Extended Surface Heat Sinks for Electronic Components: A Computer Optimization

by

John Reynold Gensure

June 1992

Thesis Advisor: Allan D. Kraus

Approved for public release; distribution is unlimited.
EXTENDED SURFACE HEAT SINKS FOR ELECTRONIC COMPONENTS: A COMPUTER OPTIMIZATION

Gensure, John Reynold

Master's Thesis

THE VIEWS EXPRESSED IN THIS THESIS ARE THOSE OF THE AUTHOR AND DO NOT REFLECT THE OFFICIAL POLICY OR POSITION OF THE DEPT. OF DEFENSE OR THE U.S. GOVT.

Extended Surfaces, Fins, Heat Sinks

Heat sinks consisting of individual fins and arrays of fins are used extensively throughout the Navy and industry. The fins serve to increase the surface area through which heat is transferred to the surrounding environment by natural convection. Extended surfaces or fins are commonly found on electronic components ranging from power supplies to transformers. The dissipation and subsequent rejection of potentially destructive self produced heat is an important aspect of electronic equipment design.

Fin design theory is examined starting with the optimization of individual fin dimensions. The insights obtained are utilized in an investigation of the optimal number and spacing of elements in an array of fins. The results are implemented in a computer program written in ADA and compiled for use on IBM compatible machines. The program takes as inputs thermal and physical data and outputs an optimized fin configuration. Menu driven, the program serves to greatly simplify and accelerate the...
Extended Surface Heat Sinks for
Electronic Components:
A Computer Optimization

by

John Reynold Gensure
Lieutenant, United States Navy
B.S., United States Naval Academy, 1986

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1992

Author: John Reynold Gensure

Approved by: Allan D. Kraus, Thesis Advisor
Matthew Dennis Kelleher, Second Reader
Michael A. Morgan, Chairman

Department of Electrical and Computer Engineering
ABSTRACT

Heat sinks consisting of individual fins and arrays of fins are used extensively throughout the Navy and industry. The fins serve to increase the surface area through which heat is transferred to the surrounding environment by natural convection. Extended surfaces or fins are commonly found on electronic components ranging from power supplies to transformers. The dissipation and subsequent rejection of potentially destructive self produced heat is an important aspect of electronic equipment design.

Fin design theory is examined starting with the optimization of individual fin dimensions. The insights obtained are utilized in an investigation of the optimal number and spacing of elements in an array of fins. The results are implemented in a computer program written in ADA and compiled for use on IBM compatible machines. The program takes as inputs thermal and physical data and outputs an optimized fin configuration. Menu driven, the program is easily employed without any amplifying documentation. The program serves to greatly simplify and accelerate the fin design process and should be an invaluable tool to electronic component designers, especially those with a limited background in heat transfer and fin optimization theory.
# TABLE OF CONTENTS

I. INTRODUCTION ........................................... 1

II. SINGLE FIN HEAT TRANSFER THEORY .......................... 4
   A. INTRODUCTION ....................................... 4
   B. ARBITRARILY SHAPED FINS ............................. 4
       1. Temperature Profile Differential Equation .......... 4
       2. Temperature Distribution Equation .................. 7
       3. Fin Tip Temperature Equation ........................ 9
       4. Heat Dissipation Equation ........................... 10
       5. Fin Efficiency ..................................... 10
   C. SPECIFIC FIN CONFIGURATIONS ........................... 12
       1. Cylindrical Spine .................................. 12
       2. Logitudinal Fin of Rectangular Profile .............. 13

III. SINGLE FIN OPTIMIZATION THEORY ........................... 15
   A. INTRODUCTION ....................................... 15
   B. CYLINDRICAL SPINE ................................... 15
       1. Maximum Heat Transfer for a Given Volume .......... 15
   C. RECTANGULAR FIN .................................... 18
       1. Maximum Heat Transfer for a Given Volume and Length. 18
       2. Minimum Volume for a Given Heat Transfer and Length... 20

IV. MULTIPLE FIN HEAT TRANSFER THEORY ........................ 22
   A. INTRODUCTION ....................................... 22
I. INTRODUCTION

Convection is the transfer of energy from a heat source to a cooler surrounding fluid due to the motion of the fluid. The rate of heat transfer by convection, $q_c$, is governed by Newton’s Law of Cooling which is

$$q_c = hA_s(T - T_\infty)$$  \hspace{1cm} (1-1)

where $h$ is the coefficient of heat transfer in convection, $A_s$ is the surface area, $T$ is the fin temperature at position $x$, and $T_\infty$ is the ambient temperature. One method of increasing $q_c$ is to increase $h$ by going from natural to forced convection. Natural convection is a phenomenon in which the fluid motion is induced by differences in buoyancy and density between parcels of fluid at the confining surface and within the bulk of the fluid. Forced convection occurs when the fluid motion is induced or forced by a fan, pump, or blower. Because the addition of a fan or pump may increase operating costs, background noise, and component size, forced convection is often undesirable. An alternative method of increasing $q_c$ is to increase $A_s$ by adding extended surfaces or fins. [Ref. 1:pp. 114-119]

In the design of electronic components, the disadvantages of forced convection often lead to the use of natural convection and fins to dissipate potentially destructive component produced heat. Fin design and optimization in a natural convective environment is the subject of this work.

Fin design theory may be examined by beginning with the single fin problem. The temperature profile differential equation can be developed for arbitrarily shaped fins and subsequently applied to fins of constant cross-
sectional area. The temperature distribution, fin tip temperature, heat dissipation, and fin efficiency equations are all derived here and the equations are then applied to the cylindrical spine and rectangular fin, which are two commonly employed fin designs.

Single fin optimization theory is explored for the cylindrical spine and rectangular fin. Two separate situations are examined. First, for a given volume or quantity of material, the fin dimensions can be optimized to maximize the heat transfer rate. Second, for a given heat transfer rate, the fin dimensions can be provided which will minimize the volume of material required.

Multiple fin heat transfer and optimization theories can also be developed for an array of symmetric isothermal rectangular fins. Here too, the equation for the heat dissipation can be derived. This leads to an optimization where the number and spacing of fins can be provided to give the maximum heat transfer rate from a wall of given dimensions. In addition, the number and spacing of fins can be optimized to maximize the heat transfer rate from each fin on a wall of given dimensions.

The single and multiple fin heat transfer and optimization equations are implemented in a computer program written in ADA and compiled for use on IBM compatible machines. The emphasis of the program is on ease of use. The program is menu driven and does not require any amplifying documentation. Knowledge of heat transfer theory is not required. When faced with a fin optimization problem, an electronic component designer is no longer forced to choose between conducting laborious heat transfer calculations or resorting to trial and error.
The computer program is illustrated by means of an actual fin array design problem. The increase in heat transfer rate resulting from a staggering of fin arrays, without any increase in materials or wall placement area, is also demonstrated.
II. SINGLE FIN HEAT TRANSFER THEORY

A. INTRODUCTION

In order to make the mathematical analysis of extended surfaces tractable, Murray [Ref. 2:p. A78] and Gardner [Ref. 3:p. 621] proposed several limiting assumptions. These are:

1. The heat flow and temperature distribution throughout the fin are independent of time i.e., the heat flow is steady.
2. The fin material is homogeneous and isotropic.
3. There are no heat sources in the fin itself.
4. The heat flow to or from the fin surface at any point is directly proportional to the temperature difference between the surface at that point and the surrounding fluid.
5. The thermal conductivity of the fin is constant.
6. The heat transfer coefficient is the same over all the fin surface.
7. The temperature of the surrounding fluid is uniform.
8. The temperature of the base of the fin is uniform.
9. The thickness is so small compared to its height that temperature gradients normal to the surface may be neglected.
10. The heat transferred through the outermost edge of the fin is negligible compared to that passing through the sides.

These assumptions serve to narrow the scope of the extended surface problem and are applicable in the analysis that follows. [Ref. 3:p. 324]

B. ARBITRARILY SHAPED FINS

1. Temperature Profile Differential Equation

Figure 2-1 is an example of an arbitrarily shaped fin of height \( b \), differential surface area \( dA \), and varying cross-sectional area \( A(x) \). The Fourier and Newton Laws are used to derive the differential equation for the
temperature profile of the fin shown in Figure 2-1. As fins are generally thin, \( b \) is assumed to be much greater than \( r \). Although the fin temperature varies with \( r \) and \( x \), the radial variation is small and assumed to be negligible. [Ref. 1:pp. 117-118]

Figure 2-1. Arbitrarily Shaped Fin of Varying Cross-Sectional Area

In Figure 2-1, Heat enters the control volume by conduction at a rate of \( q(x) \) and exits at a rate of \( q(x + dx) \). Heat is dissipated by convection through \( dA_r \) at a rate of \( dq_c \). [Ref. 1:p. 118]

Ignoring radiation and assuming no internal heat generation, the heat balance energy equation for the control volume can be written as

\[
q(x) = q(x + dx) + dq_c
\]  \hspace{1cm} (2-1)

Substituting for \( q(x+dx) \)

\[
q(x) = \left( q(x) + \frac{dq(x)}{dx} dx \right) + dq_c
\]  \hspace{1cm} (2-2)
and simplifying yields

\[-\frac{dq(x)}{dx} dx = dq_c\]  \hspace{1cm} (2-3)

Fourier's Law can be expressed as

\[q(x) = -kA(x) \frac{dT}{dx}\]  \hspace{1cm} (2-4)

where \(k\) is the thermal conductivity of the fin material and \(dT/dx\) is the temperature gradient. Differentiating Equation 2-4 with respect to \(x\) gives

\[-\frac{dq(x)}{dx} dx = -k \frac{d}{dx} \left(A(x) \frac{dT}{dx}\right)\]  \hspace{1cm} (2-5)

Newton's Law of Cooling can be written as

\[dq_c = h dA (T - T_\infty)\]  \hspace{1cm} (2-6)

Substituting Equations 2-5 and 2-6 into Equation 2-3 gives

\[k \frac{d}{dx} \left(A(x) \frac{dT}{dx}\right) dx = h dA (T - T_\infty)\]  \hspace{1cm} (2-7)

Then, differentiating and dividing both sides by \(k dx\) provides

\[\frac{dA(x)}{dx} \frac{dT}{dx} + A(x) \frac{d^2T}{dx^2} = \frac{h}{k} \frac{dA}{dx} (T - T_\infty)\]  \hspace{1cm} (2-8)

so that a simplification then gives

\[\frac{d^2T}{dx^2} + \frac{1}{A(x)} \frac{dA(x)}{dx} \frac{dT}{dx} - \frac{h}{kA(x)} \frac{dA}{dx} (T - T_\infty) = 0\]  \hspace{1cm} (2-9)

Equation 2-9 is the temperature profile differential equation for an arbitrarily shaped fin of varying cross-sectional area. [Ref. 1:pp. 118-119]
2. Temperature Distribution Equation

A general solution to Equation 2-9, is found for a fin with constant cross-sectional area, A. Uniform cross-sectional area means that \( A(x) = A \) (a constant) and permits the simplification

\[
\frac{dA(x)}{dx} = 0 \tag{2-10}
\]

and with this in Equation 2-9, the result is

\[
\frac{d^2T}{dx^2} - \frac{h}{kA} \frac{dA}{dx}(T - T_\infty) = 0 \tag{2-11}
\]

If the surface area is expressed in terms of the perimeter, \( P \)

\[
dA = P \, dx \tag{2-12}
\]

a substitution into Equation 2-11 yields

\[
\frac{d^2T}{dx^2} - \frac{hP}{kA}(T - T_\infty) = 0 \tag{2-13}
\]

and if a change of variables

\[
\theta = T - T_\infty \tag{2-14}
\]

is made, then

\[
T = \theta + T_\infty \tag{2-15}
\]

\[
\frac{dT}{dx} = \frac{d\theta}{dx} \tag{2-16}
\]

and

\[
\frac{d^2T}{dx^2} = \frac{d^2\theta}{dx^2} \tag{2-17}
\]

Use of these permits Equation 2-13 to be written as

\[
\frac{d^2\theta}{dx^2} - \frac{hP}{kA} \theta = 0 \tag{2-18}
\]

and if the parameter, \( m \), is introduced
\[ m \equiv \frac{h_P}{\sqrt{kA}} \quad (2-19) \]

then Equation 2-18 can be written as

\[ \frac{d^2 \theta}{dx^2} - m^2 \theta = 0 \quad (2-20) \]

The general solution to Equation 2-20 is

\[ \theta = C_1 \cosh(mx) + C_2 \sinh(mx) \quad (2-21) \]

where the arbitrary constants, \( C_1 \) and \( C_2 \), are evaluated from the boundary conditions

- At position \( x = 0 \), \( T = T_w \) and \( \theta = T_w - T_r \)

- At position \( x = b \), \( \frac{dT}{dx} = 0 \) and \( \frac{d\theta}{dx} = 0 \)

where \( T_w \) is the wall temperature. The second boundary condition is based on the earlier assumption that \( b \) is much greater than \( r \) so that the surface area at the fin tip is very small and that the heat dissipated at the tip is negligible. The heat convected away from the surface area at the tip of the fin is considered negligible. Applying the first boundary condition to Equation 2-21 yields

\[ \theta_w = C_1 \cdot 1 + C_2 \cdot 0 \quad (2-22) \]

so that

\[ C_1 = \theta_w \quad (2-23) \]

Then, a differentiation of Equation 2-21 gives

\[ \frac{d\theta}{dx} = \theta_w m \sinh(mx) + C_2 m \cosh(mx) \quad (2-24) \]

so that employment of the second boundary condition provides

\[ 0 = \theta_w m \sinh(mb) + C_2 m \cosh(mb) \quad (2-25) \]

and hence
\[ C_2 = -\frac{\theta_w \sinh(mb)}{\cosh(mb)} \quad (2-26) \]

After substitution for \( C_1 \) and \( C_2 \), Equation 2-21 becomes

\[ \frac{\theta}{\theta_w} = \frac{\cosh(mb) \cosh(mx) - \sinh(mb) \sinh(mx)}{\cosh(mb)} \quad (2-27) \]

Using a hyperbolic function identity in Equation 2-27 allows the representation

\[ \frac{\theta}{\theta_w} = \frac{\cosh \left[ mb \left( 1 - \frac{x}{b} \right) \right]}{\cosh(mb)} \quad (2-28) \]

and returning to \( \theta_w = T_w - T_\infty \)

\[ T = T_w + \frac{(T_w - T_\infty) \cosh \left[ mb \left( 1 - \frac{x}{b} \right) \right]}{\cosh(mb)} \quad (2-29) \]

Equation 2-29 gives the temperature profile for a fin of constant cross-sectional area. The temperature at any position \( x \) along the fin can be calculated using Equation 2-29. [Ref. 1:pp. 120-123]

3. **Fin Tip Temperature Equation**

Often, a value of interest is the temperature at the tip of the fin, \( T_{tip} \).

Substituting \( x = b \) into Equation 2-29 gives

\[ T_{tip} = T_w + \frac{(T_w - T_\infty)}{\cosh(mb)} \quad (2-30) \]

Equation 2-30 is the equation for the fin tip temperature for a fin of constant cross-sectional area. [Ref. 1:pp. 145-146]
4. **Heat Dissipation Equation**

The heat dissipation equation for a fin of constant cross-sectional area is derived using Fourier's Law and Equation 2-27. From Fourier's Law,

\[ q = -kA \frac{dT}{dx} \bigg|_{x=0} = -kA \frac{d\theta}{dx} \bigg|_{x=0} \quad (2-31) \]

Differentiation of Equation 2-27 yields

\[ \frac{d\theta}{dx} = \frac{\theta_w [m \cosh(mb) \sinh(mx) - m \sinh(mb) \cosh(mx)]}{\cosh(mb)} \quad (2-32) \]

and this may be put into Equation 2-31 to obtain

\[ q = -kA \theta_w \left[ \frac{m \cosh(mb) \sinh(mx) - m \sinh(mb) \cosh(mx)}{\cosh(mb)} \right] \bigg|_{x=0} \quad (2-33) \]

A simplification then yields

\[ q = \frac{kA \theta_w [m \sinh(mb)]}{\cosh(mb)} \quad (2-34) \]

and substituting for \( \theta_w \) gives

\[ q = kA m(T_w - T_\infty) \tanh(mb) \quad (2-35) \]

Equation 2-35 is the heat dissipation equation for a fin of constant cross-sectional area. [Ref. 1:p. 123]

5. **Fin Efficiency**

A common parameter employed in the design of finned surfaces is the fin efficiency, \( \eta \). The definition of \( \eta \) is the actual heat dissipated by the fin divided by that which would be dissipated if the fin operated throughout at the wall temperature. If Equation 2-35, which gives the actual dissipation, is
divided by Newton's Law of Cooling, which gives the ideal dissipation, the result is

\[ \eta = \frac{kAm(T_w - T_i) \tanh(mb)}{hA_s(T_w - T_i)} \]  (2-36)

Hence

\[ \eta = \frac{kAm \tanh(mb)}{hA_s} \]  (2-37)

and

\[ \eta = \frac{kAm^2 \tanh(mb)}{mHA_s} \]  (2-38)

and with \( m^2 = hP/kA \) by Equation 2-19

\[ \eta = \frac{kAhP \tanh(mb)}{kAmhA_s} \]  (2-39)

Thus

\[ \eta = \frac{P \tanh(mb)}{mA_s} \]  (2-40)

But \( A_s = Pb \), and this simplification gives the final expression for the fin of constant cross-section.

\[ \eta = \frac{\tanh(mb)}{mb} \]  (2-41)

Equation 2-41 provides the efficiency of a fin with constant cross-sectional area. [Ref. 1:p. 125]
C. SPECIFIC FIN CONFIGURATIONS

1. Cylindrical Spine

As shown in Figure 2-2, a cylindrical spine is essentially a bar of height \( b \) and diameter \( d \) attached to the surface to be cooled. As the fin has a constant cross-sectional area, the equations derived in the previous section apply to the cylindrical spine. [Ref. 1:p. 120]

![Figure 2-2. Cylindrical Spine](image)

For the cylindrical spine, the perimeter and area are written as

\[
P = \pi d
\]

(2-42)

and

\[
A = \frac{\pi d^2}{4}
\]

(2-43)

Hence, Equation 2-19 becomes
2. Longitudinal Fin of Rectangular Profile

A commonly encountered fin configuration is that of the longitudinal fin of rectangular profile. Figure 2-3 is an example of a rectangular fin of height \( b \), length \( L \), and width \( \delta \). The equations derived for fins of constant cross-sectional area are applicable to the rectangular fin configuration. [Ref. 1:p. 120]

\[ m = \sqrt{\frac{4h}{kd}} \]  

Figure 2-3. Rectangular Fin

For the longitudinal fin of rectangular profile, the perimeter and area are
\[ P = 2\delta + 2L \quad (2-45) \]

and

\[ A = L\delta \quad (2-46) \]

As rectangular fins are traditionally thin, the following simplification is made [Ref. 1:p. 137].

\[ P \equiv 2L \quad (2-47) \]

Substituting Equations 2-46 and 2-47 into Equation 2-19 gives

\[ m = \sqrt{\frac{2h}{k\delta}} \quad (2-48) \]
A. INTRODUCTION

Single fin optimization theory can be divided into two categories. In the first category the fin shape is known. Two commonly employed fin shapes, the cylindrical spine and the rectangular fin, may be examined from two different perspectives. Either the dimensions of the fin are optimized to yield the maximum heat transfer rate from a given volume or, for a given heat transfer rate, the fin dimensions are optimized to minimize the required volume of material. The second category is based on the determination of an optimal fin shape. Fin shapes are found which minimize the volume of material required to obtain a given heat transfer rate. Curved fins are commonly produced in the shape optimization problem. As curved fins are difficult and expensive to manufacture, the shape optimization problem will not be addressed. [Ref. 5:p. 155]

B. CYLINDRICAL SPINE

1. Maximum Heat Transfer for a Given Volume

Substituting Equations 2-43 and 2-44 into Equation 2-35 yields

\[
q = k \frac{\pi d^2}{4} \sqrt{\frac{4h}{kd}} (T_w - T_a) \tanh \left( \sqrt{\frac{4h}{kd} b} \right)
\] (3-1)

If
\[ \beta = b\sqrt{\frac{4h}{kd}} \]  

then

\[ q = \left[ \frac{k\pi d^2(T_w - T_\infty)}{4b} \right] \beta \tanh(\beta) \]  

and for a cylinder of volume \( V \)

\[ V = \frac{\pi d^2 b}{4} \]  

and

\[ b = \frac{4V}{\pi d^2} \]  

or

\[ d = \sqrt{\frac{4V}{\pi b}} \]  

