Home Work 11

The problems in this problem set cover lectures C13 and C14

1.

a. Define a robust algorithm to carry out integer division using repeated subtraction. Your algorithm accepts two integers and returns the quotient and the remainder. Hint: What are the preconditions and postconditions of your algorithm?

Precondtions: Two integers *x*,*y*
y is non-zero

Algorithm:

```
Set R to absolute_value(x)
Set Q to zero
While R >= absolute_value (y)
        Increment Q
        R := R- absolute_value(y)
If either x or y are negative
        If both x and y are negative
                Set R to –R
        else
                if x is negative
                        Set R to –R
                Set Q to –Q
Display Q and R
```

Postconditions: Q contains the quotient
R contains the remainder
$x = Q*y + R$, $abs(R) < abs(Q)$

b. Implement your algorithm as an Ada95 program, using exception handling to provide robustness.

```
1. ------------------------------------------------------------
2. -- Procedure to carry out robust division
3. -- Programmer: Jayakanth Srinivasan
4. -- Date Last Modified : April 17,2004
5. ------------------------------------------------------------
6.
7. with Ada.Text_Io;
8. with Ada.Integer_Text_Io;
9. use Ada.Text_Io;
10. use Ada.Integer_Text_Io;
11.
12. procedure Robust_Division is
13.    X,
14.    Y,
15.    Q,
16.    R             : Integer;
17.    Divide_By_Zero : exception;
18.
19. begin
20.    loop
21.       Ada.Text_IO.Skip_Line;
22.       begin
23.          -- get the dividend (X)
24.          Ada.Text_Io.Put("Please Enter the X : ");
25.          Ada.Integer_Text_Io.Get(X);
26.          Ada.Text_Io.Skip_Line;
27.
28.          -- get the divisor (Y)
29.          Ada.Text_Io.Put("Please Enter the Y : ");
30.          Ada.Integer_Text_Io.Get(Y);
31.          Ada.Text_Io.Skip_Line;
32.
33.          if Y = 0 then
34.             raise Divide_By_Zero;
35.          end if;
36.
37.          --set the remainder to absolute value of X
38.          R :=abs(X);
39.          -- set quotient to zero
40.          Q := 0 ;
41.          -- while remainder is greater than absolute value of y
42.          while  R >= abs(Y)  loop
43.             -- deduct absolute value of y from the remainder
44.             R := R - abs(Y) ;
45.             -- increment the quotient
46.             Q := Q + 1;
47.          end loop;
48.
49.          --ensure that the sign on the quotient is quotient
50.          if (X<0) or (Y<0) then
51.             if (X<0) and (Y<0) then
52.                -- if both x,y are negative then remainder is negative
53.                R := -1*R;
```

```
54.            else
55.              if (X<0) then
56.                -- if X is negative then remainder is negative
57.                R:= -1*R;
58.              end if;
59.              -- if either x or y not both, then quotient is negative
60.              Q := -1*Q;
61.            end if;
62.         end if;
63.         -- Display the quotient
64.         Ada.Text_Io.Put_Line(Integer'Image(Q));
65.
66.         -- display the remainder
67.         Ada.Text_Io.Put_Line(Integer'Image(R));
68.
69.         -- if the program has reached this part, there were no exceptions
70.         exit;
71.
72.
73.       exception
74.         when Data_Error =>
75.           Ada.Text_Io.Put_Line("Trying to enter a non-integer");
76.
77.         when Divide_By_Zero =>
78.           Ada.Text_Io.Put_Line("Trying to divide by zero");
79.
80.         when others =>
81.           Ada.Text_Io.Put_Line("Dont know what this exception is");
82.
83.           -- this is the end of the block created by the begin statement
84.       end;
85.       -- this is the end of the loop
86.     end loop;
87.
88. end Robust_Division;
```

88 lines: No errors

2.

