# Programming for Problem Solving Through C Language

## LAB CYCLES
## 1-12

BY
DEVATHI SAI MANI KUMAR
T20CSB051
RVR JC COLLEGE OF ENGINEERING

# Lab1 Simple computational problems using arithmetic expressions

a. **For the given Basic salary, compute DA, HRA and PF using the following criteria and write a C program to find out the Net Salary of an Employee by deducting PF and IT.**

**DA = (Basic salary *25)/1000**

**HRA= (Basic salary * 15)/100**

**Gross salary = Basic salary + DA + HRA**

**PF = Gross salary * 10/100**

**IT= Gross salary * 10/100**

**Net Salary = Basic Salary + DA + HRA - (PF + IT)**

```
#include <stdio.h>
void main()
{
   int salary;
   float net_salary, da, hra, pf,it, gross_salary;
   printf("Enter Basic salary");
   scanf("%d", &salary);
   da = (salary*25) / 100;
   hra = (salary*15) / 100;
   gross_salary = salary + da + hra;
   pf = (gross_salary*10) / 100;
   it = (gross_salary*10) / 100;
   net_salary = gross_salary-(pf + it);
   printf("Net Salary is %.2f", net_salary);
}
```

**INPUT 1:** Enter Basic salary 200

**OUTPUT 1:** Net Salary is 224

**INPUT 2:** Enter Basic salary 36550

**OUTPUT 2:** Net Salary is 40935.20

b. **Write a program to Convert the given minutes into days, hours and remaining minutes**
**Ex: M=2000 Answer is 2000 minutes contain 1 day 9 Hours 20 Minutes**

```
#include <stdio.h>
void main()
{
   int m, d, h;
   printf("Enter Minutes!");
   scanf("%d", &m);
   h = m / 60;
   m = m % 60;
   d = h / 24;
   h = h % 24;
   printf("%d days %d hours %d minutes", d, h, m);
}
```

**INPUT 1:** 3500

**OUTPUT 1:** 2 days 20 hours 20 minutes

**INPUT 2:** 2000

**OUTPUT 2:** 1 days 9 hours 20 minutes

# Lab2 Problems involving if-then-else & switch

**a. The marks obtained by a student in n different subjects are given as an input by the user. Write a C program that calculates the average marks of given n subjects and displays the grade. The student gets a grade as per the following rules:**

| **Average** | **Grade** |
|---|---|
| **90-100** | **O** |
| **80-89** | **E** |
| **70-79** | **A** |
| **60-69** | **B** |
| **50-59** | **C** |
| **<50** | **F** |

```
#include <stdio.h>
void main()
{
   int n, i, sum = 0, x;
   float avg;
   printf("Enter number of different subjects ");
   scanf("%d", &n);
   for (i = 1; i <= n; i++)
   {
      scanf("%d", &x);
      sum += x;
   }
   avg = sum / n;
   if (avg <= 100 && avg >= 90)
      printf("GRADE is O");
   else if (avg <= 89 && avg >= 80)
```

```
        printf("GRADE is E");
    else if (avg <= 79 && avg >= 70)
        printf("GRADE is A");
    else if (avg <= 69 && avg >= 60)
        printf("GRADE is B");
    else if (avg <= 59 && avg >= 50)
        printf("GRADE is C");
    else
        printf("GRADE is F ");
}
```

**INPUT 1:** Enter number of different subjects 6
78 69 89 90 95 76

**OUTPUT 1:** GRADE is E

**INPUT 2:** Enter number of different subjects 5
60 65 55 91 89

**OUTPUT 2:** GRADE is A

**b. Write a C program to find the Largest and smallest values among the given 4 values.**

```
#include <stdio.h>
void main()
{
    int n, min, max, i;
    printf("Enter four numbers: ");
    scanf("%d", &n);
    max = n;
    min = n;
    for (i = 1; i < 4; i++)
    {
        scanf("%d", &n);
        if (max < n)
            max = n;
        if (min > n)
            min = n;
    }
    printf("        smallest and largest numbers are %d,%d", min, max);
}
```

```
INPUT 1:      Enter four numbers: 1 2 3 4
OUTPUT 1:     smallest and largest numbers are 1,4
INPUT 2:      Enter four numbers: 54 67 34 89
OUTPUT 2:     smallest and largest numbers are 34,89
```

