;}

	crit_load = crit_def * rate;
	printf("\n\rRatio of l/D = %f",ratio);
	printf("\n\rCrit defln = %f",crit_def);
	printf("\n\rCritical load = %f",crit_load);

	fprintf(ptr2,"%f\n",crit_load);

	fclose(ptr2);

}
```

REFERENCES

1. Wahl, A.M. Mechanical Springs. New York: McGraw-Hill Book Company, Inc. 1963: 3-98.

2. Shigley, J. E. Mechanical Engineering Design. New York: McGraw-Hill Book Company, Inc. 1972: 352-380.

3. Esposito, A. Machine Design. Ohio: Charles E. Merill Publishing Company. 1975: 393-414.

4. Pham, D.T. (Ed.). Artificial Intelligence in Design. Springer-Verlag. 1991. 319-491.

5. Tong, C., Sriram, D. Artificial Intelligence in Engineering Design Vol. II. California: Academic Press, Inc. 1992: 1-287.

6. Tong, C., Sriram, D. Artificial Intelligence in Engineering Design Vol. III. California: Academic Press, Inc. 1992: 199-234.

7. Dologite, D.G. Developing Knowledge-Based Systems Using VP-Expert. New York: Macmillan Publishing Company. 1993: 1-356.

8. Avallone, E.A., Baumeister, T. Marks' Standard Handbook for Mechanical Engineers. New York: McGraw-Hill Book Company, Inc. 1987: 8.71-8.80.

9. Kochan, S.G., Wood, P.H. Topics in C Programming. John Wiley & Sons, Inc. 1991: 1-312.

10. Kothari, H. "Optimum Design of Helical Springs." *J. of Engineering for Industry* (1980): 69-73.

11. Hinkle, R.T., Morse, I.E. "Design of Helical Springs for Minimum Weight, Volume, and Length." *J. of Engineering for Industry* (1959): 37-42.

12. The Spring Research Association. Helical Springs. Hampshire: Oxford University Press. 1974.