Federal Urdu University of Arts, Sciences & Technology, Karachi

Gulshan-e-Iqbal Campus

Name:	**Shoaib Ahmed**	
Father's Name:	**Ahmed Hussain**	
Major Dept:	**Science & Technology**	
Dept:	**Computer Science**	
Class/Section:	**BS-6 / C**	**Morning**
Batch:	**17**	**Regular**
Roll:	**27**	
Course:	**Numerical Computing**	
Subject:	**Assignment 1 (Chapter 2)**	
Instructor:	**Prof. Syed Akhter Raza**	

2.3 Develop, debug, and document a program to determine the roots of a quadratic equation, $ax^2 + bx + c$, in either a high-level language or a macro language of your choice. Use a subroutine procedure to compute the roots (either real or complex). Perform test runs for the cases (a) a = 1, b = 6, c = 2; (b) a = 0, b = 24, c = 1.6; (c) a = 3, b = 2.5, c = 7.

Sol: **<u>Algorithm:</u>**

1. Start.
2. Input values of a,b,c .
3. Calculate d = b^2-4*a*c.
4. If (d < 0) . calculate x1 = (-b + i?n) / 2a and x2 = (-b-i?n / 2a)
   Else if (d = 0) . calculate x1 = x2 = (-b / 2a)
   Else . calculate x1 = -b + ?d / 2a and x2 = -b - ?d / 2a
5. Print x1 and x2
6. End

<u>Excel Sheet:</u>

Problem No: 2.3

Quadratic Equation				Quadratic Formula	
ax^2+bx+c=0				x=(-b+/-sqrt(b^2-4ac))/(2a)	
a	**b**	**c**			
1	6	2		x1	-0.354248689
				x2	-5.645751311
0	-4	1.6		x1	-0.4
3	2.5	7		x1	-0.416667 + 1.4696i
				x2	-0.416667 + 1.4697i

R Program:

```r

solve.quadratic <- function(a,b,c){

d <- b^2-4*a*c

if (d >= 0){
  x1 <- (-b+sqrt(d))/(2*a)
  x2 <- (-b-sqrt(d))/(2*a)
}

if (d < 0){
  x1 <- (-b+sqrt(Mod(d))*1i)/(2*a)
  x2 <- (-b-sqrt(Mod(d))*1i)/(2*a)
}

print(x1)
print(x2)
}

solve.quadratic(1, 6, 2)
solve.quadratic(0, -4, 1.6)
solve.quadratic(3, 2.5, 7)
```

11:3 solve.quadratic(a, b, c)

Console Terminal Jobs

R 4.1.1 · ~/

```
+ }
>
> solve.quadratic(1, 6, 2)
[1] -0.3542487
[1] -5.645751
> solve.quadratic(0, -4, 1.6)
[1] Inf
[1] NaN
> solve.quadratic(3, 2.5, 7)
[1] -0.416667+1.469599i
[1] -0.416667-1.469599i
>
```

2.4 The sine function can be evaluated by the following infinite series:

$$\sin x = x - x^3/3! + x^5/5! - x^7/7! + \ldots$$

Write an algorithm to implement this formula so that it computes and prints out the values of sin x as each term in the series is added. In other words, compute and print in sequence the values for

$$\sin x = x$$
$$\sin x = x - x^3/3!$$
$$\sin x = x - x^3/3! + x^5/5!$$

up to the order term n of your c/hoosing. For each of the preceding, compute and display the percent relative error as

% error = (true - series approximation / true)* 100%

Sol: **Algorithm:**

1. Start.
2. Input x,n, i=1,j=2,y=0
3. $y = y + (-1)^j * (x^i/fact(i))$
4. Increment i 2 steps from 1 to n
5. j++
6. print y
7. Error = (sin(x)-y/sin(x))*100%
8. End

Excel Sheet:

Problem No: 2.4			
i	**x**	**j**	**y**
1	5	2	5
3	5	3	-15.83333
5	5	4	10.208333
7	5	5	-5.292659
9	5	6	0.0896302
11	5	7	-1.133617
13	5	8	-0.937584
15	5	9	-0.960921
17	5	10	-0.958776
19	5	11	-0.958933
21	5	12	-0.958924
23	5	13	-0.958924
25	5	14	-0.958924

R Program:

```r
x <- 5
j <- 2
y <- 0
for (i in seq(1,25,2)){
  y <- y + (-1)^j * (x^i / factorial(i))
  j <- j+1
}

print(y)
sin(5)

Error <- {sin(5) - (-0.9589243)}/sin(5)
print(Error)
```