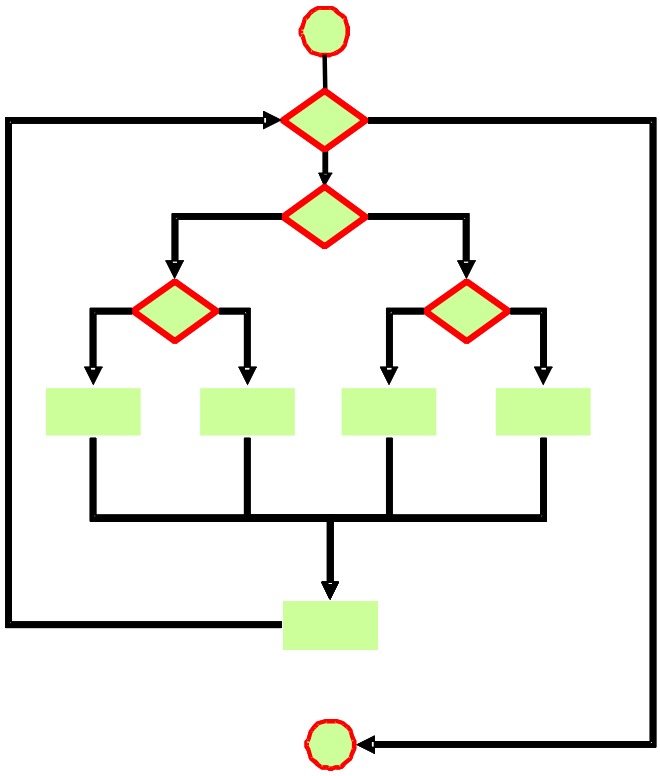
a. What is the cyclomatic complexity of the code fragment shown below?

```
loop
  exit when Flag := True;

  if A < 100 and B > 200 then
    if A > 50 then
      Sum := Sum +2;
    else
      Sum := Sum +1;
    end if;
  else
    if B < 300 then
      Sum:= Sum -1;
    else
      Sum := Sum -2;
    end if;
  end if;

end loop;
```

Hint: Draw the control flow graph



11 Nodes, 14 edges => Cyclomatic complexity = 5.

b. What is the minimum number of test cases needed to test the fragment of code shown below? Justify your answer.

```
1. if A < 100 and B > 200 then
2.       if A > 50 then
3.          Sum := Sum +2;
4.       else
5.          Sum := Sum +1;
6.       end if;
7.    else
8.       if B < 300 then
9.          Sum:= Sum -1;
10.      else
11.            Sum := Sum -2;
12.      end if;
13. end if;
```

Test Case	A	B	Line Tested
1	50 < A < 100	B > 200	Sum:=Sum+2
2	A <= 50	B > 200	Sum:=Sum+1
3	A >=100	B < 300	Sum:=Sum-1
4	Any Other combination of A and B		Sum:=Sum-2

UNIFIED PROPULSION P5 SOLUTIONS — WAITZ

$$P_{AVAIL} - P_{REQ} = \Delta P.E. + \Delta K.E.$$

$$TV - DV = W\frac{dh}{dt} + \frac{d}{dt}\left(\frac{1}{2}\frac{W}{g}V^2\right)$$

CASE i) STEADY, NO CLIMB $T = D$ $L = nW = 5W$

$$\frac{T}{W} = \frac{D}{W} \qquad D = \frac{1}{2}\rho V^2 S k C_L^2 + \frac{1}{2}\rho V^2 S C_{D_0}$$

$$\frac{T}{W} = \frac{1}{2}\rho V^2\left(\frac{1}{W/S}\right)k\frac{L^2}{\left(\frac{1}{2}\rho V^2 S\right)^2} + \frac{1}{2}\rho V^2\left(\frac{1}{W/S}\right)C_{D_0}$$

$$\frac{T}{W} = \cancel{\frac{1}{2}\rho V^2}\left(\cancel{\frac{1}{W/S}}\right)k\frac{n^2W^2}{\left(\frac{1}{2}\rho V^2\right)^{\cancel{2}}S^{\cancel{2}}} + \frac{1}{2}\rho V^2\left(\frac{1}{W/S}\right)C_{D_0}$$

$$\frac{T}{W} = \frac{kn^2}{\frac{1}{2}\rho V^2}\left(\frac{W}{S}\right) + \frac{C_{D_0}}{\left(\frac{W}{S}\right)}\frac{1}{2}\rho V^2$$