Substituting Equation 3-5 into Equation 3-3 yields

\[ q = \left[ \frac{k\pi d^4(T_w - T_\infty)}{16V} \right] \beta \tanh(\beta) \]  

Then, taking the derivative with respect to \( d \), simplifying, and setting it equal to zero leads to the following transcendental equation.

\[ 10\beta = 3 \sinh(2\beta) \]  

with a solution which can be determined by trial and error

\[ \beta = 0.9193 \]  

Substitution of this value of \( \beta \) into Equation 3-2 produces

\[ 0.9193 = b\sqrt{\frac{4h}{kd}} \]
Expressing $b$ by Equation 3-5 gives

$$0.9193 = \frac{4V}{\pi d^2} \sqrt{\frac{4h}{kd}} \quad (3-11)$$

Hence, the optimized value for the diameter, $d$, is

$$d_{\text{opt}} = 1.5031 \left( \frac{hV^2}{k} \right)^{1/5} \quad (3-12)$$

and substitution of Equation 3-6 into Equation 3-10 yields

$$0.9193 = b \sqrt{\frac{4h}{k}} \left( \frac{4V}{\pi b} \right)^{1/2} \quad (3-13)$$

Thus, the optimized value for the height, $b$, is

$$b_{\text{opt}} = 0.5636 \left( \frac{Vk^2}{h^2} \right)^{1/5} \quad (3-14)$$

Equations 3-12 and 3-14 specify the optimal dimensions of a cylindrical spine to achieve the maximum heat transfer rate for a given volume. [Ref. 5:p. 158]

2. **Minimum Volume for a Given Heat Transfer**

Solve Equation 3-10 for $b$

$$b = 0.9193 \sqrt{\frac{k d}{4h}} \quad (3-15)$$

and then substitute Equations 3-15 and 3-9 into Equation 3-3 to obtain

$$q = \left[ \frac{k \pi d^2 (T_w - T_\infty)}{4(0.9193) \sqrt{kd} \sqrt{4h}} \right] 0.9193 \tanh(0.9193) \quad (3-16)$$
Hence

\[ d_{opt} = 0.9165 \left( \frac{q^2}{hk(T_w - T_\infty)^2} \right)^{\frac{1}{3}} \]  

(3-17)

The solution of the transcendental, Equation 3-10, is for \( d \)

\[ d = \frac{4hb^2}{k(0.9193)^2} \]  

(3-18)

and substituting Equations 3-18 and 3-9 into Equation 3-3 yields

\[ q = \left[ \frac{16h^2b^4k\pi(T_w - T_\infty)}{4bk^2(0.9193)^4} \right] 0.9193 \tanh(0.9193) \]  

(3-19)

Thus

\[ b_{opt} = 0.4400 \left( \frac{qk}{h^2(T_w - T_\infty)} \right)^{\frac{1}{3}} \]  

(3-20)

Equations 3-17 and 3-20 specify the optimal dimensions of a cylindrical spine to achieve the minimum volume for a given heat transfer rate. [Ref. 5: p. 158]

C. RECTANGULAR FIN

1. Maximum Heat Transfer for a Given Volume and Length

Substituting Equations 2-46 and 2-48 into Equation 2-35 gives

\[ q = k\delta L \sqrt{\frac{2h}{k\delta}} (T_w - T_\infty) \tanh \left( b \sqrt{\frac{2h}{k\delta}} \right) \]  

(3-21)

For a rectangular fin

\[ V = \delta L b \]  

(3-22)
or

\[ b = \frac{V}{\delta L} \]  \hspace{1cm} (3-23)

and

\[ \delta = \frac{V}{bL} \]  \hspace{1cm} (3-24)

Substituting Equation 3-23 into Equation 3-21 yields

\[ q = k\delta L \sqrt{\frac{2h}{k\delta}} (T_w - T_\infty) \tanh \left( \frac{V}{\delta L} \sqrt{\frac{2h}{k\delta}} \right) \]  \hspace{1cm} (3-25)

and, once more, making a change of variables

\[ U = \frac{V}{L} \sqrt{\frac{2h}{k\delta^3}} \]  \hspace{1cm} (3-26)

gives

\[ q = k\delta L \sqrt{\frac{2h}{k\delta}} (T_w - T_\infty) \tanh(U) \]  \hspace{1cm} (3-27)

Taking the derivative with respect to \( \delta \), simplifying, and setting the result equal to zero, yields the transcendental equation

\[ 6U = \sinh(2U) \]  \hspace{1cm} (3-28)

A trial and error solution gives

\[ U = 1.4192 \]  \hspace{1cm} (3-29)

and substituting this result into Equation 3-26 gives

\[ 1.4192 = \frac{V}{L} \sqrt{\frac{2h}{k\delta^3}} \]  \hspace{1cm} (3-30)

This shows that the optimized value for the width, \( \delta \), is
\[ \delta_{opt} = 0.9977 \left( \frac{V^2 h}{L k} \right)^{1/3} \]  

(3-31)

Substituting Equation 3-24 into Equation 3-30 yields

\[ 1.4192 = \frac{V}{L} \sqrt{\frac{2h}{k \left( \frac{V}{bL} \right)^3}} \]  

(3-32)

and

\[ b_{opt} = 1.0023 \left( \frac{V k}{L h} \right)^{1/3} \]  

(3-33)

Equations 3-31 and 3-33 specify the optimal dimensions of a rectangular fin to achieve the maximum heat transfer rate for a given volume and length. [Ref. 5:p. 156]

2. Minimum Volume for a Given Heat Transfer and Length

Substituting Equation 3-29 into Equation 3-27 gives

\[ q = k \delta L \sqrt{\frac{2h}{k \delta} (T_w - T_\infty) \tanh(1.4192)} \]  

(3-34)

and from this the optimum \( \delta \) is obtained

\[ \delta_{opt} = \frac{0.6321}{hk} \left( \frac{q}{L(T_w - T_\infty)} \right)^2 \]  

(3-35)

Then, substituting Equation 3-24 into Equation 3-35 yields

\[ \frac{V}{bL} = \frac{0.6321}{hk} \left( \frac{q}{L(T_w - T_\infty)} \right)^2 \]  

(3-36)

and solving Equation 3-33 for \( V \) provides
\[ V = \frac{b^3 L h}{(1.0023)^3 k} \] 

(3-37)

Then, substituting this result into Equation 3-36 produces

\[ \frac{b^3 L h}{(1.0023)^3 k b L} = \frac{0.6321}{h k} \left( \frac{q}{L(T_w - T_i)} \right)^2 \] 

(3-38)

and the optimum value of the fin height, \( b \) is

\[ b_{\text{opt}} = 0.7978 \frac{q}{L h(T_w - T_i)} \] 

(3-39)

Equations 3-35 and 3-39 specify the optimal dimensions of a rectangular fin to achieve the minimum volume for a given heat transfer rate and length. [Ref. 5:p. 156]
IV. MULTIPLE FIN HEAT TRANSFER THEORY

A. INTRODUCTION

In their 1984 landmark paper, Bar-Cohen and Rohsenow described the heat transfer and optimization equations for an array of rectangular fins. However, they did not provide a design procedure with which to use their data to formulate optimum arrays. [Ref. 6:pp. 116-123]

B. ARRAY OF RECTANGULAR FINS

1. Heat Dissipation Equation

In Figure 4-1, adjacent fins form a channel. The channel Rayleigh number, \( Ra' \) is defined as

\[
Ra' = \frac{\rho^2 g \beta c_p z^4 (T_w - T_a)}{\mu L k_f}
\]  

(4-1)

where

- \( \rho \) = density of the surrounding fluid, kg/m\(^3\)
- \( g \) = gravitational acceleration, m/s\(^2\)
- \( \beta \) = volumetric coefficient of thermal expansion, 1/°K
- \( c_p \) = specific heat of the surrounding fluid, J/kg°C
- \( z \) = clear spacing, m
- \( T_w \) = wall temperature, °C
- \( T_a \) = ambient temperature, °C
- \( \mu \) = dynamic viscosity of the surrounding fluid, kg/m•s
The volumetric coefficient of thermal expansion is

\[ \beta = \frac{1}{T_{\text{avg}} + 460^\circ R} \]  \hspace{1cm} (4-2)

or, in SI units

\[ \beta = \frac{1}{T_{\text{avg}} + 273.15K} \]  \hspace{1cm} (4-3)
where

\[
T_{\text{avg}} = \frac{T_w + T_\infty}{2}
\]  

(4-4)

The dynamic viscosity is

\[
\mu = \frac{\nu \rho}{g_c}
\]  

(4-5)

where \( \nu \) is the kinematic viscosity of the surrounding fluid and

\[
g_c = 32.2 \frac{\text{lbm} \cdot \text{ft}}{\text{lbf} \cdot \text{s}^2}
\]  

(4-6)

or, in SI units

\[
g_c = 1.0 \frac{\text{kg} \cdot \text{m}}{\text{N} \cdot \text{s}^2}
\]  

(4-7)

If the fins are assumed to be symmetric and isothermal

\[
Nu_0 = \left[ \frac{576}{(Ra')^2} + \frac{2.873}{\sqrt{Ra'}} \right]^{-\frac{1}{2}}
\]  

(4.8)

where \( Nu_0 \) is the channel Nusselt number. Thus, the heat transfer coefficient is for \( n \) fins attached to the wall

\[
h = \frac{nu_0 k_f}{z}
\]  

(4-9)

For a rectangular fin, the parameter \( m \) is given by Equation 2-48. The fin efficiency, \( \eta \), is given by Equation 2-41. The surface area of the wall that is open for heat transfer, \( A_w \), is

\[
A_w = (L_w \delta_w) - (nL \delta)
\]  

(4-10)

and the combined heat transfer area for all fins, \( A_{\text{fins}} \), is

\[
A_{\text{fins}} = n(2bL)
\]  

(4-11)
Hence, the total area available for heat transfer, $A_{total}$, is

$$A_{total} = A_w + (\eta A_{fins})$$  \hspace{1cm} (4-12)

and according to Newton's Law of Cooling

$$q = hA_{total}(T_w - T_m)$$  \hspace{1cm} (4-13)

Equation 4-13 is the heat dissipation equation for an array of symmetric, isothermal rectangular fins. [Ref. 6:pp. 116-119]
V. MULTIPLE FIN OPTIMIZATION THEORY

A. ARRAY OF RECTANGULAR FINS

1. Maximum Heat Transfer for a Given Wall Area

In an array of symmetric, isothermal rectangular fins, the heat transfer rate of each fin decreases as fin spacing decreases. Yet, a reduction in fin spacing allows for a greater number of fins to be placed on a wall of given dimensions. An optimal fin spacing exists which maximizes the heat transfer rate for a given wall area. [Ref. 6:p. 120]

As the wall will be fully populated with fins to maximize \( q \), \( A_{\text{fins}} \) is assumed to be much greater than \( A_w \). Replacing \( A_{\text{total}} \) in Equation 4-13 with \( A_{\text{fins}} \),

\[
q = hA_{\text{fins}}(T_w - T_\infty)
\]  

(5-1)

and substitution of this into Equations 4-9 and 4-11 gives

\[
q = \frac{\mathit{Nu}_0 k_f}{z} N(2bL)(T_w - T_\infty)
\]  

(5-2)

To fully populate the wall with fins

\[
N = \frac{\delta_w}{(z + \delta)}
\]  

(5-3)

The value obtained for \( N \) from Equation 5-3 must be truncated to produce an integral number of fins. Substitution of Equations 4-8 and 5-3 into Equation 5-2 gives
and then making a change of variables,

\[ p = \frac{Ra'}{z^4} = \frac{\rho^2 \beta c_p (T_w - T_\infty)}{\mu L k_f} \]  

one obtains

\[ q = \left[ \frac{576}{(Ra')^2} + \frac{2.873}{\sqrt{Ra'}} \right]^{1/2} \frac{k_f \delta_w (2bL)(T_w - T_\infty)}{z(z + \delta)} \]  

or

\[ \frac{q}{2k_f \delta_w bL(T_w - T_\infty)} = \frac{1}{z(z + \delta)\left[ \frac{576}{P^2 z^8} + \frac{2.873}{\sqrt{P} z^2} \right]^{1/2}} \]  

Taking the derivative \( dq/dz \), simplifying, and then setting the result equal to zero gives

\[ 2z + 3\delta - 0.005 P^{3/2} z^7 = 0 \]  

If the width is assumed to be negligible,

\[ z_{\text{opt}} = \frac{2.714}{P^{3/4}} \]  

Equation 5-9 gives the optimal fin spacing to maximize the heat transfer rate for a wall of given dimensions. [Ref. 6:p. 120]
2. **Maximum Heat Transfer from Each Fin**

It is often desired to maximize the heat transfer rate from each fin in an array of symmetric, isothermal rectangular fins. Although an infinite fin spacing is theoretically required, setting $N_u$ equal to 99% of the isolated fin value leads to

$$z_{max} = \frac{4.64}{P^{1/4}}$$

(5-10)

Equation 5-10 gives the optimal fin spacing to maximize the heat transfer rate from each fin on a wall of given dimensions. Substituting the $z_{max}$ of Equation 5-3 provides the value for the maximum number of fins. [Ref. 6:p. 120]

The value of $z_{max}$ is approximately double that of the boundary layer thicknesses along each of the surfaces at the channel exit. The value $z_{opt}$ coincides with approximately 1.2 boundary layer thicknesses. [Ref. 6:p. 120]
VI. COMPUTER PROGRAM DEMONSTRATION

A. INTRODUCTION

To provide a demonstration of the computer program, a fin array design example involving a wall of given dimensions populated by two different arrangements of rectangular fins, is analyzed and the results are compared.

B. SINGLE NON-STAGGERED FIN ARRAY

In the first arrangement, the wall is populated by a single, non-staggered fin array. The dimensions of the array are shown in Figure 6-1. The computer program will optimize fin spacing to maximize the array's heat transfer rate.

Figure 6-1. Single Non-Staggered Fin Array
The program is initiated by the typing the following command at the DOS prompt.

```
C:\>finopt J
```

After the introduction and continuation screens, Figure 6-2 is presented.

![Figure 6-2. Unit System Menu](image)

As the dimensions in Figure 6-1 are in centimeters, the enter key (J) is pressed to select the SI unit system. Figure 6-3 is shown on the screen.

![Figure 6-3. Fin Optimization Menu](image)
Enter is pressed to select the multiple fin problem. The drawing in Figure 6-4 is then shown on the screen.

Figure 6-4 graphically shows the nomenclature for the parameters that will be requested by the program so that the optimization and heat transfer calculations can be instituted. After pressing any key, the menu in Figure 6-5 is displayed. Enter is pressed to choose the third selection listed in the menu.
Choose Type of Optimization
1) No Optimization
2) Maximum heat transfer capability from each fin in a given area
3) Maximum heat transfer capability for a given area
Input your selection: [3]
Press enter to accept default

Figure 6-5. Type of Optimization Menu

Figure 6-6 illustrates the format used for the input of the parameters needed by the optimization and heat transfer equations.

Required Inputs
All values must be inputted as floats. Examples: 5.0 or 2.0E-3
Do not input 0.8 as .8 !!!

Length of the fin placement area = 2.000E+01 cm
Width of the fin placement area (cm) =
Press enter to accept default or any other key to enter new value
New value = 25.0

Figure 6-6. Request for Inputs

Once the user has supplied the required inputs, the input-output summary screens in Figure 6-7 are presented.
The user can print the summary information in Figure 6-7 by performing a DOS screen dump, simultaneously pressing the Shift and Print Screen keys. The information can be imported into a word processor for editing by running the program under Microsoft Windows and using its clipboard screen capture.
functions [Ref. 7:pp. 248-250]. Pressing any key will produce the drawing in Figure 6-8.

![Drawing of input-output summary](image)

**Figure 6-8. Input-Output Summary Drawing**

The drawing in Figure 6-8 graphically displays the outputs of the optimization calculations.

C. **TWO STAGGERED FIN ARRAYS**

In the second arrangement, the wall is populated by two staggered fin arrays. The dimensions of the arrays are shown in Figure 6-9. Each array will
be dealt with separately. The heat transfer rate for a single array will be calculated and doubled to find the total heat transfer rate for the wall of given dimensions.

Figure 6-9. Two Staggered Fin Arrays
1. Same Spacing and Number of Fins

First, the staggered fin problem is examined using the spacing and number of fins found in the previous non-staggered fin example. Pressing any key after viewing the drawing in Figure 6-8 results in the continuation menu in Figure 6-10.

![Continuation Menu](image)

Pressing Enter will allow the user to continue in the program. All the defaults are set to the values inputted by the user in the preceding problem to facilitate rapid sensitivity analysis. After proceeding through the screens in Figures 6-2, 6-3, and 6-4, the first item, "No Optimization," is chosen from the menu in Figure 6-5. This option will allow the user to input the fin spacing and number of fins rather than conducting an optimization of those values. The parameters required by the heat transfer equations are inputted in the format demonstrated in Figure 6-6. Once the user has supplied the required inputs, the input-output summary screens in Figure 6-11 are presented.
### Inputs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length of the fin placement area</td>
<td>1.0000E+01 cm</td>
</tr>
<tr>
<td>Width of the fin placement area</td>
<td>2.5000E+01 cm</td>
</tr>
<tr>
<td>Length of each fin</td>
<td>1.0000E+01 cm</td>
</tr>
<tr>
<td>Height of each fin</td>
<td>5.0000E+00 cm</td>
</tr>
<tr>
<td>Width of each fin</td>
<td>2.0000E-01 cm</td>
</tr>
<tr>
<td>Spacing between fins</td>
<td>7.0603E-01 cm</td>
</tr>
<tr>
<td>Number of fins</td>
<td>2.7000E+01 fins</td>
</tr>
<tr>
<td>Density of surrounding fluid</td>
<td>1.1770E+00 kg/m^3</td>
</tr>
<tr>
<td>Specific heat of surrounding fluid</td>
<td>1.0057E+03 J/(kg*deg-K)</td>
</tr>
<tr>
<td>Thermal conductivity of material, k</td>
<td>2.1000E+02 W/(m*deg-K)</td>
</tr>
<tr>
<td>Thermal conductivity of surrounding fluid, k</td>
<td>2.6240E-02 W/(m*deg-K)</td>
</tr>
<tr>
<td>Kinematic viscosity of surrounding fluid</td>
<td>1.5680E-05 m^2/s</td>
</tr>
<tr>
<td>Ambient Temperature</td>
<td>2.5000E+01 deg-C</td>
</tr>
<tr>
<td>Wall Temperature</td>
<td>7.5000E+01 deg-C</td>
</tr>
</tbody>
</table>

### Outputs

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Heat transferred away by the fins, q</td>
<td>9.2229E+01 W</td>
</tr>
<tr>
<td>The fin efficiency</td>
<td>9.7490E-01</td>
</tr>
<tr>
<td>The temperature at the tip of the fins</td>
<td>7.3120E+01 deg-C</td>
</tr>
<tr>
<td>Channel Rayleigh number</td>
<td>1.0851E+02</td>
</tr>
<tr>
<td>Channel Nusselt number</td>
<td>1.7549E+00</td>
</tr>
</tbody>
</table>

**Figure 6-11. Input-Output Summary**

The value for the single array heat transfer rate, $q$, is doubled to obtain $q_{total}$ for the given wall area.

37
\[ q_{\text{total}} = 2(92.229) = 184.46 \text{ W} \quad (6-1) \]

The improvement is

\[ \frac{184.46 - 138.16}{138.16} = 33.5\% \quad (6-2) \]

and one may conclude that staggering the arrays produces a 33.5% increase in the heat transfer rate, without any change in the number of fins, spacing, or materials required. Unfortunately the machining required to produce staggered fin arrays may be expensive.