13:13 (Top Level) R Script

Console Terminal Jobs

R 4.1.1 · ~/

```
> x <- 5
> j <- 2
> y <- 0
> for (i in seq(1,25,2)){
+   y <- y + (-1)^j * (x^i / factorial(i))
+   j <- j+1
+ }
>
> print(y)
[1] -0.9589243
> sin(5)
[1] -0.9589243
>
> Error <- {sin(5) - (-0.9589243)}/sin(5)
> print(Error)
[1] -2.642217e-08
>
```

2.8 An amount of money P is invested in an account where interest is compounded at the end of the period. The future worth F yielded at an interest rate i after n periods may be determined from the following formula:

$$F = P(1 + i)^n$$

Write a program that will calculate the future worth of an investment for each year from 1 through n. The input to the function should include the initial investment P, the interest rate i (as a decimal), and the number of years n for which the future worth is to be calculated. The output should consist of a table with headings and columns for n and F. Run the program for P = \$100,000, i = 0.04, and n = 11 years.

Sol: **Algorithm:**

1. Start
2. Input P = 100000, i = 0.04, n = 1 to 11, F
3. $F = P(1+I)^n$ (from n=1 till n=11)
4. Print F
5. End

Excel Sheet:

Problem No: 2.8

n	i	P	F
1	0.04	100000	104000
2	0.04	100000	108160
3	0.04	100000	112486.4
4	0.04	100000	116985.856
5	0.04	100000	121665.2902
6	0.04	100000	126531.9018
7	0.04	100000	131593.1779
8	0.04	100000	136856.905
9	0.04	100000	142331.1812
10	0.04	100000	148024.4285
11	0.04	100000	153945.4056

R Program:

```r
p <- 100000
i <- 0.04

for (n in seq(1,11)){
  F <-  p * (1+i)^n
  print(F)
}
```

```
> p <- 100000
> i <- 0.04
>
> for (n in seq(1,11)){
+   F <-  p * (1+i)^n
+   print(F)
+ }
[1] 104000
[1] 108160
[1] 112486.4
[1] 116985.9
[1] 121665.3
[1] 126531.9
[1] 131593.2
[1] 136856.9
[1] 142331.2
[1] 148024.4
[1] 153945.4
>
```

2.9 Economic formulas are available to compute annual payments for loans. Suppose that you borrow an amount of money P and agree to repay it in n annual payments at an interest rate of i. The formula to compute the annual payment A is

$$A = P * i(1 + i)^n / (1 + i)^n - 1$$

Write a program to compute A. Test it with P = $55,000 and an interest rate of 6.6% (i = 0.066). Compute results for n = 1, 2, 3, 4, and 5 and display the results as a table with headings and columns for n and A.

Sol: **Algorithm:**

1. Start
2. Input n=1 ,P = 55000, i=0.066, A,I
3. For n=1 to 5
4. I = P *I
5. A = P + I
6. P = A
7. Print A
8. End

Excel Sheet:

Problem No: 2.9

n	P	i	I	A
1	55000	0.066	3630	58630
2	58630	0.066	3869.58	62499.58
3	62499.58	0.066	4124.9723	66624.55228
4	66624.55228	0.066	4397.2205	71021.77273
5	71021.77273	0.066	4687.437	75709.20973

R Program:

```r
P <- 55000
i <- 0.066
A <- 0

for (n in seq(1,5)){
  I <- P * i
  A <- P + I
  print(A)
  P <- A
}
```

```
> P <- 55000
> i <- 0.066
> A <- 0
>
> for (n in seq(1,5)){
+   I <- P * i
+   A <- P + I
+   print(A)
+   P <- A
+ }
[1] 58630
[1] 62499.58
[1] 66624.55
[1] 71021.77
[1] 75709.21
>
```

2.18 Piecewise functions are sometimes useful when the relationship between a dependent and an independent variable cannot be adequately represented by a single equation. For example, the velocity of a rocket might be described by

$$v(t) = \begin{cases} 11t^2 - 5t & 0 \le t < 10 \\ 1100 - 5t & 10 \le t < 20 \\ 50t + 2(t - 20)^2 & 20 \le t < 30 \\ 1520e^{-0.2(t-30)} & t > 30 \\ 0 & \text{Otherwise} \end{cases}$$

Develop a well-structured function to compute v as a function of t. Then use this function to generate a table of v vs t for t = -5 to 50 at increments of 0.5.