$$= 1.53 \qquad + 0.034$$

$V = 0.9(295) = 265.5\ m/s$
$\rho = 0.34\ kg/m^3$
$k = 0.21$
$C_{D_0} = 0.01$
$W/S = 3500$

$$\boxed{\frac{T}{W} = 1.56}$$

CASE ii) $TV - DV = \frac{d}{dt}\left(\frac{1}{2}\frac{W}{g}V^2\right) = \frac{1}{2}\frac{W}{g}2V\frac{dV}{dt}$

$$\frac{T}{W} = \frac{D}{W} + \frac{1}{g}\frac{dV}{dt}$$

$$\frac{T}{W} = \underbrace{\frac{kn^2}{\frac{1}{2}\rho V^2}\left(\frac{W}{S}\right) + \frac{C_{D_0}}{\left(\frac{W}{S}\right)}\frac{1}{2}\rho V^2}_{\text{from above (now } n=1)} + \frac{1}{g}\frac{(V_{final} - V_{initial})}{\Delta t}$$

UNIFIED PROPULSION P6 SOLUTIONS — WAITZ

a) $I_{sp} = \frac{u_e}{g} = \frac{1}{g}\left[M_e\sqrt{\gamma R T_c\left(\frac{1}{1+\frac{\gamma-1}{2}M_e^2}\right)}\right]$

NEED TO ITERATE TO FIND M_e FOR GIVEN $\frac{A^*}{A_e}$

$$\frac{A_e}{A^*} = \frac{0.01}{0.0006} = 16.67 = \frac{1}{M_e}\left[\frac{1+\frac{\gamma-1}{2}M_e^2}{\frac{\gamma+1}{2}}\right]^{\frac{\gamma+1}{2(\gamma-1)}} \Rightarrow M_e = 4.27$$

$\therefore I_{sp} = 271$ s $\quad (u_e = 2654$ m/s$)$

$T = \dot{m} u_e + A_e(p_e - p_a)$ $\quad p_a$ = ATMOSPHERIC PRESSURE AT LAUNCH

$$\dot{m} = \sqrt{\gamma}\left(\frac{2}{\gamma+1}\right)^{\frac{\gamma+1}{2(\gamma-1)}}\frac{p_c}{\sqrt{RT_c}}A^* = 1.85 \text{ kg/s}$$

$$p_e = \frac{p_c}{\left[1+\frac{\gamma-1}{2}M_e^2\right]^{\gamma/\gamma-1}} = 20.0 \text{ kPa}$$

$\therefore T = 4102$ N (AT LAUNCH) $\quad [= 4915$ N if you neglect pressure term$]$

b) AT BURNOUT $u_{bo} = g\left[I_{sp}\ln\left(\frac{m_{initial}}{m_{final}}\right) - t_{bo}\right]$

$$h_{bo} = g\left[-t_{bo}I_{sp}\frac{\ln\left(\frac{m_i}{m_f}\right)}{\left(\frac{m_i}{m_f}-1\right)} + t_{bo}I_{sp} - \frac{1}{2}t_{bo}^2\right]$$

$t_{bo} = m_{propellant}/\dot{m} = 35\text{kg}/2.14\text{ kg/s} = 18.9$ s

so $\quad u_{bo} = 3010.$ m/s $\quad h_{bo} = 22532$ m

GLIDE PHASE $\quad \frac{1}{2}m_{final}u_{final}^2 = m_{final}g\Delta h$

$\Delta h = 462370$ m

TOTAL HEIGHT $= h_{bo} + \Delta h =$ **485 km**

UNIFIED PROPULSION P7 SOLUTIONS — WAITZ 2001

a) $\frac{T_{T_3}}{T_0} = \frac{T_{T_3}}{T_{T_2}} \cdot \frac{T_{T_2}}{T_{T_1}} \cdot \frac{T_{T_1}}{T_0}$

$\frac{T_{T_3}}{T_{T_2}} = \tau_c$; $\frac{T_{T_2}}{T_{T_1}}$ assume inlet is isentropic $= 1$; $\frac{T_{T_1}}{T_0} = 1 + \frac{\gamma-1}{2} M_0^2$

COMPOSED OF TEMPERATURE RISE DUE TO RAM EFFECT + TEMPERATURE RISE DUE TO WORK ADDITION IN COMPRESSOR

ANY COMBINATION OF T_0 & M_0 SUCH THAT

$$T_0 \cdot \tau_c \cdot \left[1 + \frac{\gamma-1}{2} M_0^2\right] > 1000\,K$$ WILL EXCEED THE TEMPERATURE LIMIT.