2. Optimization of Spacing and Number of Fins

The increase in the heat transfer rate shown in Equation 6-2 can be further enhanced by optimizing the number of fins and spacing in the individual fin arrays. Again the third item is chosen from the menu in Figure 6-5. The input-output summary screens produced are shown in Figure 6-12. The value obtained for \( q \) is again doubled to obtain \( q_{\text{total}} \) for the given wall dimensions.

\[ q_{\text{total}} = 2(92.944) = 185.89 \text{ W} \quad (6-3) \]

and

\[ \frac{185.89 - 138.16}{138.16} = 34.5\% \quad (6-4) \]

and, in addition

\[ \frac{185.89 - 184.46}{184.46} = 0.775\% \quad (6-5) \]

The percentage increase found in Equation 6-5 is small, since the non-staggered array of fins had already been previously optimized.
**Inputs => Outputs**

**Inputs**

Length of the fin placement area = 1.0000E+01 cm
Width of the fin placement area = 2.5000E+01 cm
Length of each fin = 1.0000E+01 cm
Height of each fin = 5.0000E+00 cm
Width of each fin = 2.0000E+01 cm
Density of surrounding fluid = 1.1770E+00 kg/m^3
Specific heat of surrounding fluid = 1.0057E+03 J/(kg*deg-K)
Thermal conductivity of material, k = 2.1000E+02 W/(m*deg-K)
Thermal conductivity of surrounding fluid, k = 2.6240E-02 W/(m*deg-K)
Kinematic viscosity of surrounding fluid = 1.5680E-05 m^2/s
Ambient Temperature = 2.5000E+01 deg-C
Wall Temperature = 7.5000E+01 deg-C

**Outputs**

Heat transferred away by the fins, q = 9.2944E+01 W
Spacing between fins = 5.9370E-01 cm
Number of fins = 3.1000E+01 fins
The fin efficiency = 9.7770E-01
The temperature at the tip of the fins = 7.3329E+01 deg-C
Channel Rayleigh number = 5.4255E+01
Channel Nusselt number = 1.3066E+00

Figure 6-12. Input-Output Summary
VII. CONCLUSION

In the design of electronic equipment, a proper heat sink configuration is essential. As the design example in the previous section demonstrates, the computer program developed for this thesis can serve to greatly simplify and accelerate the fin design process. The program should prove to be valuable tool to electronic component designers, especially those with a limited background in heat transfer and fin optimization theory. A listing of the source code for the computer program is included in the Appendix.
APPENDIX

SOURCE CODE FOR THE COMPUTER PROGRAM

-- Title : EXTENDED SURFACE HEAT SINKS FOR ELECTRONIC COMPONENTS:
-- Author : John Reynold Gensure
-- Date : June 1992

with   TEXT_IO, COMMON_DISPLAY_TYPES, TTY, CURSOR, VIDEO,
       FINOPT_SINGLE, FINOPT_MULTIPLE, FINOPT_PICTURES,
       FINOPT_DRAWINGS;
use    TEXT_IO, COMMON_DISPLAY_TYPES, FINOPT_SINGLE, FINOPT_MULTIPLE;

procedure FINOPT is
   PAUSE, CONTINUE, UNITS, FIN_SING_MULT, FIN_CYL_RECT,
   FINOPT_TYP, DEFAULT_KEY : INTEGER;
   SPACING_ENGLISH, DENSITY_ENGLISH, SPECIFIC_HEAT_ENGLISH,
   NU_ENGLISH, K_FLUID_ENGLISH, WALL_LENGTH_ENGLISH,
   WALL_WIDTH_ENGLISH, DIAMETER_ENGLISH, HEIGHT_ENGLISH,
   WIDTH_ENGLISH, LENGTH_ENGLISH, VOLUME_ENGLISH, H_ENGLISH,
   K_ENGLISH, T_AMBIENT_ENGLISH, T_WALL_ENGLISH, Q_ENGLISH,
   SPACING_SI, DENSITY_SI, SPECIFIC_HEAT_SI, NU_SI,
   K_FLUID_SI, WALL_LENGTH_SI, WALL_WIDTH_SI, DIAMETER_SI,
   HEIGHT_SI, WIDTH_SI, LENGTH_SI, VOLUME_SI, H_SI, K_SI,
   T_AMBIENT_SI, T_WALL_SI, Q_SI, SPACING, DENSITY,
   SPECIFIC_HEAT, NU, K_FLUID, WALL_LENGTH, WALL_WIDTH,
   DIAMETER, HEIGHT, WIDTH, LENGTH, VOLUME, H, K, T_AMBIENT,
   T_WALL, Q, CONVERT_DIST, CONVERT_TEMP, GRAVITY, G_C,
   AREA_PROFILE, NUM_FINS, NUM_FINS_ENGLISH, NUM_FINS_SI : FLOAT;

   PI : constant := 3.14159_26535_89793_23846_26433_83279_50288_41972;
   SPACING_UNITS, WALL_LENGTH_UNITS,
   WALL_WIDTH_UNITS, DIAMETER_UNITS, HEIGHT_UNITS,
   WIDTH_UNITS, LENGTH_UNITS : STRING(1..2);
   VOLUME_UNITS, NUM_FINS_UNITS : STRING(1..4);
   T_UNITS : STRING(1..5);
   NU_UNITS, Q_UNITS : STRING(1..6);
begin

-- Introduction Page

VIDEO.SET COLOR_PALETTE (BLUE);
FINOPTPICTURES.THESIS_MSG;

-- Output Program Name

FINOPTPICTURES.FINOPT_MSG;

-- Initialize Variables English System

SPACING_ENGLISH := 0.5;
-- Spacing (in)
DENSITY_ENGLISH := 0.05928;
-- Density of surrounding fluid (lbm/ft^3)
SPECIFIC_HEAT_ENGLISH := 0.2404;
-- Specific heat of surrounding fluid (BTU/(lbm*deg-R))
NU_ENGLISH := 19.1774E-5;
-- Kinematic viscosity of surrounding fluid (ft^2/s)
K_FLUID_ENGLISH := 0.01608;
-- Thermal conductivity of surrounding fluid (BTU/(hr*ft*deg-R))
WALL_LENGTH_ENGLISH := 6.0;
-- Length of multi-fin placement area (in)
WALL WIDTH ENGLISH := 9.0;
-- Width of multi-fin placement area (in)
DIAMETER ENGLISH := 0.3125;
-- Diameter (in)
HEIGHT ENGLISH := 4.5;
-- Height (in)
WIDTH ENGLISH := 0.0625;
-- Width (in)
LENGTH ENGLISH := 2.25;
-- Length (in)
VOLUME ENGLISH := 0.3451;
-- Volume (in^3)
H ENGLISH := 1.0;
-- Convection heat transfer coefficient (BTU/(hr*ft^2*deg-R))
K ENGLISH := 24.8;
-- Thermal conductivity (BTU/(hr*ft*deg-R))
T AMBIENT ENGLISH := 70.0;
-- Ambient Temperature (deg-F)
T WALL ENGLISH := 200.0;
-- Wall temperature (deg-F)
Q ENGLISH := 3.13;
-- Heat transferred (BTU/hr)

---------------------------------------------------------------
-- Initialize Variables SI System
---------------------------------------------------------------

SPACING SI := 1.04;
-- Spacing (cm)
DENSITY SI := 1.177;
-- Density of surrounding fluid (kg/m^3)
SPECIFIC HEAT SI := 1005.7;
-- Specific heat of surrounding fluid (J/(kg*deg-K))
NU SI := 1.568E-5;
-- Kinematic viscosity of surrounding fluid (m^2/s)
K FLUID SI := 0.02624;
-- Thermal conductivity of surrounding fluid (W/(m*deg-K))
WALL LENGTH SI := 9.0;
-- Length of multi-fin placement area (in)
WALL WIDTH SI := 22.14;
-- Width of multi-fin placement area (in)
DIAMETER SI := 0.7;
-- Diameter (cm)
HEIGHT SI := 4.7;
-- Height (cm)
WIDTH SI := 0.18;
-- Width (cm)
LENGTH SI := 9.0;
-- Length (cm)
VOLUME SI := 16.3516;
-- Volume (cm^3)
H SI := 7.0;
-- Convection heat transfer coefficient (W/(m^2*deg-K))
K SI := 236.0;
-- Thermal conductivity (W/(m*deg-K))
T AMBIENT SI := 20.0;
-- Ambient temperature (deg-C)
T_WALL_SI := 60.0;
-- Wall temperature (deg-C)
Q_SI := 5.902;
-- Heat transferred (W)

Continue?

loop
TTY.CLEAR_SCREEN;
TTY.PUT ( 6, 21, " ", YELLOW, CYAN);
TTY.PUT ( 7, 21, " ", YELLOW, CYAN);
TTY.PUT ( 8, 21, " ", YELLOW, CYAN);
TTY.PUT ( 8, 55, " ", YELLOW, CYAN);
TTY.PUT ( 9, 21, " ", YELLOW, CYAN);
TTY.PUT ( 9, 55, " ", YELLOW, CYAN);
TTY.PUT (10, 21, " ", YELLOW, CYAN);
TTY.PUT (10, 55, " ", YELLOW, CYAN);
TTY.PUT (11, 21, " ", YELLOW, CYAN);
TTY.PUT (11, 55, " ", YELLOW, CYAN);
TTY.PUT (12, 21, " ", YELLOW, CYAN);
TTY.PUT (12, 55, " ", YELLOW, CYAN);
TTY.PUT (13, 21, " ", YELLOW, CYAN);
TTY.PUT (13, 55, " ", YELLOW, CYAN);
TTY.PUT (14, 21, " ", YELLOW, CYAN);
TTY.PUT (14, 55, " ", YELLOW, CYAN);
TTY.PUT (15, 21, " ", YELLOW, CYAN);
TTY.PUT (15, 55, " ", YELLOW, CYAN);
TTY.PUT (16, 21, " ", YELLOW, CYAN);
TTY.PUT (16, 55, " ", YELLOW, CYAN);
TTY.PUT (17, 21, " ", YELLOW, CYAN);
TTY.PUT (17, 55, " ", YELLOW, CYAN);
TTY.PUT (18, 21, " ", YELLOW, CYAN);
TTY.PUT (19, 21, " ", YELLOW, CYAN);
TTY.PUT ( 8, 24, " Do you wish to continue? ", YELLOW, RED);
TTY.PUT ( 9, 24, " ", BRIGHT_WHITE, RED);
TTY.PUT (10, 24, " 1) Exit to DOS ", YELLOW, RED);
TTY.PUT (11, 24, " 2) Continue ", YELLOW, RED);
TTY.PUT (12, 24, " ", YELLOW, RED);
TTY.PUT (13, 24, " ", YELLOW, RED);
TTY.PUT (14, 24, " ", YELLOW, RED);
TTY.PUT (15, 24, " Input your selection: [2] ", YELLOW, RED);
TTY.PUT (16, 24, " Press enter to accept default ", BRIGHT_WHITE, RED);
TTY.PUT (17, 24, " ", YELLOW, RED);
CURSOR.SET_SIZE(13,13);
loop
CURSOR.MOVE (15, 50);
TTY.GET (CONTINUE, CHAR);
if CONTINUE = 2 or CONTINUE = 3 or CONTINUE = 28 then
exit;

44
else
    TTY.PUT (21, 24, " Improper input, please reenter ",
             BLUE, CYAN);
    end if;
end loop;
CURSOR.INHIBIT;

Do Not Continue

exit when CONTINUE = 2;

---------------------------

Continue With Program

---------------------------

Select Unit System

---------------------------

TTY.CLEAR_SCREEN;
TTY.PUT ( 6, 21, " ",
         YELLOW, CYAN);
TTY.PUT ( 7, 21, " ",
         YELLOW, CYAN);
TTY.PUT ( 8, 21, " ",
         YELLOW, CYAN);
TTY.PUT ( 8, 55, " ",
         YELLOW, CYAN);
TTY.PUT ( 9, 21, " ",
         YELLOW, CYAN);
TTY.PUT ( 9, 55, " ",
         YELLOW, CYAN);
TTY.PUT (10, 21, " ",
         YELLOW, CYAN);
TTY.PUT (10, 55, " ",
         YELLOW, CYAN);
TTY.PUT (11, 21, " ",
         YELLOW, CYAN);
TTY.PUT (11, 55, " ",
         YELLOW, CYAN);
TTY.PUT (12, 21, " ",
         YELLOW, CYAN);
TTY.PUT (12, 55, " ",
         YELLOW, CYAN);
TTY.PUT (13, 21, " ",
         YELLOW, CYAN);
TTY.PUT (13, 55, " ",
         YELLOW, CYAN);
TTY.PUT (14, 21, " ",
         YELLOW, CYAN);
TTY.PUT (14, 55, " ",
         YELLOW, CYAN);
TTY.PUT (15, 21, " ",
         YELLOW, CYAN);
TTY.PUT (15, 55, " ",
         YELLOW, CYAN);
TTY.PUT (16, 21, " ",
         YELLOW, CYAN);
TTY.PUT (16, 55, " ",
         YELLOW, CYAN);
TTY.PUT (17, 21, " ",
         YELLOW, CYAN);
TTY.PUT (17, 55, " ",
         YELLOW, CYAN);
TTY.PUT (18, 21, " ",
         YELLOW, CYAN);
TTY.PUT (19, 21, " ",
         YELLOW, CYAN);
TTY.PUT ( 8, 24, " Select Desired Unit System ",
         BRIGHT WHITE, RED);
TTY.PUT ( 9, 24, " ",
         YELLOW, RED);
TTY.PUT (10, 24, " 1) English Engineering ",
         YELLOW, RED);
TTY.PUT (11, 24, " ",
         YELLOW, RED);
TTY.PUT (12, 24, " 2) SI ",
         YELLOW, RED);
TTY.PUT (13, 24, " ",
         YELLOW, RED);
TTY.PUT (14, 24, " ",
         YELLOW, RED);

45
TTY.PUT (15, 24, "Input your selection: [2] ", YELLOW, RED);
TTY.PUT (16, 24, " Press enter to accept default ", BRIGHT_WHITE, RED);
TTY.PUT (17, 24, " ", YELLOW, RED);
CURSOR.SET_SIZE(13,13);
loop
  CURSOR.MOVE (15, 50);
  TTY.GET (UNITS, CHAR);
  if UNITS = 2 or UNITS = 3 or UNITS = 28 then
    exit;
  else
    TTY.PUT (21, 24, " Improper input, please reenter ", BLUE, CYAN);
  end if;
end loop;
CURSOR.INHIBIT;

 Loop

Use English Engineering System

if (UNITS = 2) then
  CONVERT_DIST := 12.0;
  -- Convert inches to feet
  CONVERT_TEMP := 460.0;
  -- Convert deg-F to deg-R
  GRAVITY := 32.2;
  -- Acceleration of gravity (ft/s^2)
  G_C := 32.2;
  -- Conversion factor (lbm*ft/(lbf*s^2))
  SPACING := SPACING_ENGLISH;
  SPACING_UNITS := "in";
  DENSITY := DENSITY_ENGLISH;
  DENSITY_UNITS := "lbm/ft^3";
  SPECIFIC_HEAT := SPECIFIC_HEAT_ENGLISH;
  SPECIFIC_HEAT_UNITS := "BTU/(lbm*deg-R)";
  NU := NU_ENGLISH;
  NU_UNITS := "ft^2/s";
  WALL_LENGTH := WALL_LENGTH_ENGLISH;
  WALL_LENGTH_UNITS := "in";
  WALL_WIDTH := WALL_WIDTH_ENGLISH;
  WALL_WIDTH_UNITS := "in";
  NUM_FINS_UNITS := "fins";
  DIAMETER := DIAMETER_ENGLISH;
  DIAMETER_UNITS := "in";
  HEIGHT := HEIGHT_ENGLISH;
  HEIGHT_UNITS := "in";
  WIDTH := WIDTH_ENGLISH;
  WIDTH_UNITS := "in";
  LENGTH := LENGTH_ENGLISH;
  LENGTH_UNITS := "in";
  VOLUME := VOLUME_ENGLISH;
  VOLUME_UNITS := "in^3";
  H := H_ENGLISH;
  H_UNITS := "BTU/(hr*ft^2*deg-R)";
  K := K_ENGLISH;
  K_FLUID := K_FLUID_ENGLISH;

46
K UNITS := "BTU/(hr*ft*deg-R)";
T_AMBIENT := T_AMBIENT/english;
T_WALL := T_WALL/english;
T_UNITS := "deg-F";
Q := Q/english;
Q_UNITS := "BTU/hr";

---

Use SI System
---

else

CONVERT DIST := 100.0;
--- Convert centimeters to meters
CONVERT TEMP := 273.15;
--- Convert deg-C to deg-K
GRAVITY := 9.81;
--- Acceleration of gravity (m/s^2)
G_C := 1.0;
--- Conversion factor (kg*m/(N*s^2))
SPACING := SPACING_SI;
SPACING_UNITS := "cm";
DENSITY := DENSITY_SI;
DENSITY_UNITS := "kg/m^3";
SPECIFIC_HEAT := SPECIFIC_HEAT_SI;
SPECIFIC_HEAT_UNITS := "J/(kg*deg-K)";
NU := NU_SI;
NU_UNITS := "m^2/s";
WALL_LENGTH := WALL_LENGTH_SI;
WALL_LENGTH_UNITS := "cm";
WALL_WIDTH := WALL_WIDTH_SI;
WALL_WIDTH_UNITS := "cm";
NUM_FINS_UNITS := "fins";
DIAMETER := DIAMETER_SI;
DIAMETER_UNITS := "cm";
HEIGHT := HEIGHT_SI;
HEIGHT_UNITS := "cm";
WIDTH := WIDTH_SI;
WIDTH_UNITS := "cm";
LENGTH := LENGTH_SI;
LENGTH_UNITS := "cm";
VOLUME := VOLUME_SI;
VOLUME_UNITS := "cm^3";
H := H_SI;
H_UNITS := "W/(m^2*deg-K)";
K := K_SI;
K_FLUID := K_FLUID_SI;
K_UNITS := "W/(m*deg-K)";
T_AMBIENT := T_AMBIENT_SI;
T_WALL := T_WALL_SI;
T_UNITS := "deg-C";
Q := Q_SI;
Q_UNITS := "W";
end if;

---

Select Single or Multiple Fin Optimization
---

47
TTY.CLEAR_SCREEN;
TTY.PUT ( 6, 21, " ", YELLOW, CYAN);
TTY.PUT ( 7, 24, " ", YELLOW, CYAN);
TTY.PUT ( 8, 21, " ", YELLOW, CYAN);
TTY.PUT ( 8, 55, " ", YELLOW, CYAN);
TTY.PUT ( 9, 21, " ", YELLOW, CYAN);
TTY.PUT ( 9, 55, " ", YELLOW, CYAN);
TTY.PUT (10, 21, " ", YELLOW, CYAN);
TTY.PUT (10, 55, " ", YELLOW, CYAN);
TTY.PUT (11, 21, " ", YELLOW, CYAN);
TTY.PUT (11, 55, " ", YELLOW, CYAN);
TTY.PUT (12, 21, " ", YELLOW, CYAN);
TTY.PUT (12, 55, " ", YELLOW, CYAN);
TTY.PUT (13, 21, " ", YELLOW, CYAN);
TTY.PUT (13, 55, " ", YELLOW, CYAN);
TTY.PUT (14, 21, " ", YELLOW, CYAN);
TTY.PUT (14, 55, " ", YELLOW, CYAN);
TTY.PUT (15, 21, " ", YELLOW, CYAN);
TTY.PUT (15, 55, " ", YELLOW, CYAN);
TTY.PUT (16, 21, " ", YELLOW, CYAN);
TTY.PUT (16, 55, " ", YELLOW, CYAN);
TTY.PUT (17, 21, " ", YELLOW, CYAN);
TTY.PUT (17, 55, " ", YELLOW, CYAN);
TTY.PUT (18, 21, " ", YELLOW, CYAN);
TTY.PUT (19, 21, " ", YELLOW, CYAN);
TTY.PUT ( 8, 24, " Select Fin Optimization ", YELLOW, RED);
TTY.PUT ( 9, 24, " ", BRIGHT_WHITE, RED);
TTY.PUT (10, 24, " ", YELLOW, RED);
TTY.PUT (11, 24, " 1) Single Fin ", YELLOW, RED);
TTY.PUT (12, 24, " ", YELLOW, RED);
TTY.PUT (13, 24, " 2) Multiple Fins ", YELLOW, RED);
TTY.PUT (14, 24, " ", YELLOW, RED);
TTY.PUT (15, 24, " Input your selection: [2] ", YELLOW, RED);
TTY.PUT (16, 24, " Press enter to accept default ", BRIGHT_WHITE, RED);
TTY.PUT (17, 24, " ", YELLOW, RED);
CURSOR.SET_SIZE(13, 13);
loop
  CURSOR.MOVE (15, 50);
  TTY.GET (FIN_SING_MULT, CHAR);
  if FIN_SING_MULT = 2 or FIN_SING_MULT = 3 or
     FIN_SING_MULT = 28 then
    exit;
  else
    TTY.PUT (21, 24, " Improper input, please r:enter ", BLUE, CYAN);
    end if;
  end loop;
CURSOR.INHIBIT;
-- Single Fin Problem, Select Cylindrical or Rectangular Fin Type --

if (FIN_SING_MULT = 2) then
    TTY.CLEAR_SCREEN;
    TTY.PUT ( 6, 21, "",
        YELLOW, CYAN);
    TTY.PUT ( 7, 21, "",
        YELLOW, CYAN);
    TTY.PUT ( 8, 21, "",
        YELLOW, CYAN);
    TTY.PUT ( 8, 55, "",
        YELLOW, CYAN);
    TTY.PUT ( 9, 21, "",
        YELLOW, CYAN);
    TTY.PUT ( 9, 55, "",
        YELLOW, CYAN);
    TTY.PUT (10, 21, "",
        YELLOW, CYAN);
    TTY.PUT (10, 55, "",
        YELLOW, CYAN);
    TTY.PUT (11, 21, "",
        YELLOW, CYAN);
    TTY.PUT (11, 55, "",
        YELLOW, CYAN);
    TTY.PUT (12, 21, "",
        YELLOW, CYAN);
    TTY.PUT (12, 55, "",
        YELLOW, CYAN);
    TTY.PUT (13, 21, "",
        YELLOW, CYAN);
    TTY.PUT (13, 55, "",
        YELLOW, CYAN);
    TTY.PUT (14, 21, "",
        YELLOW, CYAN);
    TTY.PUT (14, 55, "",
        YELLOW, CYAN);
    TTY.PUT (15, 21, "",
        YELLOW, CYAN);
    TTY.PUT (15, 55, "",
        YELLOW, CYAN);
    TTY.PUT (16, 21, "",
        YELLOW, CYAN);
    TTY.PUT (16, 55, "",
        YELLOW, CYAN);
    TTY.PUT (17, 21, "",
        YELLOW, CYAN);
    TTY.PUT (17, 55, "",
        YELLOW, CYAN);
    TTY.PUT (18, 21, "",
        YELLOW, CYAN);
    TTY.PUT (19, 21, "",
        YELLOW, CYAN);
    TTY.PUT ( 8, 24, "",
        YELLOW, RED);
    TTY.PUT ( 9, 24, " Select Desired Fin Shape ",
        BRIGHT WHITE, RED);
    TTY.PUT (10, 24, "",
        YELLOW, RED);
    TTY.PUT (11, 24, " 1) Cylindrical Spine ",
        YELLOW, RED);
    TTY.PUT (12, 24, "",
        YELLOW, RED);
    TTY.PUT (13, 24, " 2) Rectangular fin ",
        YELLOW, RED);
    TTY.PUT (14, 24, "",
        YELLOW, RED);
    TTY.PUT (15, 24, " Input your selection: [2] ",
        YELLOW, RED);
    TTY.PUT (16, 24, " Press enter to accept default ",
        BRIGHT WHITE, RED);
    TTY.PUT (17, 24, "",
        YELLOW, RED);
    CURSOR.SET_SIZE(13,13);
    loop
    CURSOR.MOVE (15, 50);
TTY.GET (FIN_CYLRECT, CHAR);
if FIN_CYLRECT = 2 or FIN_CYLRECT = 3 or
FIN_CYLRECT = 28 then
    exit;
else
    TTY.PUT (21, 24, " Improper input, please reenter ",
        BLUE, CYAN);
end if;
end loop;
CURSOR.INHIBIT;

