

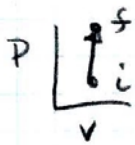
SOLUTIONS TO UNIFIED T2 (WAITZ)

- a) GIVEN TWO PROPERTIES FOR INITIAL COND. THEREFORE STATE IS DEFINED.

$$v_i = \frac{RT_i}{P_i} \quad R_{\text{He}} = \frac{R}{\text{MW}} = \frac{8.314}{4} = 2.078 \frac{\text{kJ}}{\text{kg-K}}$$

$$v_i = \frac{(2.078 \times 10^3)(300)}{100 \times 10^6} = 0.0062 \frac{\text{m}^3}{\text{kg}}$$

GIVEN PATH TO FINAL STATE (RIGID TANK $\Rightarrow \Delta v = 0$)



AND GIVEN FINAL TEMPERATURE.

$$\text{SO } v_f = v_i = 0.0062, \quad T_f = 400\text{K}$$

$$\therefore P_f = \frac{RT_f}{v_f} = \frac{(2078)400}{0.0062} = \boxed{133 \text{ MPa}}$$

- b) NO WORK WAS DONE. $\int P dv = 0$ SINCE $dv = 0$

- c) HERE A NEW PATH IS SPECIFIED, INSTEAD OF $v = \text{CONST}$, WE HAVE $\frac{dp}{dv} = 1 \times 10^5 \frac{\text{MPa}}{\text{m}^3/\text{kg}}$ $\therefore P = 1 \times 10^5 v + \text{CONST.}$

SOLVE FOR CONSTANT USING P_i, v_i , YOU GET

$$P = 1 \times 10^5 \frac{\text{MPa}}{\text{m}^3/\text{kg}} \cdot v - 523.4 \text{ MPa} \quad \left. \vphantom{P} \right\} \text{NEW PATH FULLY DEFINED}$$

BUT NOT GIVEN P_f OR v_f , INSTEAD WE ARE GIVEN T_f SO SUBSTITUTE USING IDEAL GAS

$$P = \frac{RT_f}{v_f} \quad (\text{UNITS Pa})$$

$$\frac{2078 \cdot 400}{v_f} = 1 \times 10^5 v_f - 523.4 \times 10^6$$

(NOTE EXTRA 10^6 TO MAKE UNITS CONSISTENT)
 \leftarrow

REWRITING, $V_f^2 - 5.234 \times 10^{-3} V_f - 8.314 \times 10^{-6} = 0$

BY QUADRATIC FORMULA $V_f = \frac{5.234 \times 10^{-3} \pm 7.788 \times 10^{-3}}{2}$

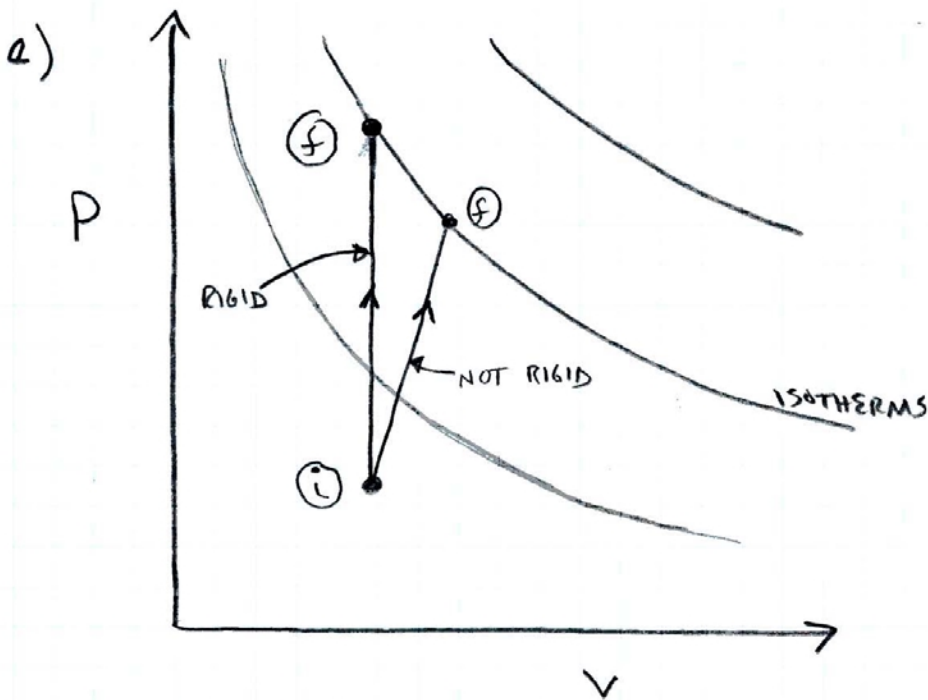
$$V_f = 0.00651 \frac{\text{m}^3}{\text{kg}}$$

