



SIDDAGANGA INSTITUTE OF TECHNOLOGY  
www.sit.ac.in

*Foundations of Computer Programming Lab*

LAB MANUAL

**Prabodh C P**  
Asst Professor  
Dept of CSE, SIT



This work is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.

# Contents

<b>I PART A</b>	<b>5</b>
1 Sales of a shop	6
2 Sum of last two digits	9
3 Roots of a Quadratic Equation	12
4 Grade Calculation	16
5 Calculate Sine of an Angle	18
6 Print Calendar of a Month	20
7 Area under the curve $y=\log(x)$	23
8 Calculating $nCr$	25
9 $x$ raised to the power $n$	27
10 Pass by Reference	29
<b>II PART B</b>	<b>31</b>
1 Set Difference	32
2 Bubble Sort	35
3 Binary Search	38
4 Swap smallest and largest elements in an array using pointers	40
5 Matrix Operations	42
6 Matrix Multiplication	44
7 Caesar Cipher	48
8 Substring Search	50
9 Student Details	52
10 Bitwise Operations	54

# Listings

1.1	A01ShopSales.c	6
2.1	A02Sum2Digits.c	9
3.1	A03Quadratic.c	12
4.1	A04GradeCalculator.c	16
5.1	A05SineAngle.c	18
6.1	A06PrintCalendar.c	20
7.1	A07AreaLogX.c	23
8.1	A08RecNCR.c	25
9.1	A09fnRecPow.c	27
10.1	A10SurfAreaVolCuboid.c	29
1.1	B01ArrayDiff.c	32
2.1	B02BubbleSort.c	35
3.1	B03BinarySearch.c	38
4.1	B04LarSmallSwapPtr.c	40
5.1	B05Matrix.c	42
6.1	B06MatMul.c	44
7.1	B07Caesar.c	48
8.1	B08StrSearch.c	50
9.1	B09StudentMarks.c	52
10.1	B10BitWise.c	54

# FCP LABORATORY

## Common to all Branches

### Instructions

- All the C programs need to be executed on Codeblocks IDE using GCC Compiler.
- Algorithms and Flowcharts are compulsory for all the programs.
- Demo experiments are only for assignment purpose, not to be included in examinations.
- All experiments must be included in practical examinations.
- Students are allowed to pick one experiment each from Part A and Part B from the lots.

### References

**Part A:** Behrouz A. Forouzan , Richard F. Gilberg , Computer Science: **A Structured programming Approach Using C** - Cengage Learning; 3rd edition

For writing flowcharts refer to **Appendix C** of the above book.

Part I  
**PART A**

# Chapter 1

## Sales of a shop

### Question

*Write a program to calculate the total sales of a shop, given the unit price, quantity, discount rate and sales tax rate of a particular product. The sales tax is 8.5% which should be specified using a defined constant. The output should be displayed in proper form using appropriate format specifiers.*

### C Code

```
=====
1  /*****
2  *File           : A01ShopSales.c
3  *Descriptio    : program to automate shop sales
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date          : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10
11 #define SALES_TAX 0.085F
12
13 int main(void)
14 {
15     int iQuantity;
16     float fDiscRate, fDiscAmount, fUnitPrice, fSubTotal;
17     float fSubTaxable, fTaxAm, fTotal;
18
19
20
21     printf("\nEnter the number of items sold...\t:");
22     scanf("%d",&iQuantity);
23
24     printf("\nEnter the unit price... \t\t:");
25     scanf("%f",&fUnitPrice);
26
27     printf("\nEnter the discount rate (%)... \t\t:");
28     scanf("%f",&fDiscRate);
29
30     fSubTotal = iQuantity * fUnitPrice;
31     fDiscAmount = fSubTotal * fDiscRate / 100.0;
32     fSubTaxable = fSubTotal - fDiscAmount;
33     fTaxAm = fSubTaxable * SALES_TAX;
34     fTotal = fSubTaxable + fTaxAm;
```

```

35
36 printf("\nQuantity sold:    \t%3d\n", iQuantity);
37 printf("Unit Price of items: %9.2f\n", fUnitPrice);
38 printf("                      -----\n");
39
40 printf("SubTotal :           %9.2f\n", fSubTotal);
41 printf("Discount :           -%8.2f\n", fDiscAmount);
42 printf("Discounted total :    %9.2f\n", fSubTaxable);
43 printf("Sales Tax :           +%8.2f\n", fTaxAm);
44 printf("Total sales :          %9.2f\n", fTotal);
45
46
47 return 0;
48 }

```

Listing 1.1: A01ShopSales.c

## Output

```

1  /*
2  $ ./a.out
3
4  Enter the number of items sold...   :35
5
6  Enter the unit price...             :15.55
7
8  Enter the discount rate (%)...      :6
9
10 Quantity sold:           35
11 Unit Price of items:     15.55
12
13 SubTotal :               544.25
14 Discount :               - 32.65
15 Discounted total :      511.60
16 Sales Tax :              + 43.49
17 Total sales :            555.08
18
19 $ ./a.out
20
21 Enter the number of items sold...   :43
22
23 Enter the unit price...             :22.75
24
25 Enter the discount rate (%)...      :8
26
27 Quantity sold:           43
28 Unit Price of items:     22.75
29
30 SubTotal :               978.25
31 Discount :               - 78.26
32 Discounted total :      899.99
33 Sales Tax :              + 76.50
34 Total sales :            976.49
35
36 */

```

Listing 1.2: A01ShopSales.c

```
Enter the number of items sold...   :35
```

Enter the unit price... :15.55

Enter the discount rate (%)... :6

Quantity sold: 35  
Unit Price of items: 15.55

-----  
SubTotal : 544.25  
Discount : - 32.65  
Discounted total : 511.60  
Sales Tax : + 43.49  
Total sales : 555.08

Enter the number of items sold... :43

Enter the unit price... :22.75

Enter the discount rate (%)... :8

Quantity sold: 43  
Unit Price of items: 22.75

-----  
SubTotal : 978.25  
Discount : - 78.26  
Discounted total : 899.99  
Sales Tax : + 76.50  
Total sales : 976.49



## Chapter 2

# Sum of last two digits

### Question

*Write a program to extract and add the two least significant digits of an integer using the following user defined functions.*

*i Function that obtains the unit place digit of a given number*

*ii Function that obtains the tenth place digit of a given number*

*iii Function that adds unit and tenth place digits of a given number*

*The main function should call these functions.*

### C Code

```
=====
1  /*****
2  *File           : A02SumTwoDigits.c
3  *Descriptio    : Program to add the last two digits of a given number
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date          : 4 August 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 int fnExtractUnit(int);
11 int fnExtractTen(int);
12 int fnAddUnitTen(int, int);
13
14 int main(void)
15 {
16     int iNum, iTen, iUnit, iSum;
17
18     printf("\nEnter a number\n");
19     scanf("%d", &iNum);
20
21     iUnit = fnExtractUnit(iNum);
22     iTen = fnExtractTen(iNum);
23     iSum = fnAddUnitTen(iTen, iUnit);
24
25     printf("\nThe unit digit in %d is %d\n", iNum, iUnit);
26     printf("\nThe ten's digit in %d is %d\n", iNum, iTen);
27     printf("\nThe sum of last two digits in %d is %d\n", iNum, iSum);
28
29     return 0;