Sol: **<u>Excel Sheet:</u>**

Problem No: 2.18

t	v
-5	0
-4.5	0
-4	0
-3.5	0
-3	0
-2.5	0
-2	0
-1.5	0
-1	0
-0.5	0
0	0
0.5	0.25
1	6
1.5	17.25
2	34
2.5	56.25
3	84
3.5	117.25
4	156
4.5	200.25
5	250
5.5	305.25
6	366
6.5	432.25
7	504
7.5	581.25

7	504
7.5	581.25
8	664
8.5	752.25
9	846
9.5	945.25
10	1050
10.5	1047.5
11	1045
11.5	1042.5
12	1040
12.5	1037.5
13	1035
13.5	1032.5
14	1030
14.5	1027.5
15	1025
15.5	1022.5
16	1020
16.5	1017.5
17	1015
17.5	1012.5
18	1010
18.5	1007.5
19	1005
19.5	1002.5
20	1000
20.5	1025.5
21	1052
21.5	1079.5

22	1108
22.5	1137.5
23	1168
23.5	1199.5
24	1232
24.5	1265.5
25	1300
25.5	1335.5
26	1372
26.5	1409.5
27	1448
27.5	1487.5
28	1528
28.5	1569.5
29	1612
29.5	1655.5
30	1700
30.5	1375.352875
31	1244.470745
31.5	1126.043695
32	1018.88647
32.5	921.9266028
33	834.1936869
33.5	754.8096618
34	682.9800255
34.5	617.9858828
35	559.1767506
35.5	505.9640472

35.5	505.9640472
36	457.8152021
36.5	414.2483254
37	374.8273852
37.5	339.1578434
38	306.8827074
38.5	277.6789566
39	251.2543101
39.5	227.3443012
40	205.7096305
40.5	186.1337709
41	168.4208007
41.5	152.3934425
42	137.891289
42.5	124.7691979
43	112.8958389
43.5	102.1523794
44	92.43129519
44.5	83.63529449
45	75.67634392
45.5	68.47478764
46	61.95855005
46.5	56.06241445
47	50.72737034
47.5	45.9000228
48	41.53205812
48.5	37.57976023
49	34.00357322
49.5	30.7677054
50	27.83977111

R Program:

```r
for (t in seq(0,10,0.5)){
  v <- 11 * t^2 - 5 * t
  print(v)
}

for (t in seq(10,20,0.5)){
  v <- 1100 - 5 * t
  print(v)
}

for (t in seq(20,30,0.5)){
  v <- 50 * t + 2 * (t-20)^2
  print(v)
}

for (t in seq(30,50,0.5)){
  v <- 1520 * exp(-0.2*(t-30))
  print(v)
}
```

Console

```
R 4.1.1 · ~/
> for (t in seq(0,10,0.5)){
+   v <- 11 * t^2 - 5 * t
+   print(v)
+ }
[1] 0
[1] 0.25
[1] 6
[1] 17.25
[1] 34
[1] 56.25
[1] 84
[1] 117.25
[1] 156
[1] 200.25
[1] 250
[1] 305.25
```

```
R 4.1.1 · ~/
[1] 250
[1] 305.25
[1] 366
[1] 432.25
[1] 504
[1] 581.25
[1] 664
[1] 752.25
[1] 846
[1] 945.25
[1] 1050
>
> for (t in seq(10,20,0.5)){
+   v <- 1100 - 5 * t
+   print(v)
+ }
[1] 1050
[1] 1047.5
[1] 1045
[1] 1042.5
[1] 1040
[1] 1037.5
[1] 1035
[1] 1032.5
[1] 1030
[1] 1027.5
[1] 1025
[1] 1022.5
[1] 1020
[1] 1017.5
[1] 1015
[1] 1012.5
[1] 1010
[1] 1007.5
```

```
R 4.1.1 · ~/
[1] 1010
[1] 1007.5
[1] 1005
[1] 1002.5
[1] 1000
>
> for (t in seq(20,30,0.5)){
+   v <- 50 * t + 2 * (t-20)^2
+   print(v)
+ }
[1] 1000
[1] 1025.5
[1] 1052
[1] 1079.5
[1] 1108
[1] 1137.5
[1] 1168
[1] 1199.5
[1] 1232
[1] 1265.5
[1] 1300
[1] 1335.5
[1] 1372
[1] 1409.5
[1] 1448
[1] 1487.5
[1] 1528
[1] 1569.5
[1] 1612
[1] 1655.5
[1] 1700
>
```

```
[1] 1700
>
> for (t in seq(30,50,0.5)){
+   v <- 1520 * exp(-0.2*(t-30))
+   print(v)
+ }
[1] 1520
[1] 1375.353
[1] 1244.471
[1] 1126.044
[1] 1018.886
[1] 921.9266
[1] 834.1937
[1] 754.8097
[1] 682.98
[1] 617.9859
[1] 559.1768
[1] 505.964
[1] 457.8152
[1] 414.2483
[1] 374.8274
[1] 339.1578
[1] 306.8827
[1] 277.679
[1] 251.2543
[1] 227.3443
[1] 205.7096
[1] 186.1338
[1] 168.4208
[1] 152.3934
[1] 137.8913
[1] 124.7692
[1] 112.8958
```

```
[1] 137.8913
[1] 124.7692
[1] 112.8958
[1] 102.1524
[1] 92.4313
[1] 83.63529
[1] 75.67634
[1] 68.47479
[1] 61.95855
[1] 56.06241
[1] 50.72737
[1] 45.90002
[1] 41.53206
[1] 37.57976
[1] 34.00357
[1] 30.76771
[1] 27.83977
```