$$P_f = \frac{RT_f}{V_f} = \frac{2078.400}{0.00651} = 127.7 \text{ MPa}$$

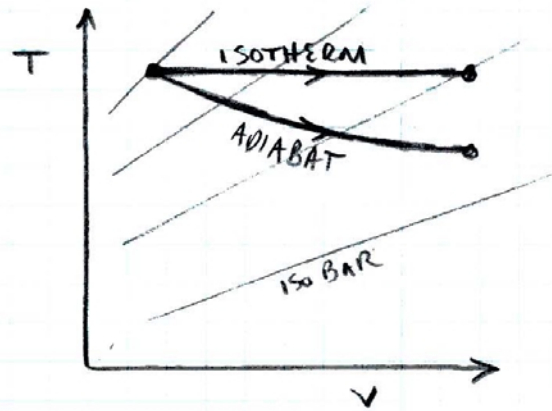
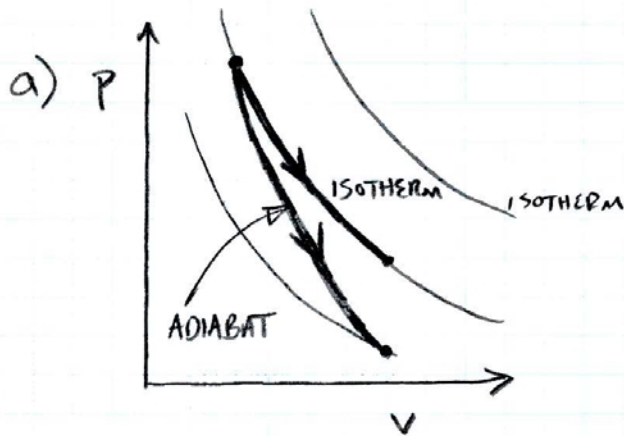
d) $w = \int_{V_i}^{V_2} P dv = \int_{V_i}^{V_2} \left(1 \times 10^5 \frac{\text{MPa}}{\frac{\text{m}^3}{\text{kg}}} \cdot V - 523.4 \text{ MPa} \right) dV$

$$W = \left. \frac{1 \times 10^5}{2} V^2 - 523.4 \times 10^6 V \right|_{V_i}^{V_2}$$

$$W = 31.4 \text{ kJ/kg}$$



SOLUTIONS TO UNIFIED T3 (WAITZ)