------------------------------------------------------------------
Single Fin Problem, Cylindrical Spine, --
--- Draw Cylindrical Spine ---
------------------------------------------------------------------
if (FIN_CYL_RECT = 2) then
    DIAMETER_MSG := "Diameter ";
    HEIGHT_MSG := " Height ";
    FINOPT_DRAWINGS.CYLINDRICAL_DRAWING(DIAMETER_MSG,
        HEIGHT_MSG);
------------------------------------------------------------------
Single Fin Problem, Cylindrical Spine --
--- Choose Optimization ---
------------------------------------------------------------------
TTY.CLEAR_SCREEN;
TTY.PUT (2, 21, " ",
        YELLOW, CYAN);
TTY.PUT (3, 21, " ",
        YELLOW, CYAN);
TTY.PUT (4, 21, " ", YELLOW, CYAN);
TTY.PUT (4, 55, " ", YELLOW, CYAN);
TTY.PUT (5, 21, " ", YELLOW, CYAN);
TTY.PUT (5, 55, " ", YELLOW, CYAN);
TTY.PUT (6, 21, " ", YELLOW, CYAN);
TTY.PUT (6, 55, " ", YELLOW, CYAN);
TTY.PUT (7, 21, " ", YELLOW, CYAN);
TTY.PUT (7, 55, " ", YELLOW, CYAN);
TTY.PUT (8, 21, " ", YELLOW, CYAN);
TTY.PUT (8, 55, " ", YELLOW, CYAN);
TTY.PUT (9, 21, " ", YELLOW, CYAN);
TTY.PUT (9, 55, " ", YELLOW, CYAN);
TTY.PUT (10, 21, " ", YELLOW, CYAN);
TTY.PUT (10, 55, " ", YELLOW, CYAN);
TTY.PUT (11, 21, " ", YELLOW, CYAN);
TTY.PUT (11, 55, " ", YELLOW, CYAN);
TTY.PUT (12, 21, " ", YELLOW, CYAN);
TTY.PUT (12, 55, " ", YELLOW, CYAN);
TTY.PUT (13, 21, " ", YELLOW, CYAN);
TTY.PUT (13, 55, " ", YELLOW, CYAN);
TTY.PUT (14, 21, " ", YELLOW, CYAN);
TTY.PUT (14, 55, " ", YELLOW, CYAN);
TTY.PUT (15, 21, " ", YELLOW, CYAN);
TTY.PUT (15, 55, " ", YELLOW, CYAN);
TTY.PUT (16, 21, " ", YELLOW, CYAN);
TTY.PUT (16, 55, " ", YELLOW, CYAN);
50
TTY.PUT (17, 21, "", YELLOW, CYAN);
TTY.PUT (17, 55, "", YELLOW, CYAN);
TTY.PUT (18, 21, "", YELLOW, CYAN);
TTY.PUT (18, 55, "", YELLOW, CYAN);
TTY.PUT (19, 21, "", YELLOW, CYAN);
TTY.PUT (19, 55, "", YELLOW, CYAN);
TTY.PUT (20, 21, "", YELLOW, CYAN);
TTY.PUT (21, 21, "", YELLOW, CYAN);
TTY.PUT ( 4, 24, "", YELLOW, RED);
TTY.PUT ( 5, 24, " Choose Type of Optimization ", BRIGHT WHITE, RED);
TTY.PUT ( 6, 24, "", YELLOW, RED);
TTY.PUT ( 7, 24, " 1) No Optimization ", YELLOW, RED);
TTY.PUT ( 8, 24, "", YELLOW, RED);
TTY.PUT ( 9, 24, " 2) Maximum heat transfer ", YELLOW, RED);
TTY.PUT (10, 24, " capability for a given ", YELLOW, RED);
TTY.PUT (11, 24, " volume ", YELLOW, RED);
TTY.PUT (12, 24, "", YELLOW, RED);
TTY.PUT (13, 24, " 3) Minimum volume for a ", YELLOW, RED);
TTY.PUT (14, 24, " given heat transfer ", YELLOW, RED);
TTY.PUT (15, 24, " capability ", YELLOW, RED);
TTY.PUT (16, 24, "", YELLOW, RED);
TTY.PUT (17, 24, " Input your selection: [3] ", YELLOW, RED);
TTY.PUT (18, 24, " Press enter to accept default ", BRIGHT WHITE, RED);
TTY.PUT (19, 24, "", YELLOW, RED);
CURSOR.SET_SIZE(13,13);
loop
  CURSOR.MOVE (17, 50);
  TTY.GET (FIN_OPT_TYP, CHAR);
  if FIN_OPT_TYP = 2 or FIN_OPT_TYP = 3 or FIN_OPT_TYP = 4 or FIN_OPT_TYP = 28 then exit;
else
  TTY.PUT (23, 24, " Improper input, please reenter ", BLUE, CYAN);
end if;
end loop;
CURSOR.INHIBIT;

51
if (FIN_OPT_TYP = 2) then
   CYLINDRICAL_NO_OPT(UNITS, CONVERT_DIST, DIAMETER,
   DIAMETER_UNITS, HEIGHT, HEIGHT_UNITS, H, H_UNITS, K,
   K_UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q_UNITS);

elsif (FIN_OPT_TYP = 3) then
   CYLINDRICAL_GIVEN_VOL(UNITS, CONVERT_DIST, VOLUME,
   VOLUME_UNITS, DIAMETER, DIAMETER_UNITS, HEIGHT,
   HEIGHT_UNITS, H, H_UNITS, K, K_UNITS, T_AMBIENT,
   T_WALL, T_UNITS, Q, Q_UNITS);

else
   CYLINDRICAL_GIVEN_Q(UNITS, CONVERT_DIST, DIAMETER,
   DIAMETER_UNITS, HEIGHT, HEIGHT_UNITS, H, H_UNITS, K,
   K_UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q_UNITS);
end if;

PUT (DIAMETER_MSG(1..10), DIAMETER, 4, 3);
DIAMETER_MSG(11) := ' ';
DIAMETER_MSG(12..13) := DIAMETER_UNITS;
PUT (HEIGHT_MSG(1..10), HEIGHT, 4, 3);
HEIGHT_MSG(11) := ' ';
HEIGHT_MSG(12..13) := HEIGHT_UNITS;
FINOPT_DRAWINGS.CYLINDRICAL_DRAWING(DIAMETER_MSG,
HEIGHT_MSG);

else
   WIDTH_MSG := "Width ";
   LENGTH_MSG := "Length ";
   HEIGHT_MSG := " Height ";
   FINOPT_DRAWINGS.RECTANGULAR_DRAWING(WIDTH_MSG, LENGTH_MSG,
   HEIGHT_MSG);
end if;
TTY.PUT (11, 24, " volume and length ", YELLOW, RED);
TTY.PUT (12, 24, " ", YELLOW, RED);
TTY.PUT (13, 24, " 3) Minimum volume for a ", YELLOW, RED);
TTY.PUT (14, 24, " given heat transfer ", YELLOW, RED);
TTY.PUT (15, 24, " capability ", YELLOW, RED);
TTY.PUT (16, 24, " ", YELLOW, RED);
TTY.PUT (17, 24, " Input your selection: [3] ", YELLOW, RED);
TTY.PUT (18, 24, " Press enter to accept default ", BRIGHT_WHITE, RED);
TTY.PUT (19, 24, " ", YELLOW, RED);
CURSOR.SET_SIZE(13,13);
loop
  CURSOR.MOVE (17, 50);
  TTY.GET (FIN_OPT_TYP, CHAR);
  if FIN_OPT_TYP = 2 or FIN_OPT_TYP = 3 or FIN_OPT_TYP = 4 or FIN_OPT_TYP = 28 then
    exit;
  else
    TTY.PUT (23, 24, " Improper input, please reenter ", BLUE, CYAN);
  end if;
end loop;
CURSOR.INHIBIT;

-- Single Fin Problem, Rectangular Fin, No Optimization --

if (FIN_OPT_TYP = 2) then
  RECTANGULAR_NO_OPT(UNITS, CONVERT DIST, LENGTH, LENGTH_UNITS, HEIGHT, HEIGHT UNITS, WIDTH, WIDTH UNITS, H, H UNITS, K, K UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q UNITS);

-- Single Fin Problem, Rectangular Fin, Maximum Heat Transfer Capability for a Given Volume and Length --

elsif (FIN_OPT_TYP = 3) then
  RECTANGULAR_GIVEN_VOL(UNITS, CONVERT DIST, VOLUME, VOLUME UNITS, LENGTH, LENGTH UNITS, HEIGHT, HEIGHT UNITS, WIDTH, WIDTH UNITS, H, H UNITS, K, K UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q UNITS);

-- Single Fin Problem, Rectangular Fin, Minimum Volume for a Given Heat Transfer Capability --

else

    RECTANGULAR_GIVEN_Q(UNITS, CONVERT_DIST, LENGTH,
    LENGTH_UNITS, HEIGHT, HEIGHT_UNITS, WIDTH,
    WIDTH_UNITS, H, H_UNITS, K, K_UNITS, T_AMBIENT,
    T_WALL, T_UNITS, Q, Q_UNITS);
end if;

--
--
--
--
--

-- Single Fin Problem, Rectangular Fin, --
-- Draw Rectangular Fin With Calculated Dimensions --

    PUT (WIDTH_MSG(1..10), WIDTH, 4, 3);
    WIDTH_MSG(II) := ' ';
    WIDTH_MSG(12..13) := WIDTH_UNITS;
    PUT (LENGTH_MSG(1..10), LENGTH, 4, 3);
    LENGTH_MSG(II) := ' ';
    LENGTH_MSG(12..13) := LENGTH_UNITS;
    PUT (HEIGHT_MSG(1..10), HEIGHT, 4, 3);
    HEIGHT_MSG(II) := ' ';
    HEIGHT_MSG(12..13) := HEIGHT_UNITS;
    FINOPT_DRAWINGS.RECTANGULAR_DRAWING(WIDTH_MSG, LENGTH_MSG,
    HEIGHT_MSG);
end if;

--
-- Multiple Fin Problem, Rectangular Fins, Draw Rectangular Fins --

else

    WIDTH_MSG := " Width ";
    LENGTH_MSG := " Length ";
    HEIGHT_MSG := " Height ";
    SPACING_MSG := " Spacing ";
    WALL_LENGTH_MSG := " Wall Length ";
    WALL_WIDTH_MSG := " Wall Width ";
    FINOPT_DRAWINGS.MULTI_FIN_DRAWING(WIDTH_MSG, LENGTH_MSG,
    HEIGHT_MSG, SPACING_MSG, WALL_LENGTH_MSG, WALL_WIDTH_MSG);

--
-- Multiple Fin Problem, Rectangular Fins, Choose Optimization --

TTY.CLEAR_SCREEN;
TTY.PUT ( 2, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 3, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 4, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 4, 55, " 
    YELLOW, CYAN);
TTY.PUT ( 5, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 5, 55, " 
    YELLOW, CYAN);
TTY.PUT ( 6, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 6, 55, " 
    YELLOW, CYAN);
TTY.PUT ( 7, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 7, 55, " 
    YELLOW, CYAN);
TTY.PUT ( 8, 21, " 
    YELLOW, CYAN);
TTY.PUT ( 8, 55, " 
    YELLOW, CYAN);
TTY.PUT ( 9, 21, " 
    YELLOW, CYAN);
TTY.PUT (9, 55, " ", YELLOW, CYAN);
TTY.PUT (10, 21, " ", YELLOW, CYAN);
TTY.PUT (10, 55, " ", YELLOW, CYAN);
TTY.PUT (11, 21, " ", YELLOW, CYAN);
TTY.PUT (11, 55, " ", YELLOW, CYAN);
TTY.PUT (12, 21, " ", YELLOW, CYAN);
TTY.PUT (12, 55, " ", YELLOW, CYAN);
TTY.PUT (13, 21, " ", YELLOW, CYAN);
TTY.PUT (13, 55, " ", YELLOW, CYAN);
TTY.PUT (14, 21, " ", YELLOW, CYAN);
TTY.PUT (14, 55, " ", YELLOW, CYAN);
TTY.PUT (15, 21, " ", YELLOW, CYAN);
TTY.PUT (15, 55, " ", YELLOW, CYAN);
TTY.PUT (16, 21, " ", YELLOW, CYAN);
TTY.PUT (16, 55, " ", YELLOW, CYAN);
TTY.PUT (17, 21, " ", YELLOW, CYAN);
TTY.PUT (17, 55, " ", YELLOW, CYAN);
TTY.PUT (18, 21, " ", YELLOW, CYAN);
TTY.PUT (18, 55, " ", YELLOW, CYAN);
TTY.PUT (19, 21, " ", YELLOW, CYAN);
TTY.PUT (19, 55, " ", YELLOW, CYAN);
TTY.PUT (20, 21, " ", YELLOW, CYAN);
TTY.PUT (21, 21, " ", YELLOW, CYAN);
TTY.PUT (4, 24, " Choose Type of Optimization ", BRIGHT WHITE, RED);
TTY.PUT (5, 24, " 1) No Optimization ", YELLOW, RED);
TTY.PUT (6, 24, " ", YELLOW, RED);
TTY.PUT (7, 24, " 2) Maximum heat transfer capability from each fin in a given area ", YELLOW, RED);
TTY.PUT (8, 24, " ", YELLOW, RED);
TTY.PUT (9, 24, " 3) Maximum heat transfer capability for a given area ", YELLOW, RED);
TTY.PUT (10, 24, " ", YELLOW, RED);
TTY.PUT (11, 24, " ", YELLOW, RED);
TTY.PUT (12, 24, " ", YELLOW, RED);
TTY.PUT (13, 24, " ", YELLOW, RED);
TTY.PUT (14, 24, " ", YELLOW, RED);
TTY.PUT (15, 24, " ", YELLOW, RED);
TTY.PUT (16, 24, " ", YELLOW, RED);
TTY.PUT (17, 24, " Input your selection: [3] ", YELLOW, RED);
TTY.PUT (18, 24, " Press enter to accept default ", BRIGHT WHITE, RED);
TTY.PUT (19, 24, "", YELLOW, RED);
CURSOR.SET_SIZE(13, 13);
loop
    CURSOR.MOVE (17, 50);
    TTY.GET (FIN_OPT_TYP, CHAR);
    if FIN_OPT_TYP = 2 or FIN_OPT_TYP = 3 or FIN_OPT_TYP = 4 or FIN_OPT_TYP = 28 then
        exit;
    else
        TTY.PUT (23, 24, " Improper input, please reenter ", BLUE, CYAN);
    end if;
end loop;
CURSOR.INHIBIT;

-- Multiple Fin Problem, Rectangular Fins, No Optimization

if (FIN_OPT_TYP = 2) then
    MULTIPLE_NO_OPT(UNITS, CONVERT DIST, CONVERT TEMP, G_C, GRAVITY, WALL LENGTH, WALL LENGTH UNITS, WALL WIDTH, WALL WIDTH UNITS, LENGTH, LENGTH UNITS, HEIGHT, HEIGHT UNITS, WIDTH, WIDTH UNITS, SPACING, SPACING UNITS, NUM FINS, NUM_FINS UNITS, DENSITY, DENSITY UNITS, SPECIFIC HEAT, SPECIFIC HEAT UNITS, K, K_FLUID, K UNITS, NU, NU UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q UNITS);

-- Multiple Fin Problem, Rectangular Fins, Maximum Heat Transfer From Each Fin in a Given area

elsif (FIN_OPT_TYP = 3) then
    MULTIPLE_MAX_FIN(UNITS, CONVERT DIST, CONVERT TEMP, G_C, GRAVITY, WALL LENGTH, WALL LENGTH UNITS, WALL WIDTH, WALL WIDTH UNITS, LENGTH, LENGTH UNITS, HEIGHT, HEIGHT UNITS, WIDTH, WIDTH UNITS, SPACING, SPACING UNITS, NUM_FINS, NUM_FINS UNITS, DENSITY, DENSITY UNITS, SPECIFIC HEAT, SPECIFIC HEAT UNITS, K, K_FLUID, K UNITS, NU, NU UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q UNITS);

-- Single Fin Problem, Rectangular Fin, Maximum Heat Transfer Capability for a Given Area

else
    MULTIPLE_MAX_Q(UNITS, CONVERT DIST, CONVERT TEMP, G_C, GRAVITY, WALL LENGTH, WALL LENGTH UNITS, WALL WIDTH, WALL WIDTH UNITS, LENGTH, LENGTH UNITS, HEIGHT, HEIGHT UNITS, WIDTH, WIDTH UNITS, SPACING, SPACING UNITS, NUM_FINS, NUM_FINS UNITS, DENSITY, DENSITY UNITS, SPECIFIC HEAT, SPECIFIC HEAT UNITS, K, K_FLUID, K UNITS, NU, NU UNITS, T_AMBIENT, T_WALL, T_UNITS, Q, Q UNITS);

57
end if;

---

Multiple Fin Problem, Rectangular Fins,

---

Draw Rectangular Fins With Calculated Dimensions

---

PUT (WIDTH_MSG(1..10), WIDTH, 4, 3);
WIDTH_MSG(11) := ' ';
WIDTH_MSG(12..13) := WIDTH_UNITS;
PUT (LENGTH_MSG(1..10), LENGTH, 4, 3);
LENGTH_MSG(11) := ' ';
LENGTH_MSG(12..13) := LENGTH_UNITS;
PUT (HEIGHT_MSG(1..10), HEIGHT, 4, 3);
HEIGHT_MSG(11) := ' ';
HEIGHT_MSG(12..13) := HEIGHT_UNITS;
PUT (SPACING_MSG(1..10), SPACING, 4, 3);
SPACING_MSG(11) := ' '; SPACING_MSG(12..13) := SPACING_UNITS;
PUT (WALL_LENGTH_MSG(1..10), WALL_LENGTH, 4, 3);
WALL_LENGTH_MSG(11) := ' '; WALL_LENGTH_MSG(12..13) := WALL_LENGTH_UNITS;
PUT (WALL_WIDTH_MSG(1..10), WALL_WIDTH, 4, 3);
WALL_WIDTH_MSG(11) := ' '; WALL_WIDTH_MSG(12..13) := WALL_WIDTH_UNITS;
FINOPT_DRAWINGS.MULTI_FIN_DRAWING(WIDTH_MSG, LENGTH_MSG,
HEIGHT_MSG, SPACING_MSG, WALL_LENGTH_MSG, WALL_WIDTH_MSG);

end if;