$\tau_c = (\pi_c)^{\frac{\gamma-1}{\gamma}} = (25)^{\frac{\gamma-1}{\gamma}} = 2.508$

FOR

$T_0 = 300K$ LIMIT IS EXCEEDED FOR $M \gtrsim 1.3$

$T_0 = 216K$ LIMIT IS EXCEEDED FOR $M \gtrsim 2.1$

b) PLUGGING INTO EQNS:

$M = 1.2$, $T = 300K$, $\eta_0 = 0.367$, $\eta_{TH} = 0.69$, $\eta_P = 0.53$

$M = 1.2$, $T = 216K$, $\eta_0 = 0.298$, $\eta_{TH} = 0.69$, $\eta_P = 0.43$

- η_{TH} DOESN'T CHANGE $\eta_{TH} = 1 - \frac{T_0}{T_{T_3}} = 1 - \frac{1}{\tau_c\left[1 + \frac{\gamma-1}{2} M_0^2\right]}$ (from above)

  IS INDEPENDENT OF T_0

- η_P CHANGES BECAUSE AT CONSTANT M_0, $u_0 \downarrow$ AS $T_0 \downarrow$

c) $T_{T_4} = 1800K$, $F/\dot{m}a_0 = 3.16$, $\eta_0 = 0.298$, $\eta_{TH} = 0.69$, $\eta_P = 043$

$T_{T_4} = 1400K$, $F/\dot{m}a_0 = 2.36$, $\eta_0 = 0.348$, $\eta_{TH} = 0.69$, $\eta_P = 0.54$

$\eta_0 \uparrow \Rightarrow$ IMPROVES RANGE & ENDURANCE

$F/\dot{m}a_0 \downarrow \Rightarrow$ REDUCED MANEUVERABILITY

Problem S13 Solutions

1.

$$G_1(j\omega) = \frac{-a^2}{(j\omega - a)(j\omega + a)}$$

2.

$$A_1(\omega) = \frac{a^2}{\sqrt{a^2+\omega^2}\sqrt{a^2+\omega^2}} = \frac{a^2}{(a^2+\omega^2)}$$

$$\phi_1(\omega) = -\tan^{-1}\left(-\frac{\omega}{a}\right) - \tan^{-1}\left(\frac{\omega}{a}\right) = 0$$

3.

$$G_2(j\omega) = \frac{a^2}{(j\omega + a)^2}$$

$$A_2(\omega) = \frac{a^2}{(a^2+\omega^2)} \quad A_1(\omega)$$

$$\phi_2(\omega) = -2\tan\left(\frac{\omega}{a}\right) =$$

4.

Write $\cos\omega t = \frac{e^{j\omega t} + e^{-j\omega t}}{2}$

$$Y = |G_1(j\omega)|\frac{e^{j\omega t}}{2}\left(e^{j\phi_1(\omega)}\right) + |G_1(-j\omega)|\frac{e^{-j\omega t}}{2}\left(e^{-j\phi_1(\omega)}\right)$$

$$A_1(\omega)\left(\frac{e^{j(\omega t+\phi_1(\omega))} + e^{-j(\omega t+\phi_1(\omega))}}{2}\right)$$

$$y_1(t) = A_1(\omega)\cos\left(\omega t + \phi_1(\omega)\right) =$$

likewise for $y_2(t)$

5.

The filters both have the same effect on the magnitude of the input, $A_1(\omega) = A_2(\omega)$.

6.

The non-causal filter produces no phase shift, while the phase shift of the causal filter is between 0 and -180^o, depending on ω.

7.

The non-causal filter produces no phase shift. Therefore, setting the input is easier and the waveform will arrive at the next stage on time. Signals with multiple frequency components would be jumbled due to the variance of pure phase shift at each frequency of the causal filter. The non-causal filter will scale each frequency but produce no phase shift, thereby making an effective multiple frequency low-pass filter.