b) Q-S ISOTHERMAL

$$T_1 = 300 \text{ K}, v_1 = 1 \text{ m}^3/\text{kg}$$

$$T_2 = 300 \text{ K}, v_2 = 10 \text{ m}^3/\text{kg}$$

$$\therefore P_2 = \frac{287 \cdot 300}{10}$$

$$P_2 = 8610 \text{ Pa}$$

$$c) w = RT \ln\left(\frac{v_2}{v_1}\right) = 287 \cdot 300 \cdot \ln(10)$$

$$w = 198 \text{ kJ/kg}$$

$$\Delta u = q - w = C_v \Delta T = 0$$

$$\therefore q = w = 198 \text{ kJ/kg}$$

Q-S ADIABATIC

$$Pv^\gamma = \text{CONST.} \quad \gamma = 1.4$$

$$P_1 = \frac{287 \cdot 300}{1} = 86100 \text{ Pa}$$

$$\frac{P_2}{P_1} = \left(\frac{v_1}{v_2}\right)^\gamma \quad \therefore$$

$$P_2 = 3428 \text{ Pa} \quad T_2 = \frac{P_2 v_2}{R}$$

$$T_2 = 119 \text{ K}$$

$$\Delta u = q - w$$

$$q = 0$$

$$w = -\Delta u = -C_v(T_2 - T_1)$$

$$= -716.5(119 - 300)$$

$$w = 129 \text{ kJ/kg}$$

G-S ISOTHERMAL

$$\begin{aligned}d) \quad h &= u + pv \\ dh &= C_p dT \\ \Delta h &= 1003.5 (T_2 - T_1) \\ \Delta h &= 0\end{aligned}$$

G-S ADIABATIC

$$\begin{aligned}h &= u + pv \\ dh &= C_p dT \\ \Delta h &= 1003.5 (T_2 - T_1) \\ &= 1003.5 (119 - 300) \\ \Delta h &= -181.6 \text{ kJ}\end{aligned}$$

Q) HEAT IS A TRANSFER OF ENERGY ACROSS A SYSTEM BOUNDARY BY VIRTUE OF A TEMPERATURE DIFFERENCE ONLY. IT IS MEASURED IN JOULES

TEMPERATURE IS A THERMODYNAMIC PROPERTY AND A FUNCTION OF THE STATE OF A SYSTEM. IT IS MEASURED IN KELVIN.

- * IT IS POSSIBLE TO HAVE AN ISOTHERMAL PROCESS WITH HEAT TRANSFER
- * IT IS POSSIBLE TO HAVE AN ADIABATIC PROCESS WITH A TEMPERATURE CHANGE

AS DEMONSTRATED IN THIS PROBLEM

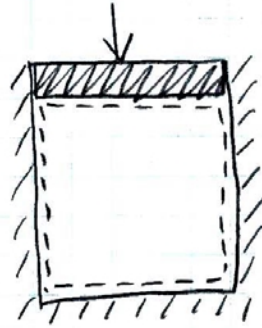
SOLUTIONS TO T4 (WAITZ)

a) QUASI-STATIC, ADIABATIC COMPRESSION

$$P_1 = 100 \text{ kPa}, \quad P_2 = 500 \text{ kPa}$$

$$T_1 = 300 \text{ K}$$

$$V_1 = \frac{287 \cdot 300}{100,000} = 0.861 \frac{\text{m}^3}{\text{kg}}$$



$$PV^\gamma = \text{CONST.} \quad (\text{FOR WHOLE PROCESS}) \quad \therefore \frac{P_2}{P_1} = \left(\frac{V_1}{V_2}\right)^\gamma \quad 5 = \left(\frac{0.861}{V_2}\right)^{1.4}$$

$$\therefore \boxed{V_2 = 0.273}$$

$$T_2 = \frac{500,000 \cdot 0.273}{287} = \boxed{475 \text{ K}}$$

$$\Delta u = \cancel{q} - w \quad \Delta u = -w, \quad C_v \Delta T = -w$$

$$w = -716.5(475 - 300) = \boxed{-125 \text{ kJ/kg}}$$

BY THE SYSTEM - NEGATIVE
SINCE ENERGY TRANSFERRED
TO SYSTEM.

b) INITIAL STATE IS THE SAME.
KNOW $P_f = 500 \text{ kPa}$ BUT DON'T KNOW T_f OR V_f

* DON'T KNOW BEHAVIOR OF STATE OF SYSTEM
DURING PROCESS \rightarrow BUT FIRST LAW STILL HOLDS.