---

Reinitialize Variables

---

if (UNITS = 2) then
  SPACING_ENGLISH := SPACING;
  DENSITY_ENGLISH := DENSITY;
  SPECIFIC_HEAT_ENGLISH := SPECIFIC_HEAT;
  NU_ENGLISH := NU;
  K_FLUID_ENGLISH := K_FLUID;
  WALL_LENGTH_ENGLISH := WALL_LENGTH;
  WALL_WIDTH_ENGLISH := WALL_WIDTH;
  NUM_FINS_ENGLISH := NUM_FINS;
  DIAMETER_ENGLISH := DIAMETER;
  HEIGHT_ENGLISH := HEIGHT;
  WIDTH_ENGLISH := WIDTH;
  LENGTH_ENGLISH := LENGTH;
  VOLUME_ENGLISH := VOLUME;
  H_ENGLISH := H;
  K_ENGLISH := K;
  T_AMBIENT_ENGLISH := T_AMBIENT;
  T_WALL_ENGLISH := T_WALL;
  Q_ENGLISH := Q;
else
  SPACING_SI := SPACING;
  DENSITY_SI := DENSITY;
  SPECIFIC_HEAT_SI := SPECIFIC_HEAT;
  NU_SI := NU;
  K_FLUID_SI := K_FLUID;

58
WALL LENGTH SI := WALL LENGTH;
WALL WIDTH SI := WALL WIDTH;
NUM FINS SI := NUM FINS;
DIAMETER SI := DIAMETER;
HEIGHT SI := HEIGHT;
WIDTH SI := WIDTH;
LENGTH SI := LENGTH;
VOLUME SI := VOLUME;
H SI := H;
K SI := K;
T_AMBIENT SI := T_AMBIENT;
T_WALL SI := T_WALL;
Q SI := Q;
end if;
end loop;

-- Exit to DOS --
FINOPT_PICTURES.EXIT_MSG;
VIDEO.SET_COLOR_PALETTE (BLACK);
CURSOR.MOVE (22, 1);
CURSOR.SET_SIZE(13,13);
end FINOPT;
package FINOPT_COMMON is

   procedure GET_INPUT
      (INPUT_VALUE : in out FLOAT;
       INPUT_MSG   : in STRING;
       SIZE_INPUT_MSG  : in INTEGER;
       INPUT_VALUE_UNITS : in STRING;
       SIZE_INPUT_VALUE_UNITS : in INTEGER;
       ROW_START : in INTEGER);

end FINOPT_COMMON;
package body FINOPT_COMMON is

package FLOAT_INOUT is new FLOATIO(FLOAT);
use FLOAT_INOUT;

procedure GET_INPUT(INPUT VALUE : in out FLOAT;
INPUT MSG : in STRING;
SIZE INPUT MSG : in INTEGER;
INPUT VALUE UNITS : in STRING;
SIZE INPUT VALUE UNITS : in INTEGER;
ROW_START : in INTEGER) is

NUMBER OUT : STRING(I..10);
CHAR : CHARACTER;
DEFAULT_KEY : INTEGER;

begin
TTY.PUT (ROWSTART, 1, INPUT_MSG, YELLOW, BLACK);
TTY.PUT (ROWSTART, 1+SIZE_INPUT_MSG, "(" , YELLOW, BLACK);
TTY.PUT (ROW_START, 3+SIZE_INPUT_MSG, INPUT VALUE UNITS(I..SIZE_INPUT VALUE UNITS), YELLOW, BLACK);
TTY.PUT (ROW_START+1, 1, "Press enter to accept default or any other key to enter new value", BRIGHT WHITE, BLACK);
PUT (NUMBER_OUT, INPUT VALUE, 4, 3);
TTY.PUT (ROW_START, SIZE_INPUT_VALUE_UNITS+7+SIZE_INPUT_MSG, NUMBER_OUT, YELLOW, BLACK);
-- CURSOR.SET_SIZE (13,13);
TTY.GET (DEFAULT_KEY, CHAR);
-- CURSOR.INHIBIT;
if DEFAULT_KEY /= 28 then
loop
begin
TTY.PUT (ROW_START, SIZE_INPUT_VALUE_UNITS+7+SIZE_INPUT_MSG, "", BLACK, BLACK);
TTY.PUT (ROW_START+2, 1, "New value = ", BRIGHT_WHITE, BLACK);
CURSOR.SET_SIZE (13,13);
GET (INPUT_VALUE);
SKIP_LINE;
end
end
end

61
CURSOR.INHIBIT;
exit;
exception
when others =>
  TTY.PUT (23, 24, " Improper input, please reenter ",
             BLUE, CYAN);
end;
end loop;
TTY.PUT (23, 24, " ",
         BLACK, BLACK);
TTY.PUT (ROW_START+2, 1, " ",
         BLACK, BLACK);
end if;
TTY.PUT (ROW_START+1, 1,
         ",
         BLACK, BLACK);
TTY.PUT (ROW_START, 1+SIZE_INPUT_MSG, 
         ",
         YELLOW, BLACK);
TTY.PUT (ROW_START, 46, " ", BLACK, BLACK);
TTY.PUT (ROW_START, 45, " = ", YELLOW, BLACK);
PUT (NUMBER_OUT, INPUT_VALUE, 4, 3);
TTY.PUT (ROW_START, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (ROW_START, 58, ", YELL0W, BLACK);
TTY.PUT (ROW_START, 59,
INPUT_VALUE_UNITS(1..SIZE_INPUT_VALUE_UNITS),
YELLOW, BLACK);
TTY.PUT (ROW_START, 59+SIZE_INPUT_VALUE_UNITS,
         ", YELLOW, BLACK);
end GET_INPUT;

end FINOPT_COMMON;
package FINOPT_DRAWINGS is

  procedure CYLINDRICAL_DRAWING(DIAMETER_MSG : in STRING;
                                 HEIGHT_MSG  : in STRING);

  procedure RECTANGULAR_DRAWING(WIDTH_MSG      : in STRING;
                                 LENGTH_MSG    : in STRING;
                                 HEIGHT_MSG    : in STRING);

  procedure MULTI_FIN_DRAWING(WIDTH_MSG   : in STRING;
                               LENGTH_MSG   : in STRING;
                               HEIGHT_MSG   : in STRING;
                               SPACING_MSG  : in STRING;
                               WALL_LENGTH_MSG  : in STRING;
                               WALL_WIDTH_MSG  : in STRING);

end FINOPT_DRAWINGS;
package body FINOPT_DRAWINGS is

procedure CYLINDRICAL_DRAWING(DIAMETER_MSG : in STRING;
HEIGTH_MSG : in STRING) is

PAUSE : INTEGER;
CHAR : CHARACTER;

begin

TTY.CLEAR_SCREEN;
TTY.PUT (0, 20, "•", YELLOW, BLACK);
TTY.PUT (1, 18, "I", YELLOW, BLACK);
TTY.PUT (1, 19, "I", YELLOW, RED);
TTY.PUT (2, 16, "I", YELLOW, BLACK);
TTY.PUT (2, 17, "I", YELLOW, RED);
TTY.PUT (3, 14, "I", YELLOW, BLACK);
TTY.PUT (3, 15, "I", YELLOW, RED);
TTY.PUT (4, 12, "I", YELLOW, BLACK);
TTY.PUT (4, 13, "I", YELLOW, RED);
TTY.PUT (5, 10, "I", YELLOW, BLACK);
TTY.PUT (5, 11, "I", YELLOW, RED);
TTY.PUT (6, 8, "I", YELLOW, BLACK);
TTY.PUT (6, 9, "I", YELLOW, RED);
TTY.PUT (7, 6, "I", YELLOW, BLACK);
TTY.PUT (7, 7, "I", YELLOW, RED);
TTY.PUT (8, 5, "I", YELLOW, RED);
TTY.PUT (8, 25, "I", YELLOW, BLACK);
TTY.PUT (9, 5, "I", YELLOW, RED);
TTY.PUT (9, 50, "I", YELLOW, BLACK);
TTY.PUT (10, 5, "I", YELLOW, RED);
TTY.PUT (10, 52, "I", YELLOW, BLACK);
TTY.PUT (11, 5, "I", YELLOW, BLACK);
TTY.PUT (11, 54, "I", YELLOW, BLACK);
TTY.PUT (11, 61, DIAMETER_MSG, YELLOW, BLACK);
TTY.PUT (12, 5, "I", YELLOW, RED);
TTY.PUT (12, 54, "I", YELLOW, BLACK);
end CYLINDRICAL_DRAWING;

procedure RECTANGULAR_DRAWING(WIDTH_MSG : in STRING;
LENGTH_MSG : in STRING;
HEIGHT_MSG : in STRING) is

PAUSE : INTEGER;
CHAR : CHARACTER;

begin

TTY.CLEARSCREEN;
TTY.PUT (0, 25, " _ \__|___ ", YELLOW, BLACK);
TTY.PUT (1, 23, "| |", YELLOW, BLACK);
TTY.PUT (1, 24, "/| |", YELLOW, RED);
TTY.PUT (2, 21, "/| |", YELLOW, BLACK);
TTY.PUT (2, 22, " /| |", YELLOW, RED);
TTY.PUT (3, 19, "/| |", YELLOW, BLACK);
TTY.PUT (3, 20, " /| |", YELLOW, RED);
TTY.PUT (4, 17, "| |", YELLOW, BLACK);
TTY.PUT (4, 18, " /| |", YELLOW, RED);
TTY.PUT (4, 32, "\| |", YELLOW, BLACK);
TTY.PUT (5, 15, "| |", YELLOW, BLACK);
TTY.PUT (5, 16, " /| |", YELLOW, BLACK);
TTY.PUT (5, 53, "| |", YELLOW, BLACK);

65
end RECTANGULAR_DRAWING;

procedure MULTI_FIN_DRAWING(WIDTH_MSG : in STRING;
LENGTH_MSG : in STRING;
HEIGHT_MSG : in STRING;
SPACING_MSG : in STRING;
WALL_LENGTH_MSG : in STRING;
WALL_WIDTH_MSG : in STRING) is
    PAUSE : INTEGER;
    CHAR : CHARACTER;

begin

    TTY.CLEAR_SCREEN;
    TTY.PUT (0, 27, WALL_WIDTH_MSG, YELLOW, BLACK);
    TTY.PUT (1, 10,

package FINOPT_MULTIPLE is

procedure MULTIPLE_NO_OPT(UNITS in INTEGER;
CONVERT DIST : in FLOAT;
CONVERT TEMP : in FLOAT;
G C : in FLOAT;
GRAVITY : in FLOAT;
WALL LENGTH : in out FLOAT;
WALL_LENGTH_UNITS : in STRING;
WALL WIDTH : in out FLOAT;
WALL_WIDTH_UNITS : in STRING;
LENGTH : in out FLOAT;
LENGTH_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
WIDTH : in out FLOAT;
WIDTH_UNITS : in STRING;
SPACING : in out FLOAT;
SPACING_UNITS : in STRING;
NUM FINES : in out FLOAT;
NUM_FINS_UNITS : in STRING;
DENSITY : in out FLOAT;
DENSITY_UNITS : in STRING;
SPECIFIC HEAT : in out FLOAT;
SPECIFIC_HEAT_UNITS : in STRING;
K : in out FLOAT;
K_FLUID : in out FLOAT;
K_UNITS : in STRING;
NU : in out FLOAT;
NU_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING);

procedure MULTIPLE_MAX_FIN(UNITS in INTEGER;
CONVERT DIST in FLOAT;
CONVERT TEMP in FLOAT;
G C in FLOAT;
GRAVITY in FLOAT;
WALL LENGTH in out FLOAT;
WALL_LENGTH_UNITS in STRING;
WALL WIDTH in out FLOAT;
WALL_WIDTH_UNITS in STRING;
LENGTH in out FLOAT;
LENGTH_UNITS in STRING;
HEIGHT in out FLOAT;
HEIGHT_UNITS in STRING;
WIDTH in out FLOAT;
WIDTH_UNITS in STRING);
procedure MULTIPLE_MAX_Q(UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
CONVERT_TEMP : in FLOAT;
G C : in FLOAT;
GRAVITY : in FLOAT;
WALL_LENGTH : in out FLOAT;
WALL_LENGTH_UNITS : in STRING;
WALL_WIDTH : in out FLOAT;
WALL_WIDTH_UNITS : in STRING;
LENGTH : in out FLOAT;
LENGTH_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
WIDTH : in out FLOAT;
WIDTH_UNITS : in STRING;
SPACING : in out FLOAT;
SPACING_UNITS : in STRING;
NUM_FINS : in out FLOAT;
NUM_FINS_UNITS : in STRING;
DENSITY : in out FLOAT;
DENSITY_UNITS : in STRING;
SPECIFIC_HEAT : in out FLOAT;
SPECIFIC_HEAT_UNITS : in STRING;
K : in out FLOAT;
K_FLUID : in out FLOAT;
K_UNITS : in STRING;
NU : in out FLOAT;
NU_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING);
package body FINOPT_MULTIPLE is

package FLOAT_INOUT is new FLOAT_IO(FLOAT);
package MY_ELEMENTARY_FUNCTIONS is
  new GENERIC_ELEMENTARY_FUNCTIONS(FLOAT);
use FLOAT_INOUT, MY_ELEMENTARY_FUNCTIONS;

procedure MULTIPLE_NO_OPT (UNITS : in INTEGER;
  CONVERT_DIST : in FLOAT;
  CONVERT_TEMP : in FLOAT;
  G_C : in FLOAT;
  GRAVITY : in FLOAT;
  WALL_LENGTH : in out FLOAT;
  WALL_LENGTH_UNITS : in STRING;
  WALL_WIDTH : in out FLOAT;
  WALL_WIDTH_UNITS : in STRING;
  LENGTH : in out FLOAT;
  LENGTH_UNITS : in STRING;
  HEIGHT : in out FLOAT;
  HEIGHT_UNITS : in STRING;
  WIDTH : in out FLOAT;
  WIDTH_UNITS : in STRING;
  SPACING : in out FLOAT;
  SPACING_UNITS : in STRING;
  NUM_FINS : in out FLOAT;
  NUM_FINS_UNITS : in STRING;
  DENSITY : in out FLOAT;
  DENSITY_UNITS : in STRING;
  SPECIFIC_HEAT : in out FLOAT;
  SPECIFIC_HEAT_UNITS : in STRING;
  K : in out FLOAT;
  K_FLUID : in out FLOAT;
  K_UNITS : in STRING;
  NU : in out FLOAT;
  NU_UNITS : in STRING;
  T_AMBIENT : in out FLOAT;
  T_WALL : in out FLOAT;
  T_UNITS : in STRING;
  Q : in out FLOAT;
  Q_UNITS : in STRING) is

NUMBER_OUT : STRING(1..10);
CHAR : CHARACTER;
PAUSE, NUM_FINS_INT : INTEGER;
PERIMETER, AREA, M, EFFICIENCY, 
DELTA T, T TIP, T AVG, BETA, MU, 
RAYLEIGH_CHANNEL, NUSSELT_CHANNEL, H, 
AREA_BASE, AREA_FIN, AREA_TOTAL : FLOAT;

begin 

-- Inputs --

FINOPT_PICTURES.INPUT_MSG;
GET_INPUT(WALL_LENGTH, "Length of the fin placement area", 32, 
WALL_LENGTH_UNITS, 2, 13);
GET_INPUT(WALL_WIDTH, "Width of the fin placement area", 31, 
WALL_WIDTH_UNITS, 2, 14);
GET_INPUT(LENGTH, "Length of each fin", 18, LENGTH_UNITS, 2, 15);
GET_INPUT(HEIGHT, "Height of each fin", 18, HEIGHT_UNITS, 2, 16);
GET_INPUT(WIDTH, "Width of each fin", 17, WIDTH_UNITS, 2, 17);
GET_INPUT(SPACING, "Spacing between fins", 20, SPACING_UNITS, 2, 18);

FINOPT_PICTURES.INPUT_CONT_MSG;
NUM_FINS := (WALL WIDTH-WIDTH)/(SPACING+WIDTH);
NUM_FINS := NUM_FINS-0.49999999999999999;
NUM_FINS_INT := INTEGER(NUM_FINS);
NUM_FINS_INT := NUM_FINS_INT+1;
NUM_FINS := FLOAT(NUM_FINS_INT);
GET_INPUT(NUM_FINS, "Number of fins, default is maximum number", 
41, NUM_FINS_UNITS, 4, 19);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
FINOPT_PICTURES.INPUT_CONT_MSG;
if (UNITS = 2) then
GET_INPUT(DENSITY, "Density of surrounding fluid", 
28, DENSITY_UNITS, 8, 13);
else
GET_INPUT(DENSITY, "Density of surrounding fluid", 
28, DENSITY_UNITS, 6, 13);
end if;

if (UNITS = 2) then
GET_INPUT(SPECIFIC_HEAT, "Specific heat of surrounding fluid", 
34, SPECIFIC_HEAT_UNITS, 15, 14);
GET_INPUT(K, "Thermal conductivity of material, k", 35, 
K_UNITS, 17, 15);
GET_INPUT(K_FLUID, 
"Thermal conductivity of surrounding fluid, k", 44, 
K_UNITS, 17, 16);
GET_INPUT(NU, "Kinematic viscosity of surrounding fluid", 
40, NU_UNITS, 6, 17);
else
GET_INPUT(SPECIFIC_HEAT, "Specific heat of surrounding fluid", 
34, SPECIFIC_HEAT_UNITS, 12, 14);
GET_INPUT(K, "Thermal conductivity of material, k", 35, 
K_UNITS, 11, 15);
GET_INPUT(K_FLUID, 
"Thermal conductivity of surrounding fluid, k", 44, 
K_UNITS, 11, 16);
end if;

72
GET_INPUT(NU, "Kinematic viscosity of surrounding fluid", 40, NU_UNITS, 5, 17);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19, T_UNITS, 5, 18);
GET_INPUT(T_WALL, "Wall Temperature", 16, T_UNITS, 5, 19);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

--- Calculations (Assume Tip is Insulated) and Length >> Width ---

T_AVG := (T_WALL+T_AMBIENT)/2.0;
BETA := 1.07(T_AVG+CONVERT TEMP);
DELTA_T := T_WALL-T_AMBIENT;
MU := (NU*Density)/G_C;
RAYLEIGH_CHANNEL := (((Density**2)*Gravity*BETA*Specific_Heat
*((Spacing/Convert Dist)**4)*Delta_T)/(MU*(Length/Convert Dist))
*K_FLUID);
NUSSELT_CHANNEL := ((576.0/(RAYLEIGH_CHANNEL**2))
+(2.8737*SQRT(RAYLEIGH_CHANNEL)))**(-0.5);
H := (NUSSELT_CHANNEL*K_FLUID)/(Spacing/Convert Dist);
PERIMETER := 2.0*Length/Convert Dist;
AREA := (Width/Convert Dist)*(Length/Convert Dist);
M := SQRT((H*PERIMETER)/(K*AREA));
EFFICIENCY := (TANH(M*HEIGHT/Convert Dist))
/(M*HEIGHT/Convert Dist);
AREA_BASE :=
(WALL_WIDTH/Convert Dist)*(WALL_LENGTH/Convert Dist))
-(NUM_FINS*(LENGTH/Convert Dist)*(WIDTH/Convert Dist));
AREA_FIN := NUM_FINS*(2.0*(HEIGHT/Convert Dist)
*(LENGTH/Convert Dist));
AREA_TOTAL := AREA_BASE+(EFFICIENCY*AREA_FIN);
2 := H*AREA_TOTAL*DELTA_T;
T_TIP := T_AMBIENT+(DELTA_T/COSH(M*HEIGHT/Convert Dist));