```

```
30 }
31
32 int fnExtractUnit(int iVal)
33 {
34     int iU;
35     iU = (iVal%10);
36     return iU;
37 }
38
39 int fnExtractTen(int iVal)
40 {
41     int iT;
42     iT = (iVal%100)/10;
43     return iT;
44 }
45
46 int fnAddUnitTen(int iVal1, int iVal2)
47 {
48     int iS;
49     iS = (iVal1 + iVal2);
50     return iS;
51 }
52 /*
53 $ ./a.out
54
55 Enter a number
56 4567
57
58 The unit digit in 4567 is 7
59
60 The ten's digit in 4567 is 6
61
62 The sum of last two digits in 4567 is 13
63 $ ./a.out
64
65 Enter a number
66 1209
67
68 The unit digit in 1209 is 9
69
70 The ten's digit in 1209 is 0
71
72 The sum of last two digits in 1209 is 9
73 $ ./a.out
74
75 Enter a number
76 76
77
78 The unit digit in 76 is 6
79
80 The ten's digit in 76 is 7
81
82 The sum of last two digits in 76 is 13
83
84 */
```

Listing 2.1: A02Sum2Digits.c

## Output

Enter a number  
4567

The unit digit in 4567 is 7

The ten's digit in 4567 is 6

The sum of last two digits in 4567 is 13

Enter a number  
1209

The unit digit in 1209 is 9

The ten's digit in 1209 is 0

The sum of last two digits in 1209 is 9

Enter a number  
76

The unit digit in 76 is 6

The ten's digit in 76 is 7

The sum of last two digits in 76 is 13

## Chapter 3

# Roots of a Quadratic Equation

### Question

**Write a program using switch statement, to find all the possible roots of Quadratic equation. Display suitable error messages for invalid inputs. Write a user defined function fnCalcDiscriminant to calculate the discriminant.**

### Summary

Any quadratic equation has two roots and the roots of the equation can be found using the formula

$$x = \frac{-b \pm \sqrt{\text{discriminant}}}{2 * a}$$

where  $\text{discriminant} = b^2 - 4 * a * c$  and a,b and c are the coefficients of the quadratic equation  $ax^2 + bx + c = 0$ . Hence as per the equation, if discriminant is equal to 0, then the value of both the roots are equal and real. Also to find x, when discriminant is not 0, we need to find square root of disc. When discriminant value is less than 0, the square root of the discriminant is imaginary and hence the roots of the equation are supposed to be imaginary and distinct. When discriminant value is greater than 0, the square root of the discriminant is real and hence the roots of the equation are supposed to be real and distinct.

### C Code

```
1  /*****
2  *File      : A04Quadratic.c
3  *Description  : Program to find the roots of a Quadratic Equation
4  *Author     : Prabodh C P
5  *Compiler   : gcc compiler, Ubuntu 14.04
6  *Date      : 3 August 2014
7  *****/
8
9  #include<stdio.h>
10 #include<stdlib.h>
11 #include<math.h>
12 float fnCalcDiscriminant(float, float, float);
13 /*****
14 *Function   : main
15 *Input parameters: no parameters
16 *RETURNS    : 0 on success
17 *****/
18
19 int main(void)
20 {
21     float fA, fB, fC, fDesc, fX1, fX2, fRealp, fImagp;
22     int iState;
23
```

```

24 printf("\n*****");
25 printf("\n*\tPROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION\t\t*\n");
26 printf("*****");
27
28
29 printf("\nEnter the coefficients of a,b,c \n");
30 scanf("%f%f%f", &fA, &fB, &fC);
31 if(0 == fA)
32 {
33     printf("\nInvalid input, not a quadratic equation - try again\n");
34     exit(0);
35 }
36
37 /*COMPUTE THE DESCRIMINANT*/
38 fDesc=fnCalcDiscriminant(fA, fB, fC);
39 ((0 == fDesc) ? (iState = 1) : ((fDesc > 0) ? (iState = 2) : (iState = 3)));
40 switch(iState)
41 {
42     case 1:
43         fX1 = fX2 = -fB/(2*fA);
44         printf("\nRoots are equal and the Roots are \n");
45         printf("\nRoot1 = %g and Root2 = %g\n", fX1, fX2);
46         break;
47     case 2:
48         fX1 = (-fB+sqrt(fDesc))/(2*fA);
49         fX2 = (-fB-sqrt(fDesc))/(2*fA);
50         printf("\nThe Roots are Real and distinct, they are \n");
51         printf("\nRoot1 = %g and Root2 = %g\n", fX1, fX2);
52         break;
53     case 3:
54         fRealp = -fB / (2*fA);
55         fImagp = sqrt(fabs(fDesc))/(2*fA);
56         printf("\nThe Roots are imaginary and they are\n");
57         printf("\nRoot1 = %g+i%g\n", fRealp, fImagp);
58         printf("\nRoot2 = %g-i%g\n", fRealp, fImagp);
59 }
60 return 0;
61 }
62
63 float fnCalcDiscriminant(float a, float b, float c)
64 {
65     return (b*b-4*a*c);
66 }
67
68 /*
69 *****
70 *          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
71 *****
72 Enter the coefficients of a,b,c
73 0 1 2
74
75 Invalid input, not a quadratic equation - try again
76 \end{Verbatim}
77 \textbf{\$. /a.out }
78 \begin{Verbatim}
79 *****
80 *          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
81 *****
82 Enter the coefficients of a,b,c
83 1 -5 6
84
85 The Roots are Real and distinct, they are

```

```

86
87 Root1 = 3 and Root2 = 2
88 \end{Verbatim}
89 \textbf{\$. /a.out }
90 \begin{Verbatim}
91 *****
92 *          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
93 *****
94 Enter the coefficients of a,b,c
95 1 4 4
96
97 Roots are equal and the Roots are
98
99 Root1 = -2 and Root2 = -2
100 \end{Verbatim}
101 \textbf{\$. /a.out }
102 \begin{Verbatim}
103 *****
104 *          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
105 *****
106 Enter the coefficients of a,b,c
107 1 3 3
108
109 The Roots are imaginary and they are
110
111 Root1 = -1.5+i0.866025
112 Root2 = -1.5-i0.866025
113
114 */

```

Listing 3.1: A03Quadratic.c

## Output

Run the following commands in your terminal:

```
$ gcc 01Quadratic.c -lm
$. /a.out
```

```

*****
*          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
*****
Enter the coefficients of a,b,c
0 1 2

```

Invalid input, not a quadratic equation - try again

```
$. /a.out
```

```

*****
*          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
*****
Enter the coefficients of a,b,c
1 -5 6

```

The Roots are Real and distinct, they are

```
Root1 = 3 and Root2 = 2
```

```
$. /a.out
```

```

*****
*          PROGRAM TO FIND ROOTS OF A QUADRATIC EQUATION          *
*****
Enter the coefficients of a,b,c

```

1 4 4

Roots are equal and the Roots are

Root1 = -2 and Root2 = -2

## Chapter 4

# Grade Calculation

### Question

Write a C program to read the marks of quiz1, quiz2, quiz3, quiz4, test1, test 2 and final exam of a student in one course. Maximum marks for each exam is given in the Table 1 below. Calculate the total of all the marks and print the appropriate grade to the student as given in the Table 2.