* HOWEVER, NOW WE MUST USE EXTERNAL INFORMATION
TO RELATE PROPERTIES AT INITIAL AND FINAL STATE

$$1^{\text{ST}} \text{ LAW: } \Delta u = \overset{\nabla^{\circ}}{q} - w$$

(STILL INSULATED)

$$C_v \Delta T = -w = \underline{\underline{-p_{\text{ext}} \Delta V}}$$

$$1) C_v (T_f - T_i) = -p_{\text{ext}} (V_f - V_i)$$

AND IDEAL GAS LAW HOLDS AT INITIAL CONDITION AND FINAL CONDITION (BUT NOT IN BETWEEN)

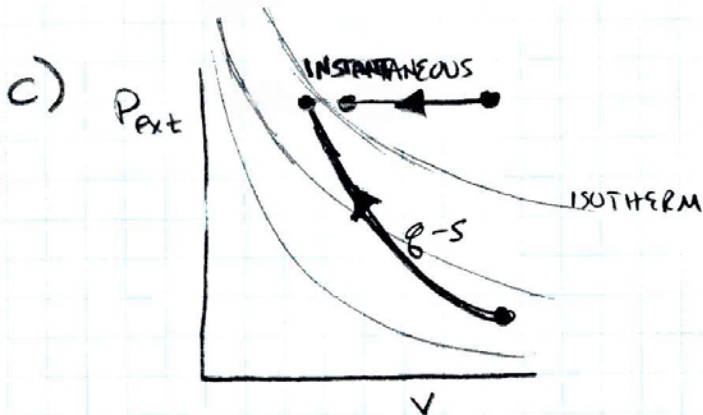
$$2) p_f V_f = RT_f \quad \text{WHERE } p_f = p_{\text{ext}}$$

2 EQNS IN 2 UNKNOWNNS (T_f, V_f)

$$\text{SUBSTITUTING, } \left\{ \begin{array}{l} T_f = \frac{p_{\text{ext}} V_i + C_v T_i}{C_v + R} = \boxed{643.2 \text{ K}} \\ V_f = \frac{287 \cdot 643.2}{500,000} = \boxed{0.369 \text{ m}^3/\text{kg}} \end{array} \right.$$

ENDED UP AT A DIFFERENT STATE!