--- Outputs ---

FINOPT_PICTURES.OUTPUT_MSG;
TTY.PUT ( 5, 36, " Inputs ", BRIGHT WHITE, GREEN);
TTY.PUT ( 7, 1, "Length of the fin placement area = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, WALL_LENGTH, 4, 3);
TTY.PUT ( 7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 7, 59, WALL_LENGTH UNITS, YELLOW, BLACK);
TTY.PUT ( 8, 1, "Width of the fin placement area = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, WALL_WIDTH, 4, 3);
TTY.PUT ( 8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, WALL_WIDTH UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "Length of each fin = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, LENGTH, 4, 3);
TTY.PUT ( 9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 9, 59, LENGTH UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Height of each fin = ",
YELLOW, BLACK);
FINOPT_PICTURES.OUTPUT_CONT_MSG;
TTY.PUT ( 6, 35, " Outputs ", BRIGHT_WHITE, GREEN);
TTY.PUT ( 8, 1, "Heat transferred away by the fins, q = ", YELLOW, BLACK);
PUT (NUMBEROUT, Q, 4, 3);
TTY.PUT ( 8, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, Q UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "The fin efficiency = ", YELLOW, BLACK);
PUT (NUMBEROUT, EFFICIENCY, 4, 3);
TTY.PUT ( 9, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT ( 10, 1, "The temperature at the tip of the fins = ", YELLOW, BLACK);
PUT (NUMBEROUT, T TIP, 4, 3);
TTY.PUT ( 10, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT ( 10, 59, T UNITS, YELLOW, BLACK);
TTY.PUT ( 11, 1, "Channel Rayleigh number = ", YELLOW, BLACK);
PUT (NUMBEROUT, RAYLEIGH_CHANNEL, 4, 3);
TTY.PUT ( 11, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT ( 11, 59, RAYLEIGH CHANNEL Units, YELLOW, BLACK);
TTY.PUT ( 12, 1, "Channel Nusselt number = ", YELLOW, BLACK);
PUT (NUMBEROUT, NUSSELT_CHANNEL, 4, 3);
TTY.PUT ( 12, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end MULTIPLE_NO_OPT;

procedure MULTIPLE_MAX_FIN(UNITS

CONVERT DIST : in INTEGER;
CONVERT TEMP : in FLOAT;
G C : in FLOAT;
GRAVITY : in FLOAT;
WALL LENGTH : in out FLOAT;
WALL LENGTH UNITS : in STRING;
WALL WIDTH : in out FLOAT;
WALL WIDTH UNITS : in STRING;
LENGTH : in out FLOAT;
LENGTH UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT UNITS : in STRING;
WIDTH : in out FLOAT;
WIDTH UNITS : in STRING;
SPACING : in out FLOAT;
SPACING UNITS : in STRING;
NUM FINS : in out FLOAT;
NUM_FINS UNITS : in STRING;
DENSITY : in out FLOAT;
DENSITY UNITS : in STRING;
SPECIFIC HEAT : in out FLOAT;
SPECIFIC HEAT UNITS : in STRING;
K : in out FLOAT;
K_FLUID : in out FLOAT;
K UNITS : in STRING;
NÜ : in out FLOAT;
begin
-----------
-- Inputs --
---------------
FINOPT PICTURES.INPUT_MSG;
GET INPUT(WALL LENGTH, "Length of the fin placement area", 32, WALL LENGTH UNITS, 2, 14);
GET INPUT(WALL WIDTH, "Width of the fin placement area", 31, WALL WIDTH UNITS, 2, 15);
GET INPUT(LENGTH, "Length of each fin", 18, LENGTH UNITS, 2, 16);
GET INPUT(HEIGHT, "Height of each fin", 18, HEIGHT UNITS, 2, 17);
GET INPUT(WIDTH, "Width of each fin", 17, WIDTH UNITS, 2, 18);
if (UNITS = 2) then
  GET INPUT(DENSITY, "Density of surrounding fluid", 28, DENSITY UNITS, 8, 19);
else
  GET INPUT(DENSITY, "Density of surrounding fluid", 28, DENSITY UNITS, 6, 19);
end if;
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
FINOPT PICTURES.INPUT_CC: I_MSG;
if (UNITS = 2) then
  GET_INPUT(SPECIFIC HEAT, "Specific heat of surrounding fluid", 34, SPECIFIC HEAT UNITS, 15, 14);
  GET_INPUT(K, "Thermal conductivity of material, k", 35, K UNITS, 17, 15);
  GET_INPUT(K FLUID, "Thermal conductivity of surrounding fluid, k", 44, K UNITS, 17, 16);
  GET_INPUT(NU, "Kinematic viscosity of surrounding fluid", 40, NU UNITS, 6, 17);
else
  GET_INPUT(SPECIFIC HEAT, "Specific heat of surrounding fluid", 34, SPECIFIC HEAT UNITS, 12, 14);
  GET_INPUT(K, "Thermal conductivity of material, k", 35, K UNITS, 11, 15);
end if;
GET_INPUT(K_FLUID, "Thermal conductivity of surrounding fluid, k", 44, K_UNITS, 11, 16);
GET_INPUT(NU, "Kinematic viscosity of surrounding fluid", 40, NU_UNITS, 5, 17);

end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19, T_UNITS, 5, 18);
GET_INPUT(T_WALL, "Wall Temperature", 16, T_UNITS, 5, 19);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

-- Calculations (Assume Tip is Insulated) and Length >> Width --

T_AVG := (T_WALL+T_AMBIENT)/2.0;
BETA := 1.0/(T_AVG+CONVERT_TEMP);
DELTA_T := T_WALL-T_AMBIENT;
MU := (NU*DENSITY)/G_C;
P := ((DENSITY**2)*GRAVITY*BETA*SPECIFIC_HEAT*DELTA_T)/(MU*(LENGTH/CONVERT_DIST)*K_FLUID);
SPACING := CONVERT_DIST*(4.64/(P**(0.25)));
NUM_FINS := (WALL_WIDTH-WIDTH)/(SPACING+WIDTH);
NUM_FINS := INT(NUM_FINS);
NUM_FINS_INT := INT(NUM_FINS);
NUM_FINS := FLOAT(NUM_FINS_INT);
RAYLEIGH_CHANNEL := ((DENSITY**2)*GRAVITY*BETA*SPECIFIC_HEAT*(SPACING/CONVERT_DIST)**4)*DELTA_T)/(MU*(LENGTH/CONVERT_DIST)*K_FLUID);
NUSSELT_CHANNEL := ((576.0/(RAYLEIGH_CHANNEL**2))+(2.8737*SQRT(RAYLEIGH_CHANNEL))**(-0.5));
H := (NUSSELT_CHANNEL*K_FLUID)/(SPACING/CONVERT_DIST);
PERIMETER := 2.0*LENGTH/CONVERT_DIST;
AREA := (WALL_WIDTH/CONVERT_DIST)*H;
M := SQRT(H/PERIMETER)/(K*AREA);
EFFICIENCY := (TANH(M*HEIGHT/CONVERT_DIST))/(M*HEIGHT/CONVERT_DIST);

AREA_BASE := (WALL_WIDTH/CONVERT_DIST)*(WALL_LENGTH/CONVERT_DIST)-NUM_FINS*(2.0*HEIGHT/CONVERT_DIST)*LENGTH/CONVERT_DIST;
AREA_FIN := NUM_FINS*HEIGHT/CONVERT_DIST;
AREA_TOTAL := AREA_BASE+(EFFICIENCY*AREA_FIN);
Q := H*AREA_TOTAL*DELTA_T;
T_TIP := T_AMBIENT+(DELTA_T*COSH(M*HEIGHT/CONVERT_DIST));

-- Outputs --

FINOPT_PICTURES.OUTPUT_MSG;
TTY.PUT (5, 36, " Inputs ", BRIGHT_WHITE, GREEN);
TTY.PUT (7, 1, "Length of the fin placement area 

1, YELLOW, BLACK);
PUT (NUMBER_OUT, WALL_LENGTH, 4, 3);
TTY.PUT (7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (7, 59, WALL_LENGTH_UNITS, YELLOW, BLACK);
TTY.PUT ( 8, 1, "Width of the fin placement area = ", YELLOW, BLACK);
TTY.PUT ( 8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, WALL_WIDTH_UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "Length of each fin = ", YELLOW, BLACK);
TTY.PUT ( 9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 9, 59, LENGTH_UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Height of each fin = ", YELLOW, BLACK);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, HEIGHT_UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Width of each fin = ", YELLOW, BLACK);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 59, WIDTH_UNITS, YELLOW, BLACK);
TTY.PUT (12, 1, "Density of surrounding fluid = ", YELLOW, BLACK);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 59, DENSITY_UNITS, YELLOW, BLACK);
TTY.PUT (13, 1, "Specific heat of surrounding fluid = ", YELLOW, BLACK);
TTY.PUT (13, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (13, 59, SPECIFIC_HEAT_UNITS, YELLOW, BLACK);
TTY.PUT (14, 1, "Thermal conductivity of material, k = ", YELLOW, BLACK);
TTY.PUT (14, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (14, 59, K_UNITS, YELLOW, BLACK);
TTY.PUT (15, 1, "Thermal conductivity of surrounding fluid, k = ", YELLOW, BLACK);
TTY.PUT (15, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (15, 59, K_FLUID_UNITS, YELLOW, BLACK);
TTY.PUT (16, 1, "Kinematic viscosity of surrounding fluid = ", YELLOW, BLACK);
TTY.PUT (16, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (16, 59, NU_UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "Ambient Temperature = ", YELLOW, BLACK);
TTY.PUT (17, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (17, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "Wall Temperature = ", YELLOW, BLACK);
TTY.PUT (18, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (18, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
FINOPT_PICTURES.OUTPUT_CONT_MSG;
TTY.PUT (6, 35, "Outputs ", BRIGHT_WHITE, GREEN);
TTY.PUT (8, 1, "Heat transferred away by the fins, q 
YELLOW, BLACK);
PUT (NUMBER_OUT, Q, 4, 3);
TTY.PUT (8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (8, 59, Q_UNITS, YELLOW, BLACK);
TTY.PUT (9, 1, "Spacing between fins 
YELLOW, BLACK);
PUT (NUMBER_OUT, SPACING, 4, 3);
TTY.PUT (9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (9, 59, SPACING_UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Number of fins 
YELLOW, BLACK);
PUT (NUMBER_OUT, NUM_FINS, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, NUM_FINS_UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "The fin efficiency 
YELLOW, BLACK);
PUT (NUMBER_OUT, EFFICIENCY, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 1, "The temperature at the tip of the fins 
YELLOW, BLACK);
PUT (NUMBER_OUT, T_TIP, 4, 3);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (13, 1, "Channel Rayleigh number 
YELLOW, BLACK);
PUT (NUMBER_OUT, RAYLEIGH_CHANNEL, 4, 3);
TTY.PUT (13, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (14, 1, "Channel Nusselt number 
YELLOW, BLACK);
PUT (NUMBER_OUT, NUSSELT_CHANNEL, 4, 3);
TTY.PUT (14, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (23, 27, "Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end MULTIPLE_MAX_FIN;

procedure MULTIPLE_MAX_Q(UNITs

CONVERT_DIST : in INTEGER;
CONVERT_TEMP : in FLOAT;
G_C : in FLOAT;
GRAVITY : in FLOAT;
WALL_LENGTH : in out FLOAT;
WALL_LENGTH_UNITS : in STRING;
WALL_WIDTH : in out FLOAT;
WALL_WIDTH_UNITS : in STRING;
LENGTH : in out FLOAT;
LENGTH_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
WIDTH : in out FLOAT;
WIDTH_UNITS : in STRING;
SPACING : in out FLOAT;

79
begin

Inputs

FINOPT_PICTURES.INPUT_MSG;
GET_INPUT(WALL_LENGTH, "Length of the fin placement area", 32, WALL_LENGTH UNITS, 2, 14);
GET_INPUT(WALL_WIDTH, "Width of the fin placement area", 31, WALL_WIDTH UNITS, 2, 15);
GET_INPUT(LENGTH, "Length of each fin", 18, LENGTH UNITS, 2, 16);
GET_INPUT(HEIGHT, "Height of each fin", 18, HEIGHT UNITS, 2, 17);
GET_INPUT(WIDTH, "Width of each fin", 17, WIDTH UNITS, 2, 18);
if (UNITS = 2) then
    GET_INPUT(DENSITY, "Density of surrounding fluid", 28, DENSITY UNITS, 8, 19);
else
    GET_INPUT(DENSITY, "Density of surrounding fluid", 28, DENSITY UNITS, 6, 19);
end if;
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
FINOPT_PICTURES.INPUT_CONT_MSG;
if (UNITS = 2) then
    GET_INPUT(SPECIFIC_HEAT, "Specific heat of surrounding fluid", 34, SPECIFIC_HEAT UNITS, 15, 14);
    GET_INPUT(K, "Thermal conductivity of material, k", 35,
K_UNITS, 17, 15);
GET_INPUT(K_FLUID,
"Thermal conductivity of surrounding fluid, k", 44,
K_UNITS, 17, 16);
GET_INPUT(NU, "Kinematic viscosity of surrounding fluid",
40, NU_UNITS, 6, 17);
else
GET_INPUT(SPECIFIC_HEAT, "Specific heat of surrounding fluid",
34, SPECIFIC_HEAT_UNITS, 12, 14);
GET_INPUT(K, "Thermal conductivity of material, k", 35,
K_UNITS, 11, 15);
GET_INPUT(K_FLUID,
"Thermal conductivity of surrounding fluid, k", 44,
K_UNITS, 11, 16);
GET_INPUT(NU, "Kinematic viscosity of surrounding fluid",
40, NU_UNITS, 5, 17);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19, T_UNITS, 5, 18);
GET_INPUT(T_WALL, "Wall Temperature", 16, T_UNITS, 5, 19);
TTY.PUT
(23, 27,
"Press any key to continue",
BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

--- Calculations (Assume Tip is Insulated) and Length >> Width ---

T_AVG := (T_WALL+T_AMBIENT)/2.0;
BETA := 1.07(T_AVG+CONVERT_TEMP);
DELTA_T := T_WALL-T_AMBIENT;
MU := (NU*DENSITY)/G
P := ((DENSITY**2)*GRAVITY*BETA*SPECIFIC_HEAT*DELTA_T)
/(MU*(LENGTH/CONVERT_DIST)*K_FLUID);
SPACING := CONVERT_DIST*(2.714/(P**0.25));
NUM_FINS := (WALL_WIDTH-WIDTH)/(SPACING+WIDTH);
NUM_FINS := NUM_FINS-0.49999999999;
NUM_FINS_INT := INTEGER(NUM_FINS);
NUM_FINS_INT := NUM_FINS_INT+1;
NUM_FINS := FLOAT(NUM_FINS_INT);
RAYLEIGH_CHANNEL := ((DENSITY**2)*GRAVITY*BETA*SPECIFIC_HEAT
*((SPACING/CONVERT_DIST)**4)*DELTA_T)/(MU*(LENGTH/CONVERT_DIST)
*K_FLUID);
NUSSELT_CHANNEL := ((576.0/RAYLEIGH_CHANNEL**2))
+2.037/SQRT(RAYLEIGH_CHANNEL))**(-0.5);
H := (NUSSELT_CHANNEL*K_FLUID)/(SPACING/CONVERT_DIST);
PERIMETER := 2.0*LENGTH/CONVERT_DIST;
AREA := (WIDTH/CONVERT_DIST)*(LENGTH/CONVERT_DIST);
M := SQRT((H*PERIMETER)/(K*AREA));
EFFICIENCY := (TANH(M*HEIGHT/CONVERT_DIST))
/(M*HEIGHT/CONVERT_DIST);
AREA_BASE :=
((WALL_WIDTH/CONVERT_DIST)*(WALL_LENGTH/CONVERT_DIST))
- (NUM_FINS*(LENGTH/CONVERT_DIST)*(WIDTH/CONVERT_DIST));
AREA_FIN := NUM_FINS*(2.0*(HEIGHT/CONVERT_DIST)
*(LENGTH/CONVERT_DIST));
AREA_TOTAL := AREA_BASE+(EFFICIENCY*AREA_FIN);
Q := H*AREA_TOTAL*DELTA_T;
T_TIP := T_AMBIENT+(DELTA_T/COSH(M*HEIGHT/CONVERT_DIST));
FINOPT_PICTURES.OUTPUT_MSG;
TTY.PUT ( 5, 36, " Inputs ", BRIGHT_WHITE, GREEN);
TTY.PUT ( 7, 1, "Length of the fin placement area = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, WALL_LENGTH, 4, 3);
TTY.PUT ( 7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 7, 59, WALL_LENGTH_UNITS, YELLOW, BLACK);
TTY.PUT ( 8, 1, "Width of the fin placement area = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, WALL_WIDTH, 4, 3);
TTY.PUT ( 8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, WALL_WIDTH_UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "Length of each fin = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, LENGTH, 4, 3);
TTY.PUT ( 9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 9, 59, LENGTH_UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Height of each fin = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, HEIGHT, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, HEIGHT_UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Width of each fin = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, WIDTH, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 59, WIDTH_UNITS, YELLOW, BLACK);
TTY.PUT (12, 1, "Density of surrounding fluid = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, DENSITY, 4, 3);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 59, DENSITY_UNITS, YELLOW, BLACK);
TTY.PUT (13, 1, "Specific heat of surrounding fluid = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, SPECIFIC_HEAT, 4, 3);
TTY.PUT (13, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (13, 59, SPECIFIC_HEAT_UNITS, YELLOW, BLACK);
TTY.PUT (14, 1, "Thermal conductivity of material, k = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, K, 4, 3);
TTY.PUT (14, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (14, 59, K_UNITS, YELLOW, BLACK);
TTY.PUT (15, 1, "Thermal conductivity of surrounding fluid, k = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, K_FLUID, 4, 3);
TTY.PUT (15, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (15, 59, K_UNITS, YELLOW, BLACK);
TTY.PUT (16, 1, "Kinematic viscosity of surrounding fluid = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, NU, 4, 3);
TTY.PUT (16, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (16, 59, NU_UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "Ambient Temperature = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, T_AMBIENT, 4, 3);
TTY.PUT (17, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (17, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "Wall Temperature = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, T_WALL, 4, 3);
TTY.PUT (18, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (18, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
FINOPT_PICTURES.OUTPUT_CONT_MSG;
TTY.PUT (6, 35, " Outputs ", BRIGHT WHITE, GREEN);
TTY.PUT (8, 1, "Heat transferred away by the fins, q = ", YELLOW, BLACK);
TTY.PUT (8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (8, 59, Q_UNITS, YELLOW, BLACK);
TTY.PUT (9, 1, "Spacing between fins = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, SPACING, 4, 3);
TTY.PUT (9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (9, 59, SPACING_UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Number of fins = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, NUM_FINS, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, NUM_FINS_UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "The fin efficiency = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, EFFICIENCY, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 1, "The temperature at the tip of the fins = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, T_TIP, 4, 3);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (13, 1, "Channel Rayleigh number = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, RAYLEIGH_CHANNEL, 4, 3);
TTY.PUT (13, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (14, 1, "Channel Nusselt number = ", YELLOW, BLACK);
TTY.PUT (NUMBER_OUT, NUSSELT_CHANNEL, 4, 3);
TTY.PUT (14, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end MULTIPLE_MAX_Q;
end FINOPT_MULTIPLE;
package FINOPT_PICTURES is
    procedure THESIS_MSG;
    procedure FINOPT_MSG;
    procedure INPUT_MSG;
    procedure INPUT_CONT_MSG;
    procedure OUTPUT_MSG;
    procedure OUTPUT_CONT_MSG;
    procedure EXIT_MSG;
end FINOPT_PICTURES;
package body FINOPT_PICTURES is

procedure THESIS_MSG is

PAUSE : INTEGER;
CHAR : CHARACTER;

begin
CURSOR.INHIBIT;
TTY.CLEAR_SCREEN;
TTY.PUT (2, 28, "NAVAL POSTGRADUATE SCHOOL", YELLOW, BLACK);
TTY.PUT (4, 31, "Monterey, California", YELLOW, BLACK);
TTY.PUT (6, 37, "THESIS", YELLOW, BLACK);
TTY.PUT (8, 16, "EXTENDED SURFACE HEAT SINKS FOR ELECTRONIC COMPONENTS:
A COMPUTER OPTIMIZATION by John Reynold Gensure
June 1992")

end THESIS_MSG;
TTY.PUT (19, 16, " Thesis Advisor: Allan D. Kraus
YELLOW, RED);
TTY.PUT (20, 16, "
YELLOW, RED);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end THESIS_MSG;

procedure FINOPT_MSG is

PAUSE : INTEGER;
CHAR  : CHARACTER;