Table 1

Type of Exam	Total Marks	Reduced to
Quiz 1	15	3
Quiz 2	15	6
Quiz 3	15	
Quiz 4	15	3
Test 1	50	17
Test 2	50	17
Assignment		4
Internal CIE		50
Final Exam	100	50
Total Marks	Final Exam + Internal CIE	

Table 2

Total Marks	Grades
90 and above	S
75 to 89	A
61 to 74	B
50 to 60	C
40 to 49	D
less than 40	F

### C Code

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 int main()
6 {
7     int iQMark1, iQMark2, iQMark3, iQMark4, iTMark1, iTMark2, iAsMark, iExMark;
8     int iCIE, iSEE, iTotal;
9     char cGrade;
10
11     printf("\nEnter marks in Quiz1\n");
12     scanf("%d", &iQMark1);
13     printf("\nEnter marks in Quiz2\n");
14     scanf("%d", &iQMark2);
```



```

15     printf("\nEnter marks in Quiz1\n");
16     scanf("%d", &iQMark3);
17     printf("\nEnter marks in Quiz1\n");
18     scanf("%d", &iQMark4);
19     printf("\nEnter marks in Test1\n");
20     scanf("%d", &iTMark1);
21     printf("\nEnter marks in Test2\n");
22     scanf("%d", &iTMark2);
23     printf("\nEnter marks in Assignment\n");
24     scanf("%d", &iAsMark);
25     printf("\nEnter marks in Final Exam\n");
26     scanf("%d", &iExMark);
27
28     iCIE = ceil(iQMark1/5)+ceil(iTMark1*17/50)+ceil((iQMark2+iQMark3)/5)+ceil(
iTMark2*17/50)+ceil(iQMark4/5)+iAsMark;
29     iSEE = ceil(iExMark/2);
30
31     printf("\nInternal CIE obtained is %d")
32     iTTotal = iCIE + iSEE;
33
34     if(iTotal > 89)
35         cGrade = 'S';
36     else if(iTotal > 74)
37         cGrade = 'A';
38     else if(iTotal > 60)
39         cGrade = 'B';
40     else if(iTotal > 49)
41         cGrade = 'C';
42     else if(iTotal > 39)
43         cGrade = 'D';
44     else if(iTotal > 39)
45         cGrade = 'F';
46
47     printf("\nYou have obtained %c grade", cGrade);
48     return 0;
49 }

```

Listing 4.1: A04GradeCalculator.c

## Output

## Chapter 5

# Calculate Sine of an Angle

### Question

Write a program to find Sine of an angle using the series

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} \dots\dots\dots$$

For given N terms using the for loop structure. Also print sin(x) using Library Function.

### C Code

```
=====
1  /*****
2  *File      : A05bSineAngle.c
3  *Descriptio  : Program to calculate the sine of an angle
4  *Author    : Prabodh C P
5  *Compiler  : gcc compiler, Ubuntu 14.04
6  *Date     : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <math.h>
10
11 #define PI 3.1416
12
13 int main(void)
14 {
15     float fAngD, fAngR;
16     float fTerm, fNum, fDen, fVal;
17     int i, iNum;
18
19     printf("Enter the Angle ....\n");
20     scanf("%f", &fAngD);
21
22     printf("Angle = %g\n", fAngD);
23
24     printf("Enter the Number of terms...\n");
25     scanf("%d", &iNum);
26     printf("No of terms = %d\n", iNum);
27
28     fAngR= (fAngD*PI)/180 ;
29
30     fNum=fAngR;
31     fDen=1.0;
32     fVal =0.0;
33     fTerm=fNum/fDen;
```

```
34     for(i=1;i<=iNum;i++)
35     {
36         fVal = fVal + fTerm;
37         fNum = -fNum * fAngR * fAngR ;
38         fDen = fDen * (2*i) * (2*i+1);
39         fTerm = fNum/fDen;
40     }
41     printf(" Calculated value is :sin(%g) = %g\n",fAngD,fVal);
42     printf("Built In function value is :sin(%g) = %g\n",fAngD, sin(fAngR));
43     return 0;
44 }
```

Listing 5.1: A05SineAngle.c

## Output

```
Enter the Angle ....
60
Angle = 60
Enter the Number of terms...
3
No of terms = 3
  Calculated value is :sin(60) = 0.866297
  Built In function value is :sin(60) = 0.866027

Enter the Angle ....
60
Angle = 60
Enter the Number of terms...
8
No of terms = 8
  Calculated value is :sin(60) = 0.866027
  Built In function value is :sin(60) = 0.866027

Enter the Angle ....
90
Angle = 90
Enter the Number of terms...
8
No of terms = 8
  Calculated value is :sin(90) = 1
  Built In function value is :sin(90) = 1
```

## Chapter 6

# Print Calendar of a Month

### Question

*Write a program to display the calendar of a month whose starting day of the week and number of days in the month are given as input. (0,1,2,3...6 represent Sunday, Monday.....Saturday respectively). Write a function printMonth which takes these values as parameters and prints the calendar of the month. Perform input validation as well.*

### C Code

```
=====
1  /*****
2  *File           : A06PrintCalendar.c
3  *Description    : Program to print a calendar of the month
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date         : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 void printMonth(int, int);
11
12 int main(void)
13 {
14     int iNumOfDays, iStartDay;
15     printf("\nEnter the day of the week on which the month starts\n");
16     scanf("%d", &iStartDay);
17     printf("\nEnter the number of days in the month\n");
18     scanf("%d", &iNumOfDays);
19
20     if(iStartDay < 0 || iStartDay > 6 || iNumOfDays >31 || iNumOfDays <28)
21     {
22         printf("\nInvalid Input\n");
23         exit(0);
24     }
25     printMonth(iStartDay, iNumOfDays);
26     return 0;
27 }
28 void printMonth(int iStDay, int iNoDays)
29 {
30     int i;
31     printf("\nSun\tMon\tTue\tWed\tThu\tFri\tSat\n");
32     i=1;
33     while(i<=iStDay)
34     {
35         printf("\t");
```