2.26 The height of a small rocket y can be calculated as a function of time after blastoff with the following piecewise function:

$y = 38.1454t + 0.13743t^3$ $\quad 0 \leq t < 15$

$y = 1036 + 130.909(t - 15) + 6.18425(t - 15)^2$ $\quad 15 \leq t < 33$
$- 0.428(t\ 2\ 15)^3$

$y = 2900 - 62.468(t - 33)-16.9274(t - 33)^2$ $\quad t > 33$
$+ 0.41796(t -33)^3$

Develop a well-structured pseudo code function to compute y as a function of t. Note that if the user enters a negative value of t or if the rocket has hit the ground ($y \leq 0$) then return a value of zero for y. Also, the function should be invoked in the calling program as height(t). Write the algorithm as (a) pseudo code, or (b) in the high-level language of your choice.

Sol: **<u>Excel Sheet:</u>**

Problem No: 2.26

t	y
0	0
1	38.28283
2	77.39024
3	118.14681
4	161.37712
5	207.90575
6	258.55728
7	314.15629
8	375.52736
9	443.49507
10	518.884
11	602.51873
12	695.22384
13	797.82391
14	911.14352
15	1036
16	1172.66525
17	1319.131
18	1472.82925
19	1631.192
20	1791.65125
21	1951.639
22	2108.58725
23	2259.928
24	2403.09325
24	2403.09325
25	2535.515
26	2654.62525
27	2757.856
28	2842.63925
29	2906.407
30	2946.59125
31	2960.624
32	2945.93725
33	2899.963
34	2821.02256
35	2710.69808
36	2571.53432
37	2406.03904
38	2216.72
39	2006.08496
40	1776.64168
41	1530.89792
42	1271.36144
43	1000.54
44	720.94136
45	435.07328
46	145.44352
47	-145.44016
48	-435.07
49	-720.93824
50	-1000.53712

R Program:

```
for (t in seq(0,15)){
  y <- 38.1454 * t + 0.13743 * t^3
  print(y)
}

for (t in seq(15,33)){
  y <- 1036 + 130.909*(t - 15) + 6.18425*(t - 15)^2 - 0.428*(t - 15)^3
  print(y)
}

for (t in seq(34,50)){
  y <- 2900 - 62.468 * (t - 33) - 16.9274 * (t - 33)^2 + 0.41796 * (t - 33)^3
  print(y)
}
```

16:1 (Top Level) R Script

Console Terminal Jobs

R 4.1.1 · ~/

```
> for (t in seq(0,15)){
+   y <- 38.1454 * t + 0.13743 * t^3
+   print(y)
+ }
[1] 0
[1] 38.28283
[1] 77.39024
[1] 118.1468
[1] 161.3771
[1] 207.9058
[1] 258.5573
[1] 314.1563
[1] 375.5274
[1] 443.4951
[1] 518.884
[1] 602.5187
[1] 695.2238
[1] 797.8239
[1] 911.1435
[1] 1036.007
```

```
+ }
[1] 1036
[1] 1172.665
[1] 1319.131
[1] 1472.829
[1] 1631.192
[1] 1791.651
[1] 1951.639
[1] 2108.587
[1] 2259.928
[1] 2403.093
[1] 2535.515
[1] 2654.625
[1] 2757.856
[1] 2842.639
[1] 2906.407
[1] 2946.591
[1] 2960.624
[1] 2945.937
[1] 2899.963
```

```
+ }
[1] 2821.023
[1] 2710.698
[1] 2571.534
[1] 2406.039
[1] 2216.72
[1] 2006.085
[1] 1776.642
[1] 1530.898
[1] 1271.361
[1] 1000.54
[1] 720.9414
[1] 435.0733
[1] 145.4435
[1] -145.4402
[1] -435.07
[1] -720.9382
[1] -1000.537
```