begin
TTY.CLEAR_SCREEN;
TTY.PUT (4, 34, "Welcome To", GREEN, BLACK);
TTY.PUT (8, 16, " ", YELLOW, GREEN);
TTY.PUT (8, 24, " ", YELLOW, GREEN);
TTY.PUT (8, 27, " ", YELLOW, GREEN);
TTY.PUT (8, 31, " ", YELLOW, GREEN);
TTY.PUT (8, 34, " ", YELLOW, GREEN);
TTY.PUT (8, 41, " ", YELLOW, GREEN);
TTY.PUT (8, 48, " ", YELLOW, GREEN);
TTY.PUT (8, 55, " ", YELLOW, GREEN);
TTY.PUT (8, 58, " ", YELLOW, GREEN);
TTY.PUT (8, 61, " ", YELLOW, GREEN);
TTY.PUT (9, 18, ", YELLOW, GREEN);
TTY.PUT (9, 24, " ", YELLOW, GREEN);
TTY.PUT (9, 27, " ", YELLOW, GREEN);
TTY.PUT (9, 31, " ", YELLOW, GREEN);
TTY.PUT (9, 34, " ", YELLOW, GREEN);
TTY.PUT (9, 38, " ", YELLOW, GREEN);
TTY.PUT (9, 41, " ", YELLOW, GREEN);
TTY.PUT (9, 45, " ", YELLOW, GREEN);
TTY.PUT (9, 50, " ", YELLOW, GREEN);
TTY.PUT (9, 55, " ", YELLOW, GREEN);
TTY.PUT (9, 58, " ", YELLOW, GREEN);
TTY.PUT (9, 61, " ", YELLOW, GREEN);
TTY.PUT (10, 18, " ", YELLOW, GREEN);
TTY.PUT (10, 24, " ", YELLOW, GREEN);
TTY.PUT (10, 27, " ", YELLOW, GREEN);
TTY.PUT (10, 29, " ", YELLOW, GREEN);
TTY.PUT (10, 31, " ", YELLOW, GREEN);
TTY.PUT (10, 34, " ", YELLOW, GREEN);
TTY.PUT (10, 38, " ", YELLOW, GREEN);
TTY.PUT (10, 41, " ", YELLOW, GREEN);
TTY.PUT (10, 50, " ", YELLOW, GREEN);
TTY.PUT (10, 55, " ", YELLOW, GREEN);
TTY.PUT (10, 58, " ", YELLOW, GREEN);
TTY.PUT (10, 61, " ", YELLOW, GREEN);
TTY.PUT (11, 18, " ", YELLOW, GREEN);
TTY.PUT (11, 24, " ", YELLOW, GREEN):
end FINOPT_MSG;

procedure INPUT_MSG is
begin
    TTY.CLEAR_SCREEN;
    TTY.PUT ( 1, 19, "", YELLOW, CYAN);
    TTY.PUT ( 2, 19, "", YELLOW, CYAN);
    TTY.PUT ( 3, 19, "", YELLOW, CYAN);
    TTY.PUT ( 3, 58, "", YELLOW, CYAN);
    TTY.PUT ( 4, 19, "", YELLOW, CYAN);
    TTY.PUT ( 4, 58, "", YELLOW, CYAN);
    TTY.PUT ( 5, 19, "", YELLOW, CYAN);
    TTY.PUT ( 5, 58, "", YELLOW, CYAN);
    TTY.PUT ( 6, 19, "", YELLOW, CYAN);
    TTY.PUT ( 6, 58, "", YELLOW, CYAN);
    TTY.PUT ( 7, 19, "", YELLOW, CYAN);
    TTY.PUT ( 7, 58, "", YELLOW, CYAN);
    TTY.PUT ( 8, 19, "", YELLOW, CYAN);
    TTY.PUT ( 8, 58, "", YELLOW, CYAN);
    TTY.PUT ( 9, 19, "", YELLOW, CYAN);
    TTY.PUT ( 9, 58, "", YELLOW, CYAN);
    TTY.PUT (10, 19, "", YELLOW, CYAN);
    TTY.PUT (11, 19, "", YELLOW, CYAN);
    TTY.PUT ( 3, 22, "", YELLOW, RED);
    TTY.PUT ( 4, 22, "", BRIGHT_WHITE, RED);
    TTY.PUT ( 5, 22, "", YELLOW, RED);
    TTY.PUT ( 6, 22, "", Press enter to choose default or ",

    "Version 1.0 dated June 1992", GREEN, BLACK);
    TTY.PUT (19, 24, "Written by John Reynold Gensure", GREEN, BLACK);
    TTY.PUT (23, 26, "Press any key to continue ", BLUE, CYAN);
    TTY.GET (PAUSE, CHAR);
end INPUT_MSG;

procedure INPUT_CONT_MSG is

begin

TTY.CLEAR_SCREEN;
TTY.PUT (1, 19, " ", YELLOW, CYAN);
TTY.PUT (2, 19, ", YELOW, CYAN);
TTY.PUT (3, 19, ", YELOW, CYAN);
TTY.PUT (3, 58, ", YELOW, CYAN);
TTY.PUT (4, 19, ", YELOW, CYAN);
TTY.PUT (4, 58, ", YELOW, CYAN);
TTY.PUT (5, 19, ", YELOW, CYAN);
TTY.PUT (5, 58, ", YELOW, CYAN);
TTY.PUT (6, 19, ", YELOW, CYAN);
TTY.PUT (6, 58, ", YELOW, CYAN);
TTY.PUT (7, 19, ", YELOW, CYAN);
TTY.PUT (7, 58, ", YELOW, CYAN);
TTY.PUT (8, 19, ", YELOW, CYAN);
TTY.PUT (8, 58, ", YELOW, CYAN);
TTY.PUT (9, 19, ", YELOW, CYAN);
TTY.PUT (9, 58, ", YELOW, CYAN);
TTY.PUT (10, 19, ", YELOW, CYAN);
TTY.PUT (11, 19, ", YELOW, CYAN);
TTY.PUT (3, 22, " ", YELLOW, RED);
TTY.PUT (4, 22, " Required Inputs Continued ", BRIGHT_WHITE, RED);
TTY.PUT (5, 22, " ", YELLOW, RED);
TTY.PUT (6, 22, " Press enter to choose default or ", YELLOW, RED);
TTY.PUT (7, 22, " any other key to input new value. ", YELLOW, RED);
TTY.PUT (6, 22, " All values must be inputted as ", YELLOW, RED);
TTY.PUT (7, 22, " floats. Examples: 5.0 or 2.0E-3 ", YELLOW, RED);
TTY.PUT (8, 22, " Do not input 0.8 as .8 !!! ", BRIGHT_WHITE, RED);

end INPUT_CONT_MSG;
TTY.PUT ( 9, 22, " 
YELLOW, RED);
end INPUT_CONT_MSG;

procedure OUTPUT_MSG is
begin
TTY.CLEAR_SCREEN;
TTY.PUT ( 1, 26, " 
YELLOW, CYAN);
TTY.PUT ( 2, 26, " 
YELLOW, CYAN);
TTY.PUT ( 2, 53, " 
YELLOW, CYAN);
TTY.PUT ( 3, 26, " 
YELLOW, CYAN);
TTY.PUT ( 2, 28, " Inputs => Outputs 
BRIGHT_WHITE, RED);
end OUTPUT_MSG;

procedure OUTPUT_CONT_MSG is
begin
TTY.CLEAR_SCREEN;
TTY.PUT ( 1, 26, " 
YELLOW, CYAN);
TTY.PUT ( 2, 26, " 
YELLOW, CYAN);
TTY.PUT ( 2, 53, " 
YELLOW, CYAN);
TTY.PUT ( 3, 26, " 
YELLOW, CYAN);
TTY.PUT ( 3, 53, " 
YELLOW, CYAN);
TTY.PUT ( 4, 26, " 
YELLOW, CYAN);
TTY.PUT ( 2, 28, " Inputs => Outputs 
BRIGHT_WHITE, RED);
TTY.PUT ( 3, 28, " Continued 
BRIGHT_WHITE, RED);
end OUTPUT_CONT_MSG;

procedure EXIT_MSG is
begin
TTY.CLEAR_SCREEN;
TTY.PUT ( 11, 18, " 
YELLOW, CYAN);
TTY.PUT ( 11, 58, " 
YELLOW, CYAN);
TTY.PUT ( 12, 18, " 
YELLOW, CYAN);
TTY.PUT ( 12, 58, " 
YELLOW, CYAN);
TTY.PUT ( 13, 18, " 
YELLOW, CYAN);
TTY.PUT ( 13, 58, " 
YELLOW, CYAN);
TTY.PUT ( 9, 18, " 
YELLOW, CYAN);
TTY.PUT ( 10, 18, " 
YELLOW, CYAN);
TTY.PUT ( 14, 18, " 
YELLOW, CYAN);
TTY.PUT ( 15, 18, " 
YELLOW, CYAN);
TTY.PUT ( 11, 21, " 
YELLOW, RED);
end EXIT_MSG;
TTY.PUT (12, 21, " Thank you for using FINOPT !!! ", YELLOW, RED);
TTY.PUT (13, 21, " ", YELLOW, RED);
end EXIT_MSG;
end FINOPT_PICTURES;
package FINOPT_SINGLE is

procedure CYLINDRICAL_NO_OPT(UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
DIAMETER : in out FLOAT;
DIAMETER_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
H : in STRING;
H_UNITS : in out FLOAT;
K : in out STRING;
K_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING);

procedure CYLINDRICAL_GIVEN_VOL(UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
VOLUME : in out FLOAT;
VOLUME_UNITS : in STRING;
DIAMETER : in out FLOAT;
DIAMETER_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
H : in out FLOAT;
H_UNITS : in STRING;
K : in out STRING;
K_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING);

procedure CYLINDRICAL_GIVEN_Q(UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
DIAMETER : in out FLOAT;
DIAMETER_UNITS : in STRING;
VOLUME : in out FLOAT;
VOLUME_UNITS : in STRING;
DIAMETER : in out FLOAT;
DIAMETER_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
H : in out FLOAT;
H_UNITS : in STRING;
K : in out STRING;
K_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING);
FLOAT;

HEIGHT  : in out
HEIGHT_UNITS : in STRING;
H  : in out
H_UNITS : in STRING;
K  : in out
K_UNITS : in STRING;
T_AMBIENT  : in out
T_UNITS : in STRING;
T_WALL  : in out
Q  : in out
Q_UNITS : in STRING);

procedure RECTANGULAR_NO_OPT(UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
LENGTH  : in out FLOAT;
LENGTH_UNITS : in STRING;
HEIGHT  : in out FLOAT;
HEIGHT_UNITS : in STRING;
WIDTH  : in out FLOAT;
WIDTH_UNITS : in STRING;
H  : in out FLOAT;
H_UNITS : in STRING;
K  : in out FLOAT;
K_UNITS : in STRING;
T_AMBIENT  : in out FLOAT;
T_UNITS : in STRING;
T_WALL  : in out FLOAT;
Q  : in out FLOAT;
Q_UNITS : in STRING);

procedure RECTANGULAR_GIVEN_VOL(UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
VOLUME  : in out FLOAT;
VOLUME_UNITS : in STRING;
LENGTH  : in out FLOAT;
LENGTH_UNITS : in STRING;
HEIGHT  : in out FLOAT;
HEIGHT_UNITS : in STRING;
WIDTH  : in out FLOAT;
WIDTH_UNITS : in STRING;
H  : in out FLOAT;
H_UNITS : in STRING;
92
procedure RECTANGULAR_GIVEN_Q(UNITS
CONVERT_DIST
LENGTH
FLOAT;
LENGTH_UNITS
HEIGHT
FLOAT;
HEIGHT_UNITS
WIDTH
FLOAT;
WIDTH_UNITS
H
FLOAT;
H_UNITS
K
FLOAT;
K_UNITS
T_AMBIENT
FLOAT;
T_WALL
FLOAT;
T_UNITS
Q
FLOAT;
Q_UNITS
STRING);
package body FINOPT_SINGLE is

package FLOAT INOUT is new FLOAT_IO(FLOAT);

package MY_ELEMENTARY_FUNCTIONS is
  new GENERIC_ELEMENTARY_FUNCTIONS(FLOAT);

use FLOAT_INOUT, MY_ELEMENTARY_FUNCTIONS;

procedure CYLINDRICAL_NOOPT(UNITS:
in INTEGER;
CONVERT_DIST:
in out FLOAT;
DIAMETER:
in FLOAT;
DIAMETER_UNITS:
in STRING;
HEIGHT:
in out FLOAT;
HEIGHT_UNITS:
in STRING;
H:
in out FLOAT;
H_UNITS:
in STRING;
K:
in out FLOAT;
K_UNITS:
in STRING;
T_AMBIENT:
in out FLOAT;
T_WALL:
in out FLOAT;
T_UNITS:
in STRING;
Q:
in out FLOAT;
Q_UNITS:
in STRING) is

  NUMBER_OUT
STRING(1..10);

  CHAR
: CHARACTER;

  PAUSE
: INTEGER;

  PERIMETER, AREA, M, EFFICIENCY,
DELTAT, TTIP
: FLOAT;

  PI
: constant :=
3.14159_26535_89793_23846_26433_83279_50288_41972 ;

begin

-- Inputs --

FINOPT_PICTURES.INPUT_MSG;
GET_INPUT(DIAMETER, "Diameter of the cylindrical spine", 33,
DIAMETER_UNITS, 2, 14);
GET_INPUT(HEIGHT, "Height of the cylindrical spine", 31,
HEIGHT_UNITS, 2, 15);
if (UNITS = 2) then
    GETINPUT(H, "Convection heat transfer coefficient, h", 39, 
    H UNITS, 19, 16);
    GETINPUT(K, "Thermal conductivity of material, k", 35, 
    K UNITS, 17, 17);
else
    GETINPUT(H, "Convection heat transfer coefficient, h", 39, 
    H UNITS, 13, 16);
    GETINPUT(K, "Thermal conductivity of material, k", 35, 
    K UNITS, 11, 17);
end if;
GET INPUT(T AMBIENT, "Ambient Temperature", 19, 
T UNITS, 5, 18);
GET INPUT(T WALL, "Wall Temperature", 16, 
T UNITS, 5, 19);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

-- Calculations (Assume Tip is Insulated) --

PERIMETER := PI*DIAMETER/CONVERT DIST;
AREA := (PI*(DIAMETER/CONVERT DIST)**2)/4.0;
M := SQRT((H*PERIMETER)/(K*AREA));
DELTA T := T WALL-T AMBIENT;
Q := K*AREA*M*DELTA T*TANH(M*HEIGHT/CONVERT DIST);
EFFICIENCY := (TANH(M*HEIGHT/CONVERT DIST))
//(M*HEIGHT/CONVERT DIST);
T TIP := T AMBIENT+(DELTA T/COSH(M*HEIGHT/CONVERT DIST));

-- Outputs --
FINOPT PICTURES.OUTPUT(MSG;
TTY.PUT (5, 36, " Inputs ", BRIGHT WHITE, GREEN);
TTY.PUT (7, 1, " Diameter of the cylindrical spine 
    = ", 
    YELLOW, BLACK);
PUT (NUMBER OUT, DIAMETER, 4, 3);
TTY.PUT (7, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (7, 59, DIAMETER UNITS, YELLOW, BLACK);
TTY.PUT (8, 1, " Height of the cylindrical spine 
    = ", 
    YELLOW, BLACK);
PUT (NUMBER OUT, HEIGHT, 4, 3);
TTY.PUT (8, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (8, 59, HEIGHT UNITS, YELLOW, BLACK);
TTY.PUT (9, 1, " Convection heat transfer coefficient, h 
    = ", 
    YELLOW, BLACK);
PUT (NUMBER OUT, H, 4, 3);
TTY.PUT (9, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (9, 59, H UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, " Thermal conductivity of material, k 
    = ", 
    YELLOW, BLACK);
PUT (NUMBER OUT, K, 4, 3);
TTY.PUT (10, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (10, 59, K UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, " Ambient Temperature 
    = ", 

95
YELLOW, BLACK;
PUT (NUMBER_OUT, T_AMBIENT, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (12, 1, "Wall Temperature = ", YELLOW, BLACK);
PUT (NUMBER_OUT, T_WALL, 4, 3);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (14, 35, " Outputs ", BRIGHT_WHITE, GREEN);
TTY.PUT (16, 1, "Heat transferred away by the fin, q = ", YELLOW, BLACK);
PUT (NUMBER_OUT, Q, 4, 3);
TTY.PUT (16, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (16, 59, Q_UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "The fin efficiency = ", YELLOW, BLACK);
PUT (NUMBER_OUT, EFFICIENCY, 4, 3);
TTY.PUT (17, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (17, 59, Q_UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "The temperature at the tip = ", YELLOW, BLACK);
PUT (NUMBER_OUT, T_TIP, 4, 3);
TTY.PUT (18, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (18, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end CYLINDRICAL_NO_OPT;

procedure CYLINDRICAL_GIVEN_VOL (UNITS : in INTEGER;
CONVERT_DIST : in FLOAT;
VOLUME : in out FLOAT;
VOLUME_UNITS : in STRING;
DIAMETER : in out FLOAT;
DIAMETER_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
H : in out FLOAT;
H_UNITS : in STRING;
K : in out FLOAT;
K_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING);
NUMBER_OUT
STRING(1.10);

CHAR : CHARACTER;

PAUSE : INTEGER;

PERIMETER, AREA, M, EFFICIENCY,
DELTA_T, T_TIP : FLOAT;

PI : constant :=
3.14159_26535_89793_23846_26433_83279_50288_41972 ;

begin

----------------------------------------------------------------- Inputs -----------------------------------------------------------------
FINOPT PICTURES.INPUT_MSG;
GET_INPUT(VOLUME, "Volume of the cylindrical spine", 31,
VOLUME UNITS, 4, 14);
if (UNITS = 2) then
  GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
  H UNITS, 19, 15);
  GET_INPUT(K, "Thermal conductivity of material, k", 35,
  K UNITS, 17, 16);
else
  GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
  H UNITS, 13, 15);
  GET_INPUT(K, "Thermal conductivity of material, k", 35,
  K UNITS, 11, 16);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19,
T UNITS, 5, 17);
GET_INPUT(T_WALL, "Wall Temperature", 16,
T UNITS, 5, 18);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

----------------------------------------------------------------- Calculations (Assume Tip is Insulated) -----------------------------------------------------------------
DIAMETER := CONVERT DIST*1.5031*(H/K*(VOLUME
/CONVERT DIST**3)**2)**(0.2);
HEIGHT := CONVERT DIST*0.5636*((VOLUME
/CONVERT DIST**3)*(K/H)**2)**(0.2);
PERIMETER := PI*DIAMETER/CONVERT DIST;
AREA := (PI*((DIAMETER/CONVERT DIST)**2))/4.0;
M := SQRT((H*PERIMETER)/(K*AREA));
DELTA_T := T_WALL-T_AMBIENT;
Q := K*AREA*M*DELTA_T*TANH(M*HEIGHT/CONVERT DIST);
EFFICIENCY := (TANH(M*HEIGHT/CONVERT DIST))
/(M*HEIGHT/CONVERT DIST);
T_TIP := T_AMBIENT+(DELTA_T/COSH(M*HEIGHT/CONVERT DIST));
FINOPT PICTURES. TPUT_MSG;
TTY.PUT ( 5, 36, " Inputs ", BRIGHT WHITE, GREEN);
TTY.PUT ( 7, 1, "Volume of the cylindrical spine = ",
YELLOW, BLACK);  
PUT (NUMBER_OUT, VOLUME, 4, 3);
TTY.PUT ( 7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 7, 59, VOLUME_UNITS, YELLOW, BLACK);
TTY.PUT ( 8, 1, "Convection heat transfer coefficient, h = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, H, 4, 3);
TTY.PUT ( 8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, H_UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "Thermal conductivity of material, k = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, K, 4, 3);
TTY.PUT ( 9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 9, 59, K_UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Ambient Temperature = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_AMBIENT, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Wall Temperature = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_WALL, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (13, 35, " Outputs ", BRIGHT WHITE, GREEN);
TTY.PUT (15, 1, "Optimum diameter of the cylindrical spine = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, DIAMETER, 4, 3);
TTY.PUT (15, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (15, 59, DIAMETER_UNITS, YELLOW, BLACK);
TTY.PUT (16, 1, "Optimum height of the cylindrical spine = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, HEIGHT, 4, 3);
TTY.PUT (16, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (16, 59, HEIGHT_UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "Heat transferred away by the fin, q = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, Q, 4, 3);
TTY.PUT (17, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (17, 59, Q_UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "The fin efficiency = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, EFFICIENCY, 4, 3);
TTY.PUT (18, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (19, 1, "The temperature at the tip 
YELLOW, BLACK);
PUT (NUMBER_OUT, T_TIP, 4, 3);
TTY.PUT (19, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (19, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN):
TTY.GET (PAUSE, CHAR);

end CYLINDRICAL_GIVEN_VOL;

procedure CYLINDRICAL_GIVEN_Q(UNITS
CONVERT_DIST
FLOAT;
DIAMETER
FLOAT;
DIAMETER_UNITS
FLOAT;
HEIGHT
FLOAT;
HEIGHT_UNITS
FLOAT;
H
FLOAT;
H_UNITS
FLOAT;
K
FLOAT;
K_UNITS
FLOAT;
T_AMBIENT
FLOAT;
T_WALL
FLOAT;
T_UNITS
FLOAT;
Q
FLOAT;
Q_UNITS
is
NUMBER_OUT
STRING(1..10);
CHAR
PAUSE
PERIMETER, AREA, M, EFFICIENCY,
DELTA_T, T_TIP
PI : constant :=
3.14159_26535_89793_23846_26433_83279_50288_41972 ;

begin

--------------------------------------------------------------
-- Inputs
--------------------------------------------------------------
FINOPT_PICTURES.INPUT_MSG;
if (UNITS = 2) then
  GET_INPUT(Q, "Heat transferred away by the fin, q", 35,
    Q_UNITS, 6, 14);
  GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
    H_UNITS, 19, 15);
  GET_INPUT(K, "Thermal conductivity of material, k", 35,
    K_UNITS, 17, 16);
else
  GET_INPUT(Q, "Heat transferred away by the fin, q", 35,
Q UNITS, 1, 14);
GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
H UNITS, 13, 15);
GET_INPUT(K, "Thermal conductivity of material, k", 35,
K UNITS, 11, 16);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19,
T UNITS, 5, 17);
GET_INPUT(T_WALL, "Wall Temperature", 16,
T UNITS, 5, 18);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