```

36     i++;
37 }
38 i=1;
39 while(i<=iNoDays)
40 {
41     printf("%3d\t",i);
42     if(0 == (i+iStDay)%7)
43         printf("\n");
44     i++;
45 }
46 printf("\n");
47 }
48
49 /*
50 $ ./a.out
51
52 Enter the day of the week on which the month starts
53 3
54
55 Enter the number of days in the month
56 31
57
58 Sun Mon Tue Wed Thu Fri Sat
59     1  2  3  4
60  5  6  7  8  9 10 11
61 12 13 14 15 16 17 18
62 19 20 21 22 23 24 25
63 26 27 28 29 30 31
64
65 $ ./a.out
66
67 Enter the day of the week on which the month starts
68 3
69
70 Enter the number of days in the month
71 31
72
73 Sun Mon Tue Wed Thu Fri Sat
74     1  2  3  4
75  5  6  7  8  9 10 11
76 12 13 14 15 16 17 18
77 19 20 21 22 23 24 25
78 26 27 28 29 30 31
79
80 $ ./a.out
81
82 Enter the day of the week on which the month starts
83 6
84
85 Enter the number of days in the month
86 30
87
88 Sun Mon Tue Wed Thu Fri Sat
89     1
90  2  3  4  5  6  7  8
91  9 10 11 12 13 14 15
92 16 17 18 19 20 21 22
93 23 24 25 26 27 28 29
94 30
95
96 $ ./a.out
97

```

```
98 Enter the day of the week on which the month starts
99 7
100
101 Enter the number of days in the month
102 28
103
104 Invalid Input
105 */
```

Listing 6.1: A06PrintCalendar.c

## Output

# Chapter 7

## Area under the curve $y=\log(x)$

### Question

*Write a Program to calculate the area under the curve  $y = \log(x)$  within a given left and right limit. Use trapezoidal approximation. Write an user defined function `fnTrapArea` to calculate the area of a trapezoid.*

### C Code

```
1  /*****
2  *File      : A07AreaLogX.c
3  *Description : Program to calculate the area under the log(x) curve
4  *Author    : Prabodh C P
5  *Compiler  : gcc compiler, Ubuntu 14.04
6  *Date     : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <math.h>
11
12 double fnAreaTrapezoid(double, double, double);
13
14 int main(void)
15 {
16     double i, dDx=0.1, dX1, dX2, dY1, dY2;
17     double dLeft, dRight, dArea=0.0, dTotArea=0.0;
18
19     printf("\nEnter the left and right limit\n");
20     scanf("%lf%lf", &dLeft, &dRight);
21
22     printf("\nEnter the increment dx\n");
23     scanf("%lf", &dDx);
24
25     for(i=dLeft; i<dRight; i+=dDx)
26     {
27         dX1 = i;
28         dX2 = i+dDx;
29         dY1 = log(dX1);
30         dY2 = log(dX2);
31
32         dArea = fnAreaTrapezoid(dY1, dY2, dDx);
33         dTotArea += dArea;
34     }
35
36     printf("\nArea under the curve is %lf units\n", dTotArea);
37     return 0;
38 }
```

```
39
40 double fnAreaTrapezoid(double dHeight1, double dHeight2, double dBase)
41 {
42     double dArea;
43     dArea = dBase * ((dHeight1 + dHeight2)/2);
44     return dArea;
45 }
46
47 /*
48 $ ./a.out
49
50 Enter the left and right limit
51 2 5
52
53 Enter the increment dx
54 0.001
55
56 Area under the curve is 3.660895 units
57
58 $ ./a.out
59
60 Enter the left and right limit
61 1 2
62
63 Enter the increment dx
64 0.001
65
66 Area under the curve is 0.386988 units
67
68 $ ./a.out
69
70 Enter the left and right limit
71 6 9
72
73 Enter the increment dx
74 0.01
75
76 Area under the curve is 6.046442 units
77 */
```

Listing 7.1: A07AreaLogX.c

## Output



## Chapter 8

# Calculating nCr

### Question

**Write a recursive C function to find the factorial of a number,  $n!$ , defined by  $fact(n)=1$ , if  $n=0$ . Otherwise  $fact(n)=n*fact(n-1)$ . Using this function, Write a program to compute the binomial coefficient  $nCr$ . Perform input validation as well.**

### C Code

```
1  /*****
2  *File           : A08RecNCR.c
3  *Description    : Program to calculate NCR recursively
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date         : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10
11 int fnFactorial(int);
12
13 int main(void)
14 {
15     int iN, iR, iNCR;
16     printf("\nEnter the value of n\n");
17     scanf("%d", &iN);
18     printf("\nEnter the value of r\n");
19     scanf("%d", &iR);
20
21     if(iN < iR)
22     {
23         printf("\nnCr does not exist\n");
24         exit(0);
25     }
26
27     iNCR = (fnFactorial(iN)/(fnFactorial(iR)*fnFactorial(iN-iR)));
28
29     printf("\nFor n = %d and r = %d, %dC%d = %d\n", iN ,iR, iN, iR, iNCR);
30     return 0;
31 }
32
33 int fnFactorial(int iVal)
34 {
35     if(0 == iVal)
36         return 1;
37     else
38         return (iVal * fnFactorial(iVal - 1));
```

```
39 }
40
41 /*
42 $ ./a.out
43
44 Enter the value of n
45 3 6
46
47 Enter the value of r
48
49 nCr does not exist
50
51 $ ./a.out
52
53 Enter the value of n
54 5 2
55
56 Enter the value of r
57
58 For n = 5 and r = 2, 5C2 = 10
59
60 $ ./a.out
61
62 Enter the value of n
63 7 5
64
65 Enter the value of r
66
67 For n = 7 and r = 5, 7C5 = 21
68 */
```

Listing 8.1: A08RecNCR.c

## Output

# Chapter 9

## x raised to the power n

### Question

*Write a Program in C to find the value of 'x' raised to the power 'n' using recursion.*

### C Code

```
1  /*****
2  *File           : A09bfnRecPow.c
3  *Description    : Program to calculate x raised to n recursively
4  *Author         : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date          : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <math.h>
11
12 int fnRecPow(int, int);
13 int main(void)
14 {
15     int iX, iN;
16     float fRes;
17     printf("\n Enter value of x and n\n");
18     scanf("%d%d", &iX, &iN);
19
20     if(iN<0)
21     {
22         fRes = 1.0/fnRecPow(iX,abs(iN));
23     }
24     else
25     {
26         fRes = fnRecPow(iX,iN);
27     }
28     printf("\n%d raised to the power %d is %g\n",iX, iN, fRes);
29     return 0;
30 }
31
32 int fnRecPow(int iX, int iN)
33 {
34     if(0==iN)
35         return 1;
36     else if(1 == iN)
37         return iX;
38     else
39         return (iX * fnRecPow(iX, iN-1));
```

40 }

Listing 9.1: A09fnRecPow.c

## Output

# Chapter 10

## Pass by Reference

### Question

*Write a program that reads the length, breadth and height of a cuboid. The program has to calculate the surface area and volume of the cuboid using the function `fnCalcVolSurfArea`. The results are then to be displayed by the main function. (Hint: use pass by reference)*

### C Code