$$W = -C_v \Delta T = -716.5 (643.2 - 300) = \boxed{-245.9 \text{ kJ/kg}}$$



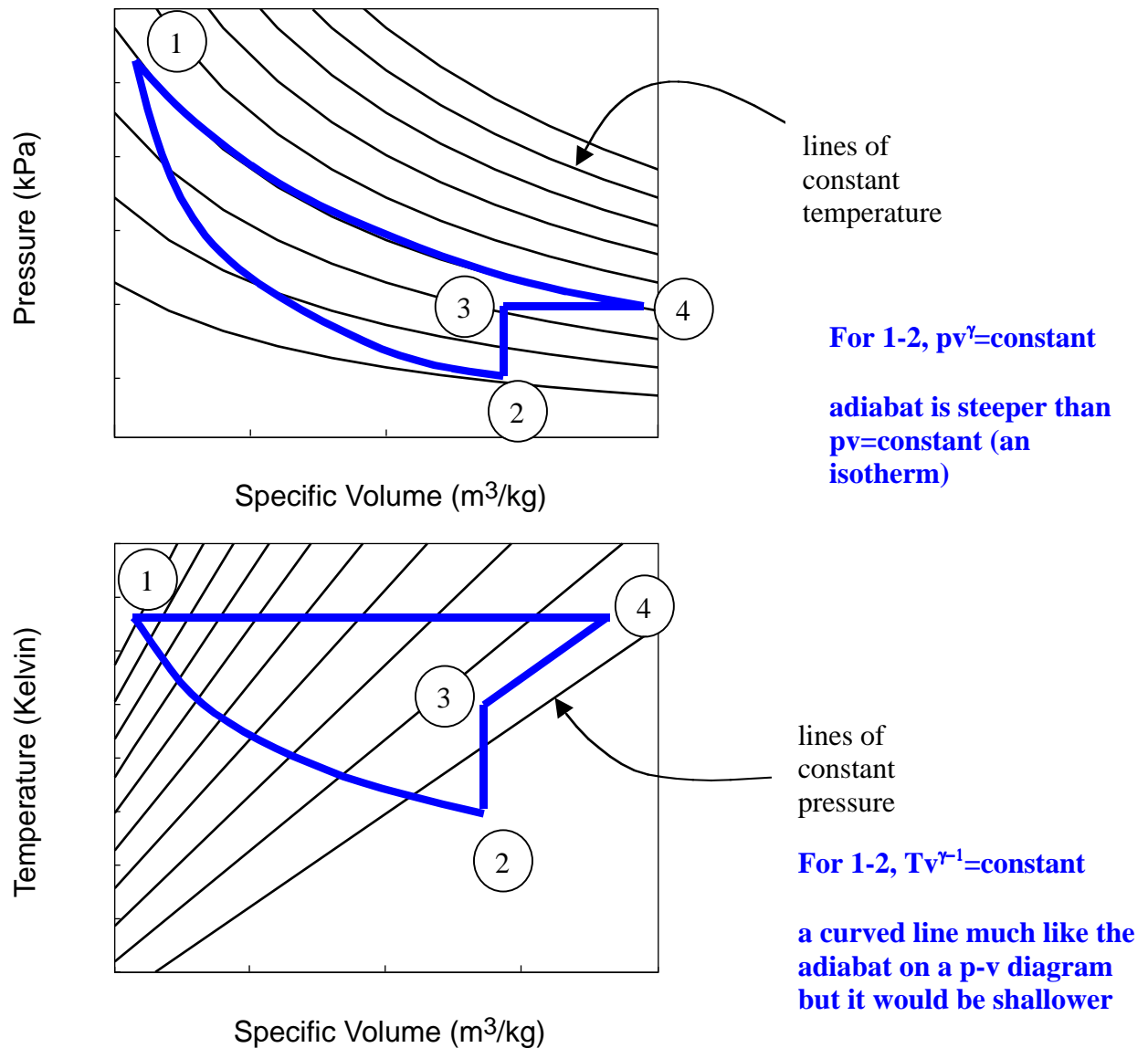
LESS WORK REQUIRED TO COMPRESS THE GAS TO THE SAME PRESSURE VIA A Q-S PROCESS.
∴ MORE EFFICIENT

T5 SOLUTIONS (Waitz)

a) Draw a thermodynamic cycle on p-v and T- v diagrams consisting of

- Leg 1-2: adiabatic expansion
- Leg 2-3: constant volume heat addition
- Leg 3-4: constant pressure expansion
- Leg 4-1: isothermal compression

Assume that all processes are quasi-static and involve an ideal gas.



b) For each leg determine if the heat and work transfers are (+), (-), or zero.

| | Q (+, -, or zero) | W (+, -, or zero) |
|---------|-------------------|-------------------|
| Leg 1-2 | 0 | + |
| Leg 2-3 | + | 0 |
| Leg 3-4 | + | + |
| Leg 4-1 | - | - |

c) Is the net work for this cycle positive or negative?

The net work for this cycle is negative. The area under the expansion process is less than the area under the compression processes.

d) What common purpose might you use a cycle like this for and why?

This cycle could serve as a cooler or refrigerator. Overall it takes in energy in the form of heat from cold temperatures and expels energy in the form of heat from high temperatures. The net work for the cycle as a whole is negative, meaning that energy is put into the system to enable these transfers of heat to take place.