-- Calculations (Assume Tip is Insulated) --

DIAMETER := CONVERT DIST*0.9165*{(Q**2)
/(H*K*((T_WALL-T_AMBIENT)**2))**1.0/3.0);
HEIGHT := CONVERT DIST*0.4400*{(Q*K)
/(H**2)*((T_WALL-T_AMBIENT))**1.0/3.0);
PERIMETER := PI*DIAMETER/CONVERT DIST;
AREA := (PI*((DIAMETER/CONVERT DIST)**2))/4.0;
M := SQRT((H*PERIMETER)/(K*AREA));
DELTA_T := T_WALL-T_AMBIENT;
Q := K*AREA*M*DELTA_T*TANH (M*HEIGHT/CONVERT DIST);
EFFICIENCY := (TANH(M*HEIGHT/CONVERT DIST))
/(M*HEIGHT/CONVERT DIST);
T_TIP := T_AMBIENT+(DELTA_T/COSH(M*HEIGHT/CONVERT DIST));

-- Outputs --

FINOPT_PICTURES.OUTPUT MSG;
TTY.PUT ( 5, 36, " Inputs ", BRIGHT WHITE, GREEN);
TTY.PUT ( 7, 1, "Heat transferred away by the fin, q = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, Q, 4, 3);
TTY.PUT ( 7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 7, 59, Q UNITS, YELLOW, BLACK);
TTY.PUT ( 8, 1, "Convection heat transfer coefficient, h = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, H, 4, 3);
TTY.PUT ( 8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, H UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "Thermal conductivity of material, k = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, K, 4, 3);
TTY.PUT ( 9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 9, 59, K UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Ambient Temperature = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_AMBIENT, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Wall Temperature = ",
YELLOW, BLACK);

100
PUT (NUMBEROUT, T_WALL, 4, 3);
TTY.PUT (11, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT (11, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (13, 35, "Outputs ", BRIGHT_WHITE, GREEN);
TTY.PUT (15, 1, "Optimum diameter of the cylindrical spine = ",
YELLOW, BLACK);
PUT (NUMBEROUT, DIAMETER, 4, 3);
TTY.PUT (15, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT (15, 59, DIAMETER_UNITS, YELLOW, BLACK);
TTY.PUT (16, 1, "Optimum height of the cylindrical spine = ",
YELLOW, BLACK);
PUT (NUMBEROUT, HEIGHT, 4, 3);
TTY.PUT (16, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT (16, 59, HEIGHT_UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "The fin efficiency = ",
YELLOW, BLACK);
PUT (NUMBEROUT, EFFICIENCY, 4, 3);
TTY.PUT (17, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT (18, 1, "The temperature at the tip = ",
YELLOW, BLACK);
PUT (NUMBEROUT, T_TIP, 4, 3);
TTY.PUT (18, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT (18, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end CYLINDRICAL_GIVEN_Q;

procedure RECTANGULAR_NO_OPT(UNITS

CONVERT_DIST : in FLOAT;
LENGTH : in out FLOAT;
LENGTH_UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT_UNITS : in STRING;
WIDTH : in out FLOAT;
WIDTH_UNITS : in STRING;
H : in out FLOAT;
H_UNITS : in STRING;
K : in out FLOAT;
K_UNITS : in STRING;
T_AMBIENT : in out FLOAT;
T_WALL : in out FLOAT;
T_UNITS : in STRING;
Q : in out FLOAT;
Q_UNITS : in STRING) is

NUMBER_OUT
STRING(1..10);

CHAR : CHARACTER;
PAUSE : INTEGER;

PERIMETER, AREA, M, EFFICIENCY,
DELTA_T, T_TIP : FLOAT;
begin

---

Inputs
---

FINOPT_PICTURES.INPUT.MSG;
GET_INPUT(LENGTH, "Length of the rectangular fin", 29,
LENGTH_UNITS, 2, 14);
GET_INPUT(HEIGHT, "Height of the rectangular fin", 29,
HEIGHT_UNITS, 2, 15);
GET_INPUT(WIDTH, "Width of the rectangular fin", 28,
WIDTH_UNITS, 2, 16);
if (UNITS = 2) then
GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
H_UNITS, 19, 17);
GET_INPUT(K, "Thermal conductivity of material, k", 35,
K_UNITS, 17, 18);
else
GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
H_UNITS, 13, 17);
GET_INPUT(K, "Thermal conductivity of material, k", 35,
K_UNITS, 11, 18);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19,
T_UNITS, 5, 19);
GET_INPUT(T_WALL, "Wall Temperature", 16,
T_UNITS, 5, 20);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

---

Calculations (Assume Tip is Insulated) and Length >> Width
---

PERIMETER := 2.0*LENGTH/CONVERT_DIST;
AREA := (WIDTH/CONVERT_DIST)*(LENGTH/CONVERT_DIST);
M := SQRT((H*PERIMETER)/(K*AREA));
DELTA_T := T_WALL-T_AMBIENT;
Q := K*AREA*M*DELTA_T*TANH(M*HEIGHT/CONVERT_DIST);
EFFICIENCY := (TANH(M*HEIGHT/CONVERT_DIST))
/ (M*HEIGHT/CONVERT_DIST);
T_TIP := T_AMBIENT+(DELTA_T/COSH(M*HEIGHT/CONVERT_DIST));

---

Outputs
---

FINOPT_PICTURES.OUTPUT.MSG;
TTY.PUT ( 5, 36, " Inputs ", BRIGHT_WHITE, GREEN);
TTY.PUT ( 7, 1, "Length of the rectangular fin 
= ",
YELLOW, BLACK);
PUT (NUMBER_OUT, LENGTH, 4, 3);
TTY.PUT ( 7 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT ( 7 59, LENGTH_UNITS, YELLOW, BLACK);
TTY.PUT ( 8 1, "Height of the rectangular fin 
= ",
YELLOW, BLACK);
PUT (NUMBER_OUT, HEIGHT, 4, 3);

102
TTY.PUT ( 8, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT ( 8, 59, HEIGHT UNITS, YELLOW, BLACK);
TTY.PUT ( 9, 1, "Width of the rectangular fin = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, WIDTH, 4, 3);
TTY.PUT ( 9, 48, NUMBEROUT, YELLOW, BLACK);
TTY.PUT ( 9, 59, WIDTH UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Convection heat transfer coefficient, h = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, H, 4, 3);
TTY.PUT (10, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (10, 59, H UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Thermal conductivity of material, k = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, K, 4, 3);
TTY.PUT (11, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (11, 59, K UNITS, YELLOW, BLACK);
TTY.PUT (12, 1, "Ambient Temperature = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_AMBIENT, 4, 3);
TTY.PUT (12, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (12, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (13, 1, "Wall Temperature = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_WALL, 4, 3);
TTY.PUT (13, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (13, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (15, 35, " Outputs ", BRIGHT WHITE, GREEN);
TTY.PUT (17, 1, "Heat transferred away by the fin, q = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, Q, 4, 3);
TTY.PUT (17, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (17, 59, Q UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "The fin efficiency = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, EFFICIENCY, 4, 3);
TTY.PUT (18, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (19, 1, "The temperature at the tip = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_TIP, 4, 3);
TTY.PUT (19, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (19, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
end RECTANGULAR_NO_OPT;

procedure RECTANGULAR_GIVEN_VOL(UNITS
INTEGER;
CONVERT DIST : in FLOAT;
VOLUME : in out
FLOAT;
VOLUME UNITS : in STRING;
LENGTH : in out
FLOAT;
LENGTH UNITS : in STRING;
begin

--- Inputs ---

FINOPT_PICTURES.INPUT_MSG;
GET_INPUT(VOLUME, "Volume of the rectangular fin", 29,
VOLUME UNITS, 4, 14);
GET_INPUT(LENGTH, "Length of the rectangular fin", 29,
LENGTH UNITS, 2, 15);
if (UNITS = 2) then
    GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
        H UNITS, 19, 16);
    GET_INPUT(K, "Thermal conductivity of material, k", 35,
        K UNITS, 17, 17);
else
    GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
        H UNITS, 13, 16);
    GET_INPUT(K, "Thermal conductivity of material, k", 35,
        K UNITS, 11, 17);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19,
T UNITS, 5, 18);
GET_INPUT(T_WALL, "Wall Temperature", 16,
Calculations (Assume Tip is Insulated) and Length >> Width

\[
\text{AREA PROFILE} := \frac{(\text{VOLUME} / (\text{CONVERT DIST}^2))}{(\text{LENGTH} / \text{CONVERT DIST})};
\]
\[
\text{WIDTH} := \text{CONVERT DIST} * 0.9977 * ((\text{AREA PROFILE}^2) * H / K)^{(1.0/3.0)};
\]
\[
\text{HEIGHT} := \text{CONVERT DIST} * 1.0023 * ((\text{AREA PROFILE} * K / H)^{(1.0/3.0)});
\]
\[
\text{PERIMETER} := 2.0 \times \text{LENGTH} / \text{CONVERT DIST};
\]
\[
\text{AREA} := (\text{WIDTH} / \text{CONVERT DIST}) \times (\text{LENGTH} / \text{CONVERT DIST});
\]
\[
M := \sqrt{\left(\frac{H \times \text{PERIMETER}}{(K \times \text{AREA})}\right)};
\]
\[
\text{DELTA T} := \text{T WALL} - \text{T AMBIENT};
\]
\[
\text{Q} := K \times \text{AREA} \times M \times \text{DELTA T} \times \tanh\left(M \times \frac{\text{HEIGHT}}{\text{CONVERT DIST}}\right);
\]
\[
\text{EFFICIENCY} := \frac{\tanh\left(M \times \frac{\text{HEIGHT}}{\text{CONVERT DIST}}\right)}{M \times \frac{\text{HEIGHT}}{\text{CONVERT DIST}}};
\]
\[
\text{T TIP} := \text{T AMBIENT} + \frac{\text{DELTA T}}{\cosh\left(M \times \frac{\text{HEIGHT}}{\text{CONVERT DIST}}\right)};
\]

Outputs

FINOPT_PICTURES.OUTPUT_MSG;
TTY.PUT (5, 36, " Inputs ", BRIGHT WHITE, GREEN);
TTY.PUT (7, 1, "Volume of the rectangular fin = ", NUMBEROUT, VOLUME, 4, 3);
TTY.PUT (7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (7, 59, VOLUME UNITS, YELLOW, BLACK);
TTY.PUT (8, 1, "Length of the rectangular fin = ", NUMBEROUT, LENGTH, 4, 3);
TTY.PUT (8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (8, 59, LENGTH UNITS, YELLOW, BLACK);
TTY.PUT (9, 1, "Convection heat transfer coefficient, h = ", NUMBEROUT, H, 4, 3);
TTY.PUT (9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (9, 59, H UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Thermal conductivity of material, k = ", NUMBEROUT, K, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, K UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Ambient Temperature = ", NUMBEROUT, TAMBIENT, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (12, 1, "Wall Temperature = ", NUMBEROUT, TWALL, 4, 3);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
end RECTANGULAR GIVEN_VOL;

procedure RECTANGULAR GIVEN Q(UNITS : in INTEGER;
CONVERT DIST : in FLOAT;
LENGTH : in out FLOAT;
LENGTH UNITS : in STRING;
HEIGHT : in out FLOAT;
HEIGHT UNITS : in STRING;
WIDTH : in out FLOAT;
WIDTH UNITS : in STRING;
H : in out FLOAT;
H UNITS : in STRING;
K : in out FLOAT;
K UNITS : in STRING;
T AMBIENT : in out FLOAT;
T WALL : in out FLOAT;
T UNITS : in STRING;
Q : in out FLOAT;

TTY.PUT (12, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (14, 35, " Outputs ", BRIGHT WHITE, GREEN);
TTY.PUT (16, 1, "Optimum height of the rectangular fin = ",
YELLOW, BLACK);
PUT (NUMBER OUT, HEIGHT, 4, 3);
TTY.PUT (16, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (16, 59, HEIGHT UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "Optimum width of the rectangular fin = ",
YELLOW, BLACK);
PUT (NUMBER OUT, WIDTH, 4, 3);
TTY.PUT (17, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (17, 59, WIDTH UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "Heat transferred away by the fin, q = ",
YELLOW, BLACK);
PUT (NUMBER OUT, Q, 4, 3);
TTY.PUT (18, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (18, 59, Q UNITS, YELLOW, BLACK);
TTY.PUT (19, 1, "The fin efficiency = ",
YELLOW, BLACK);
PUT (NUMBER OUT, EFFICIENCY, 4, 3);
TTY.PUT (19, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (19, 59, EFFICIENCY UNITS, YELLOW, BLACK);
TTY.PUT (20, 1, "The temperature at the tip = ",
YELLOW, BLACK);
PUT (NUMBER OUT, T TIP, 4, 3);
TTY.PUT (20, 48, NUMBER OUT, YELLOW, BLACK);
TTY.PUT (20, 59, T UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);
Q_UNITS

is

NUMBER_OUT : in STRING;
STRING(1..10);
CHAR : CHARACTER;
PAUSE : INTEGER;
PERIMETER, AREA, M, EFFICIENCY,
DELTA_T, T_TIP : FLOAT;

begin

--- Inputs ---
FINOPT_ PICTURES.INPUT_MSG;
if (UNITS = 2) then
  GET_INPUT(Q, "Heat transferred away by the fin, q", 35,
    Q_UNITS, 6, 14);
else
  GET_INPUT(Q, "Heat transferred away by the fin, q", 35,
    Q_UNITS, 1, 14);
end if;
GET_INPUT(LENGTH, "Length of the rectangular fin", 29,
  LENGTH_UNITS, 2, 15);
if (UNITS = 2) then
  GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
    H_UNITS, 19, 16);
  GET_INPUT(K, "Thermal conductivity of material, k", 35,
    K_UNITS, 17, 17);
else
  GET_INPUT(H, "Convection heat transfer coefficient, h", 39,
    H_UNITS, 13, 16);
  GET_INPUT(K, "Thermal conductivity of material, k", 35,
    K_UNITS, 11, 17);
end if;
GET_INPUT(T_AMBIENT, "Ambient Temperature", 19,
  T_UNITS, 5, 18);
GET_INPUT(T_WALL, "Wall Temperature", 16,
  T_UNITS, 5, 19);
TTY.PUT (23, 27, " Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

--- Calculations (Assume Tip is Insulated) and Length >> Width ---
WIDTH := CONVERT_DIST*(0.6321/(H*K))
  *((Q/(LENGTH/CONVERT_DIST))/(T_WALL-T_AMBIENT))**2);
HEIGHT := CONVERT_DIST*0.7978
  *(Q/(LENGTH/CONVERT_DIST))/(H*(T_WALL-T_AMBIENT));
PERIMETER := 2.0*LENGTH/CONVERT_DIST;
AREA := (WIDTH/CONVERT_DIST)*(LENGTH/CONVERT_DIST);
M := SQRT((H*PERIMETER)/(K*AREA))

107
DELTA T := T_WALL - T_AMBIENT;
Q := K*AREA*M*DELTA_T*TANH(M*HEIGHT/CONVERT_DIST);
EFFICIENCY := (TANH(M*HEIGHT/CONVERT_DIST))
/ (M*HEIGHT/CONVERT_DIST);
T_TIP := T_AMBIENT + (DELTA_T / COSH(M*HEIGHT/CONVERT_DIST));

---

Outputs
---

FINOPT_PICTURES.OUTPUT_MSG;
TTY.PUT (5, 36, "Inputs ", BRIGHTWHITE, GREEN);
TTY.PUT (7, 1, "Heat transferred away by the fin, q = ", YELLOW, BLACK);
PUT (NUMBER_OUT, Q, 4, 3);
TTY.PUT (7, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (7, 59, Q_UNITS, YELLOW, BLACK);
TTY.PUT (8, 1, "Length of the rectangular fin = ", YELLOW, BLACK);
PUT (NUMBER_OUT, LENGTH, 4, 3);
TTY.PUT (8, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (8, 59, LENGTH_UNITS, YELLOW, BLACK);
TTY.PUT (9, 1, "Convection h at transfer coefficient, h = ", YELLOW, BLACK);
PUT (NUMBER_OUT, H, 4, 3);
TTY.PUT (9, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (9, 59, H_UNITS, YELLOW, BLACK);
TTY.PUT (10, 1, "Thermal conductivity of material, k = ", YELLOW, BLACK);
PUT (NUMBER_OUT, K, 4, 3);
TTY.PUT (10, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (10, 59, K_UNITS, YELLOW, BLACK);
TTY.PUT (11, 1, "Ambient Temperature = ", YELLOW, BLACK);
PUT (NUMBER_OUT, T_AMBIENT, 4, 3);
TTY.PUT (11, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (11, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (12, 1, "Wall Temperature = ", YELLOW, BLACK);
PUT (NUMBER_OUT, T_WALL, 4, 3);
TTY.PUT (12, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (12, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (14, 35, "Outputs ", BRIGHTWHITE, GREEN);
TTY.PUT (16, 1, "Optimum height of the rectangular fin = ", YELLOW, BLACK);
PUT (NUMBER_OUT, HEIGHT, 4, 3);
TTY.PUT (16, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (16, 59, HEIGHT_UNITS, YELLOW, BLACK);
TTY.PUT (17, 1, "Optimum width of the rectangular fin = ", YELLOW, BLACK);
PUT (NUMBER_OUT, WIDTH, 4, 3);
TTY.PUT (17, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (17, 59, WIDTH_UNITS, YELLOW, BLACK);
TTY.PUT (18, 1, "The fin efficiency = ", YELLOW, BLACK);
PUT (NUMBER_OUT, EFFICIENCY, 4, 3);
TTY.PUT (18, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (19, 1, "The temperature at the tip = ",
YELLOW, BLACK);
PUT (NUMBER_OUT, T_TIP, 4, 3);
TTY.PUT (19, 48, NUMBER_OUT, YELLOW, BLACK);
TTY.PUT (19, 59, T_UNITS, YELLOW, BLACK);
TTY.PUT (23, 27, "Press any key to continue ", BLUE, CYAN);
TTY.GET (PAUSE, CHAR);

end RECTANGULAR_GIVEN_Q;

end FINOPT_SINGLE;
LIST OF REFERENCES


## INITIAL DISTRIBUTION LIST

1. **Defense Technical Information Center**  
   Cameron Station  
   Alexandria, VA 22304-6145  
   
2. **Library, Code 52**  
   Naval Postgraduate School  
   Monterey, CA 93943-5002  
   
3. **Chairman, Code EC**  
   Department of Electrical and Computer Engineering  
   Naval Postgraduate School  
   Monterey, CA 93943-5000  
   
4. **Professor Allan D. Kraus, Code EC/Ks**  
   Department of Electrical and Computer Engineering  
   Naval Postgraduate School  
   Monterey, CA 93943-5000  
   
5. **Professor Matthew D. Kelleher, Code ME/Kk**  
   Department of Mechanical Engineering  
   Naval Postgraduate School  
   Monterey, CA 93943-5000  
   
6. **LT John R. Gensure**  
   Commanding Officer  
   Pearl Harbor Naval Shipyard  
   Box 400  
   Pearl Harbor, HI 96860-5350