```
1  /*****
2  *File           : B05SurfAreaVolCuboid.c
3  *Description    : Program to illustrate pass by reference
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date         : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10
11 void fnCalcVolSurfArea(int, int, int, int*, int*);
12 int main(void)
13 {
14     int iLength, iBreadth, iHeight, iSurfArea, iVolume;
15
16     printf("\nEnter the length, breadth and height of the cuboid\n");
17     scanf("%d%d%d", &iLength, &iBreadth, &iHeight);
18
19     fnCalcVolSurfArea(iLength, iBreadth, iHeight, &iSurfArea, &iVolume);
20
21     printf("\nSurface Area of cuboid is %d units\n", iSurfArea);
22     printf("\nVolume of cuboid is %d cubic units\n", iVolume);
23     return 0;
24 }
25
26 void fnCalcVolSurfArea(int iL, int iB, int iH, int *iA, int *iV)
27 {
28     *iA = 2*iL*iB + 2*iB*iH + 2*iH*iL;
29     *iV = iL*iB*iH;
30 }
31
32 /*
33 $ ./a.out
34
35 Enter the length, breadth and height of the cuboid
36 3 4 5
37
38 Surface Area of cuboid is 94 units
```

```
39
40 Volume of cuboid is 60 cubic units
41 $ ./a.out
42
43 Enter the length, breadth and height of the cuboid
44 6 6 6
45
46 Surface Area of cuboid is 216 units
47
48 Volume of cuboid is 216 cubic units
49 */
```

Listing 10.1: A10SurfAreaVolCuboid.c

## Output

**Part II**  
**PART B**

# Chapter 1

## Set Difference

### Question

**Write a program to find the Difference of two sets(sets have distinct elements).**

$$A - B = \{X | X \in A \text{ and } X \ni B\}$$

### C Code

```
1  /*****
2  *File           : B01aArrayDiff.c
3  *Description    : Program to find the difference of two sets
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date         : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <stdbool.h>
11 #define MAX 10
12 int main(void)
13 {
14     int iaA[MAX], iaB[MAX], iaC[MAX], iNum1, iNum2;
15     int i, j, iCount = 0;
16     bool bNotOccur;
17
18     printf("\nEnter no of elements in Set A\n");
19     scanf("%d", &iNum1);
20     printf("\nEnter no of elements in Set B\n");
21     scanf("%d", &iNum2);
22
23     printf("\nEnter %d elements in Set A\n", iNum1);
24     for(i=0; i<iNum1; ++i)
25     {
26         scanf("%d", &iaA[i]);
27     }
28     printf("\nEnter %d elements in Set B\n", iNum2);
29     for(i=0; i<iNum2; ++i)
30     {
31         scanf("%d", &iaB[i]);
32     }
33
34     /*Copy those elements of X that are not in Y into C*/
35     for(i=0; i<iNum1; i++)
36     {
37         bNotOccur = true;
38         for(j=0; j<iNum2; j++)
```



```

39     {
40         if(iaA[i] == iaB[j])
41             bNotOccur = false;
42     }
43     if(bNotOccur == true)
44     {
45         iaC[iCount++] = iaA[i];
46     }
47 }
48 printf("\nElements of Set A:\n");
49 for(i=0;i<iNum1;++i)
50 {
51     printf("%d\t",iaA[i]);
52 }
53
54 printf("\nElements of Set B:\n");
55 for(i=0;i<iNum2;++i)
56 {
57     printf("%d\t",iaB[i]);
58 }
59
60 printf("\nDifference of A and B: \n");
61 for(i=0;i<iCount;++i)
62 {
63     printf("%d\t",iaC[i]);
64 }
65 printf("\n");
66 return 0;
67 }
68
69 /*
70
71 $ ./a.out
72
73 Enter no of elements in Set A
74 4
75
76 Enter no of elements in Set B
77 5
78
79 Enter 4 elements in Set A
80 1 2 3 4
81
82 Enter 5 elements in Set B
83 3 4 5 6 7
84
85 Elements of Set A:
86 1 2 3 4
87 Elements of Set B:
88 3 4 5 6 7
89 Difference of A and B:
90 1 2
91
92 $ ./a.out
93
94 Enter no of elements in Set A
95 5
96
97 Enter no of elements in Set B
98 4
99
100 Enter 5 elements in Set A

```

```
101 3 4 5 6 7
102
103 Enter 4 elements in Set B
104 1 2 3 4
105
106 Elements of Set A:
107 3 4 5 6 7
108 Elements of Set B:
109 1 2 3 4
110 Difference of A and B:
111 5 6 7
112
113 $ ./a.out
114
115 Enter no of elements in Set A
116 5
117
118 Enter no of elements in Set B
119 4
120
121 Enter 5 elements in Set A
122 3 5 7 4 6
123
124 Enter 4 elements in Set B
125 4 1 2 3
126
127 Elements of Set A:
128 3 5 7 4 6
129 Elements of Set B:
130 4 1 2 3
131 Difference of A and B:
132 5 7 6
133
134 */
```

Listing 1.1: B01ArrayDiff.c

## Output

## Chapter 2

# Bubble Sort

### Question

Write a program that reads  $N$  integer numbers and arrange them in ascending order using Bubble Sort technique. Write the following user defined functions.

- i* function to generate  $n$  random numbers in the range 0 to 999 and store it in an array.
- ii* function to sort the array using Selection sort.
- iii* function to sort the array using Insertion sort.
- iv* function to display the elements of an array.

### C Code

```
1  /*****
2  *File      : B02BubbleSort.c
3  *Description : Program to sort an array using Insertion Sort
4  *Author    : Prabodh C P
5  *Compiler  : gcc compiler 4.6.3, Ubuntu 12.04
6  *Date     : 29 Sep 2017
7  *****/
8
9  #include <stdio.h>
10 #include <stdlib.h>
11 #include <time.h>
12
13 void fnGenRandInput(int [], int);
14 void fnDispArray( int [], int);
15 void fnBubbleSort(int [], int);
16
17
18 /*****
19 *Function  : main
20 *Input parameters: no parameters
21 *RETURNS   : 0 on success
22 *****/
23 int main(void)
24 {
25
26     int iaArr[100], iNum;
27
28     printf("\nBubble Sort\n");
29     printf("\nEnter the number of elements to sort\n");
30     scanf("%d", &iNum);
31     fnGenRandInput(iaArr, iNum);
32     printf("\nUnsorted Array\n");
```

```

33     fnDispArray(iaArr, iNum);
34     fnBubbleSort(iaArr, iNum);
35     printf("\nSorted Array\n");
36     fnDispArray(iaArr, iNum);
37
38     return 0;
39 }
40
41 /*****
42 *Function      : fnBubbleSort
43 *Description   : Function to sort elements in an iaArray using Bubble Sort
44 *Input parameters:
45 *   int A[] - iaArray to hold integers
46 *   int n   - no of elements in the array
47 *RETURNS      : no value
48 *****/
49
50 void fnBubbleSort(int A[], int n)
51 {
52     int i, j, iTemp;
53     for(i=0; i<n-1; i++)
54     {
55         for(j=0; j<n-1-i; j++)
56         {
57             if(A[j]>A[j+1])
58             {
59                 iTemp = A[j];
60                 A[j] = A[j+1];
61                 A[j+1] = iTemp;
62             }
63         }
64     }
65 }
66
67
68 /*****
69 *Function      : GenRandInput
70 *Description   : Function to generate a fixed number of random elements
71 *Input parameters:
72 *   int X[] - array to hold integers
73 *   int n   - no of elements in the array
74 *RETURNS      :no return value
75 *****/
76
77 void fnGenRandInput(int X[], int n)
78 {
79     int i;
80     srand(time(NULL));
81     for(i=0; i<n; i++)
82     {
83         X[i] = rand()%1000;
84     }
85 }
86
87 /*****
88 *Function      : DispArray
89 *Description   : Function to display elements of an array
90 *Input parameters:
91 *   int X[] - array to hold integers
92 *   int n   - no of elements in the array
93 *RETURNS      : no return value
94 *****/

```