C2 Solutions

1. Modify the “Hello” program shown in class (Lecture C2) to display the following text on the screen:

```
Hello World
My name is Your Name
```

GNAT 3.13p (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/joeb/desktop/16070/concep~1/hello_world.adb (source file time stamp: 2003-09-10 08:55:28)

```
1. -----
2. -- Program : To Display "Hello World
3. --           My Name is Joe B"
4. -- Programmer : Joe B
5. -- Date Last Modified : 09/10/2003
6. -----
7.
8.
9. with Ada.Text_Io;
10.
11. procedure Hello_World is
12.
13. begin -- Hello_World
14.
15.   Ada.Text_Io.Put(Item => "Hello World ");
16.   Ada.Text_Io.New_Line;
17.   Ada.Text_Io.Put(Item => "My Name is Joe B");
18.
19. end Hello_World;
```

19 lines: No errors

2. There are two errors that are seen:

- i. “raised SCREEN.WIN32_FILL_SCREEN_ERROR : screen.adb:99” is seen when the output is redirected to the file.
- ii. “raised SPIDER.HIT_THE_WALL: spider.adb:224” is seen when the output is only displayed on the screen

3. Write an algorithm to use the Feldman “spider package” to draw an inverted triangle as shown below. Turn in a hard copy of your code listing and an electronic copy of your code.

```
RRRRRRR
 R   R
  R  R
   R
    R
```

Problem Analysis:

The inverted triangle consists of 4 lines with the following features:

- i. The top line consists of 7 symbols with no gaps between them
- ii. The following lines have (n-1) blank spaces, followed by a symbol, followed by (7-2n) blank spaces and one symbol (if $7-2n > 0$)

Algorithm:

1. Face the spider East
2. Set Spider color to Red
3. Move spider East 7 steps
4. for I in 2 .. 4 loop
 - i. Set Spider color to None
 - ii. Set Spider direction to South
 - iii. Move Spider 1 step
 - iv. Set Spider direction to West
 - v. Move Spider 7 steps
 - vi. Set Spider direction East
 - vii. Move spider (I-1) spaces
 - viii. Set Spider color to Red
 - ix. Move spider one step in the same direction as last move
 - x. Set Spider color to Black
 - xi. Move spider (n-2*I) steps
 - xii. If (n-2*I) > 0 then
 1. Set Spider color to Red
 2. Move one step
 3. Set Spider color to none
 4. move I-1 steps
5. Stop program execution

Code Listing:

GNAT 3.13p (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/joeb/desktop/adatex~1/pset1~1/spider_triangle.adb (source file time stamp: 2003-09-10 10:28:10)

```
1. with Spider;
2. procedure Spider_Triangle is
3. -----
4. --| Program : This program demonstrates display an inverted triangle.
5. --| Programmer : Joe B
6. --| Date Last Modified : 09/10/2003
7. -----
8. N : Integer;
9.
10. begin -- Spider_Triangle
```

```

11. -- initialise the number of lines R's per line to be 7
12. N:=7;
13. Spider.Start;
14. -- initialize the direction to be east and set the symbol to Red
15. Spider.Face(Whichway => Spider.East);
16. Spider.Changecolor(Newcolor => Spider.Red);
17. -- draw the top line with n Red Symbols
18. for I in 1..N loop
19.   Spider.Step;
20. end loop;
21. -- the number of lines for n symbols is (n/2)+1 for n odd
22. for I in 2 .. ((N/2)+1) loop
23.   -- move the spider down to the next line
24.   Spider.Face(Whichway => Spider.South);
25.   Spider.Changecolor(Newcolor => Spider.None);
26.   Spider.Step;
27.   -- face the opposite direction and trace back to the starting point
28.   Spider.Face(Whichway => Spider.West);
29.   for J in 1..N loop
30.     Spider.Step;
31.   end loop;
32.   -- turn the spider back in the right direction
33.   Spider.Face(Whichway => Spider.East);
34.   -- draw the required number of blank spaces
35.   for J in 1 .. I-1 loop
36.     Spider.Step;
37.   end loop;
38.   -- change the symbol to Red
39.   Spider.Changecolor(Newcolor => Spider.Red);
40.   Spider.Step;
41.   -- return the symbol to none
42.   Spider.Changecolor(Newcolor => Spider.None);
43.   -- draw the required number of blank spaces
44.   for J in 1 .. (N-2*I) loop
45.     Spider.Step;
46.   end loop;
47.   -- check to ensure that it is not the last line
48.   if (N-2*I > 0) then
49.     -- change the symbol to red and draw the symbol
50.     Spider.Changecolor(Newcolor => Spider.Red);
51.     Spider.Step;
52.     -- reset the symbol to none and draw the required number of blank spaces
53.     Spider.Changecolor(Newcolor => Spider.None);
54.     for J in 1 .. I-1 loop
55.       Spider.Step;
56.     end loop;
57.   end if;
58.
59. end loop;
60. Spider.Quit;
61. end Spider_Triangle;