```
95
96 void fnDispArray( int X[], int n)
97 {
98     int i;
99     for(i=0;i<n;i++)
100         printf(" %4d \n",X[i]);
101 }
```

Listing 2.1: B02BubbleSort.c

## Output

## Chapter 3

# Binary Search

### Question

**Write a program that reads  $N$  integer numbers in ascending order and perform a search operation on these sorted numbers by accepting a key element from the user applying Binary Search method. Report the result SUCCESS or FAILURE as the case may be. Write user defined functions for the following**

- i reads  $N$  integer numbers***
- ii display  $N$  integers***
- iii search for a key element using Binary Search***

### C Code

```
1  /*****
2  *File           : B03BinarySearch.c
3  *Description    : Program to sort and search an array using Bubble sort
4  *              : and Binary search
5  *Author         : Prabodh C P
6  *Compiler       : gcc compiler, Ubuntu 14.04
7  *Date          : 18 Novemeber 2014
8  *****/
9  #include <stdio.h>
10 #include <stdlib.h>
11
12 void fnReadArray(int [], int);
13 void fnDispArray(int [], int);
14 int fnSearchArray(int [], int, int);
15
16 int main(void)
17 {
18     int iaList[100], iKey, iNum, iPos;
19
20     printf("\nEnter the no. of elements\n");
21     scanf("%d", &iNum);
22
23     printf("\nEnter %d elements in ascending order\n", iNum);
24     fnReadArray(iaList, iNum);
25     printf("\nEntered values are :\n");
26     fnDispArray(iaList, iNum);
27
28     printf("\nEnter the key element\n");
29     scanf("%d", &iKey);
30     iPos = fnSearchArray(iaList, iKey, iNum);
31
```

```
32     if(-1 == iPos)
33     {
34         printf("\nElement not found\n");
35     }
36     else
37     {
38         printf("\nElement %d found at position %d\n", iKey, iPos+1);
39     }
40     return 0;
41 }
42
43 void fnReadArray(int iaArr[], int n)
44 {
45     int i;
46     for(i=0;i<n;i++)
47         scanf("%d", &iaArr[i]);
48 }
49
50 void fnDispArray(int iaArr[], int n)
51 {
52     int i;
53     for(i=0;i<n;i++)
54         printf("\t%d", iaArr[i]);
55     printf("\n");
56 }
57
58
59 int fnSearchArray(int iaArr[], int iKey, int iNum)
60 {
61     int iProbe, iLow, iHigh, iMid;
62     iLow = 0;
63     iHigh = iNum -1;
64     iProbe = 0;
65     while(iLow <= iHigh)
66     {
67         iProbe++;
68         iMid = (iLow + iHigh)/2;
69         if(iKey == iaArr[iMid])
70         {
71             printf("\nTotal number of probes required is %d\n", iProbe);
72             return iMid;
73         }
74         else if(iKey < iaArr[iMid])
75             iHigh = iMid-1;
76         else
77             iLow = iMid +1;
78     }
79     printf("\nTotal number of probes required is %d\n", iProbe);
80     return -1;
81 }
```

Listing 3.1: B03BinarySearch.c

## Output

## Chapter 4

# Swap smallest and largest elements in an array using pointers

### Question

*Write a program to find the smallest and largest elements in an array using pointers and then swap these elements and display the resultant array.*

### C Code

```
1  /*****
2  *File      : B04LarSmallSwapPtr.c
3  *Description : Program to swap smallest and largest elements in an array
4  *          : using pointers
5  *Author    : Prabodh C P
6  *Compiler  : gcc compiler, Ubuntu 14.04
7  *Date     : 18 Novemeber 2014
8  *****/
9  #include <stdio.h>
10 #include <stdlib.h>
11
12 int main(void)
13 {
14     int iaList[50], iNum, i, j, iTemp, iPos;
15     int *iPtr1,*iPtr2;
16
17     iPtr1 = iPtr2 = iaList;
18     printf("\nEnter the value of n : ");
19     scanf("%d", &iNum);
20
21     printf("\nEnter the elements: ");
22     for(i=0;i<iNum;++i)
23     {
24         scanf("%d", &iaList[i]);
25     }
26
27     for(i=1;i<iNum;++i)
28     {
29         if(iaList[i] < *iPtr1)
30         {
31             iPtr1 = &iaList[i];
32         }
33         if(iaList[i] > *iPtr2)
34         {
35             iPtr2 = &iaList[i];
```



```
36     }
37 }
38
39 printf("\nArray elements before swapping the largest and smallest elements\n");
40 ;
41 for(i=0;i<iNum;i++)
42 {
43     printf("%d\t", iaList[i]);
44 }
45
46 printf("\nSmallest Element = %d\n",*iPtr1);
47 printf("\nLargest Element = %d\n",*iPtr2);
48
49 iTemp = *iPtr1;
50 *iPtr1 = *iPtr2;
51 *iPtr2 = iTemp;
52
53 printf("\nArray elements after swapping the largest and smallest elements\n");
54 for(i=0;i<iNum;i++)
55 {
56     printf("%d\t", iaList[i]);
57 }
58 printf("\n");
59 return 0;
60 }
```

Listing 4.1: B04LarSmallSwapPtr.c

## Output

## Chapter 5

# Matrix Operations

### Question

**Write a program that reads a matrix of order  $M \times N$ . Write user defined functions for the following.**

- i Reads  $M \times N$  Matrix.**
  - ii Display the Matrix.**
  - iii To check whether given matrix is Identity or Not Identity Matrix.**
- The main function should call these functions.**

### C Code

```
1  /*****
2  *File      : B05Matrix.c
3  *Description : Program to perform matrix operations
4  *Author    : Prabodh C P
5  *Compiler  : gcc compiler, Ubuntu 14.04
6  *Date     : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <stdbool.h>
11
12 void fnReadMatrix(int[][10], int, int);
13 void fnShowMatrix(int[][10], int, int);
14 bool fnCheckIdentity(int[][10], int, int);
15
16 int main(void)
17 {
18     int iaMat[10][10], iRow, iCol, i, j;
19
20     printf("\nEnter the order of the matrix\n");
21     scanf("%d%d", &iRow, &iCol);
22
23     printf("\nEnter matrix elements\n");
24     fnReadMatrix(iaMat, iRow, iCol);
25
26     printf("\nEntered Matrix is\n");
27
28     fnShowMatrix(iaMat, iRow, iCol);
29
30     if(fnCheckIdentity(iaMat, iRow, iCol))
31     {
```

```

32     printf("\nMatrix is a Identity Matrix\n");
33 }
34 else
35 {
36     printf("\nMatrix is not a Identity Matrix\n");
37 }
38
39 return 0;
40 }
41
42 void fnReadMatrix(int iaMatrix[][10], int iM, int iN)
43 {
44     int i, j;
45     for(i=0; i<iM; i++)
46         for(j=0; j<iN; j++)
47             scanf("%d", &iaMatrix[i][j]);
48
49 }
50
51 void fnShowMatrix(int iaMatrix[][10], int iM, int iN)
52 {
53     int i, j;
54     for(i=0; i<iM; i++)
55     {
56         for(j=0; j<iN; j++)
57         {
58             printf("%8d", iaMatrix[i][j]);
59         }
60         printf("\n");
61     }
62 }
63
64 bool fnCheckIdentity(int iaMatrix[][10], int iM, int iN)
65 {
66     int i, j;
67     bool isIdentity = true;
68     if(iM != iN)
69         isIdentity = false;
70     for(i=0; i<iM; i++)
71     {
72         for(j=0; j<iN; j++)
73         {
74             if(i != j && iaMatrix[i][j] != 0)
75                 isIdentity = false;
76             if(i == j && iaMatrix[i][j] != 1)
77                 isIdentity = false;
78         }
79     }
80     return isIdentity;
81 }

```

Listing 5.1: B05Matrix.c

## Output

## Chapter 6

# Matrix Multiplication

### Question

**Write a program that reads two matrices  $A$  ( $m \times n$ ) and  $B$  ( $p \times q$ ) and Compute the product  $A \times B$ . Read matrix  $A$  in row major order and matrix  $B$  in column major order. Print both the input matrices and resultant matrix with suitable headings and in matrix format. Display the trace of the matrix if it exists. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.**

### C Code

```
1  /*****
2  *File      : MatrixMul.c
3  *Description : Program to implement Matrix Multiplication
4  *Author    : Prabodh C P
5  *Compiler  : gcc 4.6.3 compiler, Ubuntu 10.04
6  *Date      : Wednesday, July 16 2014
7  *****/
8
9  #include<stdio.h>
10 #include<stdlib.h>
11
12 /*****
13 *Function   : main
14 *Input parameters: no parameters
15 *RETURNS   : 0 on success
16 *****/
17
18 int main(void)
19 {
20     int iM, iN, iP, iQ, i, j, k, iaMat1[10][10], iaMat2[10][10], iTrace = 0;
21     int iaProd[10][10] = {0}; //Initialize the matrix with all zeros
22
23
24     printf("\n*****");
25     printf("\n*\tPROGRAM TO IMPLEMENT MATRIX MULIPLICATION\t*\n");
26     printf("*****");
27
28     printf("\nEnter the order of Matrix1\n");
29     scanf("%d%d", &iM, &iN);
30
31     printf("\nEnter the order of Matrix2\n");
32     scanf("%d%d", &iP, &iQ);
33
34     if( iN != iP)
35     {
```

```
36     printf("\nMatrix Multiplication not possible\n");
37     exit(0);
38 }
39
40
41 printf("\nEnter the elements of Matrix 1 in row major order\n");
42 for(i=0; i<iM; i++)
43     for(j=0; j<iN; j++)
44         scanf("%d", &iaMat1[i][j]);
45
46 printf("\nEnter the elements of Matrix 2 in column major order\n");
47 for(i=0; i<iQ; i++)
48     for(j=0; j<iP; j++)
49         scanf("%d", &iaMat2[j][i]);
50
51
52 for(i=0; i<iM; i++)
53 {
54     for(j=0; j<iQ; j++)
55     {
56         for(k=0; k<iN; k++)
57         {
58             iaProd[i][j] += iaMat1[i][k] * iaMat2[k][j];
59         }
60     }
61 }
62
63 printf("\nMatrix 1\n");
64
65 for(i=0; i<iM; i++)
66 {
67     for(j=0; j<iN; j++)
68     {
69         printf("%d\t", iaMat1[i][j]);
70     }
71     printf("\n");
72 }
73 printf("\n");
74
75 printf("\nMatrix 2\n");
76
77 for(i=0; i<iP; i++)
78 {
79     for(j=0; j<iQ; j++)
80     {
81         printf("%d\t", iaMat2[i][j]);
82     }
83     printf("\n");
84 }
85 printf("\n");
86
87 printf("\nThe Product matrix is \n");
88
89 for(i=0; i<iM; i++)
90 {
91     for(j=0; j<iQ; j++)
92     {
93         printf("%d\t", iaProd[i][j]);
94     }
95     printf("\n");
96 }
97 printf("\n");
```

```

98
99     if(iM == iQ)
100    {
101        for(i=0;i<iM;i++)
102            iTrace += iaProd[i][i];
103        printf("\nTrace of the product matrix is : %d\n", iTrace);
104    }
105    else
106    {
107        printf("\nTrace does not exist\n");
108    }
109    return 0;
110 }
111 /*
112 $ ./a.out
113
114 *****
115 *   PROGRAM TO IMPLEMENT MATRIX MULIPLICATION   *
116 *****
117 Enter the order of Matrix1
118 2 3
119
120 Enter the order of Matrix2
121 3 2
122
123 Enter the elements of Matrix 1 in row major order
124 1 2 3 4 5 6
125
126 Enter the elements of Matrix 2 in column major order
127 1 2 3 4 5 6
128
129 Matrix 1
130 1  2  3
131 4  5  6
132
133
134 Matrix 2
135 1  4
136 2  5
137 3  6
138
139
140 The Product matrix is is
141 14 32
142 32 77
143
144 $ ./a.out
145
146 *****
147 *   PROGRAM TO IMPLEMENT MATRIX MULIPLICATION   *
148 *****
149 Enter the order of Matrix1
150 3 4
151
152 Enter the order of Matrix2
153 3 4
154
155 Matrix Multiplication not possible
156
157 */

```

Listing 6.1: B06MatMul.c

## Output

## Chapter 7

# Caesar Cipher

### Question

*Write a Program to implement Caesar Cipher. Input a message and then encodes it by replacing each character in the message by a character that is three positions ahead in the English alphabet sequence, wrap back to 'a' if the character is 'z'. Display the encoded message. Decode the message using the inverse procedure and display it.*

### C Code

```
1  /*****
2  *File           : B06Caesar.c
3  *Description    : program to implement Caesar Cipher
4  *Author         : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date          : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <ctype.h>
11 void fnEncrypt(char []);
12 void fnDecrypt(char []);
13
14 int main(void)
15 {
16     char acMesg[50];
17
18     printf("\nEnter your plain text in lower case\n");
19     gets(acMesg);
20
21     fnEncrypt(acMesg);
22     printf("Encrypted Ciphertext\n");
23     puts(acMesg);
24
25     fnDecrypt(acMesg);
26     printf("Decrypted Plaintext\n");
27     puts(acMesg);
28
29     return 0;
30 }
31
32 void fnEncrypt(char acStr[50])
33 {
34     int i;
35     for(i=0;acStr[i]!='\0';i++)
36     {
```