```

61 lines: No errors

C3 Solutions

1. Distance_With_Errors Listing File

GNAT 3.13p (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/joeb/desktop/adatex~1/fk3-w95/distance_with_errors.adb (source file time stamp: 1998-09-13 23:11:36)

```
1. WITH Ada.Text_IO;
2. WITH Ada.Float_Text_IO;
3. PROCEDURE Distance_with_Errors IS
4. -----
5. --| Finds distance, given travel time and average speed
6. --| Author: Michael Eldman, The George Washington University
7. --| Last Modified: June 1998
8. -----
9.   HowLong : Natural;
10.  HowFast : Float;
11.  HowFar : Natural;
12.
13. BEGIN -- Distance_with_Errors
14.
15.   -- prompt user for hours and average speed
16.   Ada.Text_IO.Put
17.     (Item => "How long will you be driving (integer) ? ");
18.   Ada.Float_Text_IO.Get (Item => HowLong);
      |
      >>> invalid parameter list in call (use -gnatf for details)
19.   Ada.Text_IO.Put
20.     (Item => "At what speed (miles per hour, integer)?");
21.   Ada.Float_Text_IO.Get (Item => HowFast);
22.
23.   -- compute distance driven
24.   HowFast := HowLong * HowFar;
      |
      >>> expected type "Standard.Float"
      >>> found type "Standard.Integer"
25.
26.   -- display results
27.   Ada.Text_IO.Put (Item => "You will travel about ");
28.   Ada.Float_Text_IO.Put (Item => HowFar);
      |
      >>> invalid parameter list in call (use -gnatf for details)
      >>> possible missing instantiation of Text_IO.Integer_IO
29.   Ada.Text_IO.Put (Item => " miles");
30.   Ada.Text_IO.New_Line;
31.
32. END Distance_with_Errors;
33.
33 lines: 5 errors
```

2. Distance_With_Errors with bug fixes

GNAT 3.13p (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/joeb/desktop/adatex~1/fk3-w95/distance_with_errors.adb (source file time stamp: 2003-09-10 10:41:38)

```
1. WITH Ada.Text_IO;
2. with Ada.Float_Text_Io;
3. with Ada.Integer_Text_IO;
4. PROCEDURE Distance_with_Errors IS
5. -----
6. -- Finds distance, given travel time and average speed
7. --| Author: Michael eldman, The George Washington University
8. --| Last Modified: June 1998
9. -----
10. HowLong : Natural;
11. HowFast : Float;
12. HowFar : Natural;
13.
14. BEGIN -- Distance_with_Errors
15.
16. -- prompt user for hours and average speed
17. Ada.Text_IO.Put
18.   (Item => "How long will you be driving (integer) ? ");
19. Ada.Integer_Text_IO.Get (Item => HowLong);
20. Ada.Text_IO.Put
21.   (Item => "At what speed (miles per hour, integer)?");
22. Ada.Float_Text_IO.Get (Item => HowFast);
23.
24. -- compute distance driven
25. HowFar:= HowLong * Integer(HowFast);
26.
27. -- display results
28. Ada.Text_IO.Put (Item => "You will travel about ");
29. Ada.Integer_Text_IO.Put (Item => HowFar);
30. Ada.Text_IO.Put (Item => " miles");
31. Ada.Text_IO.New_Line;
32.
33. END Distance_with_Errors;
34.
```