```

37     if(!isspace(acStr[i]) && islower(acStr[i]))
38     {
39         acStr[i] = (acStr[i]-'a'+3+26)%26+'a';
40     }
41     else if(!isspace(acStr[i]) && isupper(acStr[i]))
42     {
43         acStr[i] = (acStr[i]-'A'+3+26)%26+'A';
44     }
45
46
47     }
48 }
49 void fnDecrypt(char acStr[50])
50 {
51     int i;
52     for(i=0;acStr[i]!='\0';i++)
53     {
54         if(!isspace(acStr[i]))
55             acStr[i] = (acStr[i]-'a'-3+26)%26+'a';
56     }
57 }
58
59 /*
60 $ ./a.out
61
62 Enter your plain text in lower case
63 ramanu kaadige hodanu
64 Encrypted Ciphertext
65 udpdqx nddgljh krgdqx
66 Decrypted Plaintext
67 ramanu kaadige hodanu
68 $ ./a.out
69
70 Enter your plain text in lower case
71 you are what you think you are
72 Encrypted Ciphertext
73 brx duh zkdw brx wklqn brx duh
74 Decrypted Plaintext
75 you are what you think you are
76
77 */

```

Listing 7.1: B07Caesar.c

## Output

## Chapter 8

# Substring Search

### Question

*Write a program to read a line of text from the keyboard and print the number of occurrences of a given substring using the builtin function strstr().*

### C Code

```
1  /*****
2  *File           : B07StrSearch.c
3  *Description    : program to search for a sub string in a given string
4  *Author        : Prabodh C P
5  *Compiler      : gcc compiler, Ubuntu 14.04
6  *Date          : 18 Novemeber 2014
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <string.h>
11
12 int main()
13 {
14     char acText[100], acPattern[10];
15     int iCount = 0;
16     char *pos = NULL, *start = NULL;
17     printf("\nEnter the text\n");
18     gets(acText);
19
20     printf("\nEnter the substring to search for\n");
21     gets(acPattern);
22
23     start = acText;
24     while((pos = strstr(start, acPattern)))
25     {
26
27         iCount++;
28         printf("\nSubstring found at position %ld\n", pos-acText+1);
29         start = pos+1;
30     }
31
32     printf("\nNo of occurences is %d\n", iCount);
33
34     return 0;
35 }
36
37 /*
38 $ ./a.out
```

```
39
40 Enter the text
41 The cat with five legs ran down the hall
42
43 Enter the substring to search for
44 he
45
46 Substring found at position 2
47
48 Substring found at position 34
49
50 No of occurrences is 2
51
52 $ ./a.out
53
54 Enter the text
55 Freedom of the mind
56
57 Enter the substring to search for
58 faith
59
60 No of occurrences is 0
61 */
```

Listing 8.1: B08StrSearch.c

## Output

## Chapter 9

# Student Details

### Question

**Write a program in C that stores the details of N number of students given by the user. The following are the information stored for each student i)Name ii)USN iii)Marks scored in 3 subjects Then find and display details of**

**i Average marks of each student**

**ii Topper of the batch**

### C Code

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 #define STRSIZ 40
6
7 typedef struct {
8     char acName[STRSIZ];
9     char acUsn[11];
10    int iMarks[3];
11 }STUDENT_TYPE;
12
13 int main()
14 {
15     STUDENT_TYPE s[100];
16     int iNum, i, iMax;
17     float fAvg, fMaxAvg=0.0f;
18
19     printf("\nEnter the no. of students\n");
20     scanf("%d", &iNum);
21
22     printf("\nEnter students info\n");
23     for(i=0; i<iNum; i++)
24     {
25         printf("\nStudent %d\n=====\n", i+1);
26         printf("\nName : ");
27         scanf("%s", s[i].acName);
28         printf("\nUSN : ");
29         scanf("%s", s[i].acUsn);
30         printf("\nMarks 1 : ");
31         scanf("%d", &s[i].iMarks[0]);
32         printf("\nMarks 2 : ");
33         scanf("%d", &s[i].iMarks[1]);
34         printf("\nMarks 3 : ");
```

```
35     scanf("%d", &s[i].iMarks[2]);
36 }
37
38 printf("%15s%12s%4s%4s%4s%8s\n", "Name", "USN", "M1", "M2", "M3", "AVG");
39 for(i=0;i<iNum;i++)
40 {
41     fAvg = (s[i].iMarks[0] + s[i].iMarks[1] + s[i].iMarks[2])/3.0;
42     printf("%15s%12s%4d%4d%4d%8.2f\n",s[i].acName, s[i].acUsn, s[i].iMarks[0],
43     s[i].iMarks[1], s[i].iMarks[2], fAvg);
44     if(fAvg > fMaxAvg)
45     {
46         fMaxAvg = fAvg;
47         iMax = i;
48     }
49 }
50 printf("\nThe topper of the batch is %s\n", s[iMax].acName);
51 return 0;
52 }
```

Listing 9.1: B09StudentMarks.c

## Output

## Chapter 10

# Bitwise Operations

### Question

**Write a program that reads an unsigned integer and then perform the following operations using user defined functions**

***i check given input is even or odd***

***ii perform of swapping of two numbers***

### C Code

```
1  /*****
2  *File      : B10BitWise.c
3  *Description : program to illustrate bitwise operations
4  *Author    : Prabodh C P
5  *Compiler  : gcc compiler, Ubuntu 14.04
6  *Date     : 30 September 2017
7  *****/
8  #include <stdio.h>
9  #include <stdlib.h>
10 #include <stdbool.h>
11
12 bool fnCheckEven(unsigned);
13 void fnSwapNums(int*, int*);
14
15 int main(void)
16 {
17     unsigned uNum;
18     int iVal1, iVal2;
19
20     printf("\nEnter a value\n");
21     scanf("%u", &uNum);
22
23     if(fnCheckEven(uNum))
24         printf("\n%u is a Even number\n", uNum);
25     else
26         printf("\n%u is a Odd number\n", uNum);
27
28     printf("\nEnter two values\n");
29     scanf("%d%d", &iVal1, &iVal2);
30
31     printf("\nValues before swapping\n%d\t%d\n", iVal1, iVal2);
32     fnSwapNums(&iVal1, &iVal2);
33     printf("\nValues after swapping\n%d\t%d\n", iVal1, iVal2);
34     return 0;
35 }
```

```
36
37 bool fnCheckEven(unsigned iVal)
38 {
39     int iRes;
40     iRes = iVal & 1;
41     if(iRes == 0)
42         return true;
43     else
44         return false;
45 }
46
47 void fnSwapNums(int *iX, int *iY)
48 {
49     *iX = *iX ^ *iY;
50     *iY = *iY ^ *iX;
51     *iX = *iX ^ *iY;
52 }
```

Listing 10.1: B10BitWise.c

## Output