34 lines: No errors

3. Write an algorithm to
 - a. Accept the weight of the user (in kilograms)
 - b. Compute the equivalent weight in pounds
 - c. Display “weight_in_kg” kg = “weight_in_pounds” lb

Where weight_in_kg is the entered value and weight_in_pounds is the computed value.

1. Prompt the user to enter his/her weight.
2. Read the user input.
3. Convert the weight from kilograms into pounds using the formula
1 Pound = 0.453592 kilograms

$$\text{Weight in pounds} = \text{Weight in Kilograms} / 0.453592$$

4. Display the output to the user in the weight_in_kg” kg = “weight_in_pounds” lb format.

4. Code listing of the implementation of the algorithm.

GNAT 3.13p (20000509) Copyright 1992-2000 Free Software Foundation, Inc.

Compiling: c:/docume~2/joeb/desktop/16070/concep~1/convert_weight.adb (source file time stamp: 2003-09-10 18:46:40)

```
1. -----
2. -- Program : To Convert the user weight in kilograms
3. --           into pounds.
4. -- Programmer : Joe B
5. -- Date Last Modified : 09/10/2003
6. -----
7.
8.
9. with Ada.Text_IO;
10. with Ada.Float_Text_IO;
11.
12. procedure Convert_Weight is
13.
14.   Weight_In_Kg, Weight_In_Lb : Float;
15.   -- set the conversion factor to convert between kilograms and pounds
16.   Conversion_Factor : constant Float := 0.453592;
17.
18. begin -- Convert_Weight
19.   -- get user input
20.   Ada.Text_IO.Put(Item => "Please Enter Your Weight in Kilograms ");
21.   Ada.Float_Text_IO.Get(Item => Weight_In_Kg);
22.   Ada.Text_IO.Skip_Line;
23.
24.   Ada.Text_IO.New_Line;
25.   -- perform the conversion
26.   Weight_in_lb := Weight_in_Kg / Conversion_Factor;
27.
28.   -- display the computed result to the user
29.   Ada.Float_Text_IO.Put(Item => Weight_In_Kg, Fore => 4, Aft => 3, Exp => 0);
30.   Ada.Text_IO.Put(Item => " kg = ");
31.
32.   Ada.Float_Text_IO.Put(Item => Weight_In_lb, Fore => 4, Aft => 3, Exp => 0);
33.   Ada.Text_IO.Put(Item => " lb");
34.
35. end Convert_Weight;
```

35 lines: No errors

C4 Solutions

1. Convert '2 + 3 = 5' into ASCII

| | | |
|-----|---|----|
| '2' | - | 50 |
| ' ' | - | 32 |
| '+' | - | 43 |
| ' ' | - | 32 |
| '3' | - | 51 |
| '=' | - | 61 |
| '5' | - | 53 |

2. Convert the following binary numbers into hexadecimal.

a. $\frac{0000}{0}$ $\frac{1111}{F}$ $\frac{0000}{0}$ $\frac{1111}{F}$

b. $\frac{0011}{3}$ $\frac{0011}{3}$ $\frac{0000}{0}$ $\frac{0000}{0}$ $\frac{1000}{8}$ $\frac{0000}{0}$

c. $\frac{0000}{0}$ $\frac{1010}{A}$ $\frac{1010}{A}$ $\frac{0000}{0}$