**c. Write a C program to Print the month name when month number is given and Print the week name when week number is given.**

```
#include <stdio.h>
void main()
{
    int month, week;
    printf("enter month no, week no:");
    scanf("%d%d", &month, &week);
    switch (month)
    {
    case 1:
        printf(" January ");
        break;
    case 2:
        printf(" February ");
        break;
```

```
case 3:
    printf(" March ");
    break;
case 4:
    printf(" April ");
    break;
case 5:
    printf(" May ");
    break;
case 6:
    printf(" June ");
    break;
case 7:
    printf(" July ");
    break;
case 8:
    printf(" August ");
    break;
case 9:
    printf("September ");
    break;
case 10:
    printf(" October ");
    break;
case 11:
    printf("November ");
    break;
case 12:
    printf("December ");
    break;
default:
        printf("INVALID");
    }
    switch (week)
    {
    case 1:
        printf("Monday");
        break;
    case 2:
        printf("Tuesday ");
        break;

    case 3:
        printf("Wednesday");
        break;
    case 4:
        printf("Thursday");
        break;
    case 5:
        printf("Friday");
        break;
    case 6:
        printf("Saturday ");
        break;
    case 7:
        printf("Sunday ");
        break;
    default:
        printf(" INVALID");
    }
}
```

**INPUT 1:** enter month no, week no: 1 5

**OUTPUT 1:** January Friday

**INPUT 2:** enter month no, week no: 7 12

**OUTPUT 2:** July  INVALID

d. **Write a C program to Check whether the given character is numeric or non-numeric**.

```
#include <stdio.h>

void main()

{

    char c;

    printf(" please enter the character: ");

    scanf("%c", &c);

    if (c >= '0' && c <= '9')

        printf(" Numeric");

    else

        printf("Non-Numeric ");

}
```

**INPUT 1:** please enter the character: 4

**OUTPUT 1:** Numeric

**INPUT 2:** please enter the character: *

**OUTPUT 2:** Non-Numeric

# Lab3 Iterative problems

a. **Write a C program to read in two numbers, x and n, and then compute the sum of this geometric progression: $1+x+x^2+x^3+$………… $+x^n$. For example: if n is 3 and x is 5, then the program computes 1+5+25+125. Print x, n, the sum. Perform error checking. For example, the formula does not make sense for negative exponents – if n is less than 0. Have your program print an error message if n is negative**

```
#include <stdio.h>
#include <math.h>
void main()
{
   int x, n, i, sum = 1, y;
   printf(" Enter x, n Values");
   scanf("%d %d", &x, &n);
   if (n >= 0)
   {
      for (i = 0; i < n; i++)
      {
         y = x * (pow(x, i));
         sum = sum + y;
      }
      printf("%d", sum);
   }
   else
   {
      printf("ERROR");
   }
}
```

**INPUT 1:** Enter x, n Values 5 3

**OUTPUT 1:** 155

**INPUT 2:** Enter x, n Values 5 -1

**OUTPUT 2:** ERROR

**b. Write a C Program to print Floyd triangle for the user given number of rows. If the user entered 4 rows, then the output follows:**

```
1
2   3
4   5  6
7   8  9 10
```

```
#include <stdio.h>
void main()
{
   int n, i, j, k = 1;
   printf("ENTER N Value "); // n is no. of rows
   scanf("%d", &n);
   for (i = 1; i < n; i++)
   {
      for (j = 1; j <= i; j++)
         printf("%d ", k++);
      printf("\n");
   }
}
```

**INPUT 1:** ENTER N Value 6

**OUTPUT 1:**

```
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
```

c. **Write a C Program to check whether the given number is a palindrome or not.**

**Hint: To check whether a number is a palindrome or not, reverse the given number And compare the reversed number with the given number, if both are same then the Number is palindrome, otherwise not.**
**Example: Given Number = 121, Reversed number = 121. Hence, given number is Palindrome.**

```
#include <stdio.h>
void main()
{
   int sum = 0, n, x;
   printf("Enter the number");
   scanf("%d", &n);
   x = n;
   while (x > 0)
   {
      sum = sum * 10 + x % 10;
      x = x / 10;
   }
   if (sum == n)
      printf(" It is Palindrome");
   else
      printf("  It is not palindrome");
}
```

**INPUT 1:** Enter the number12321

**OUTPUT 1:** It is Palindrome

**INPUT 2:** Enter the number1232

**OUTPUT 2:** It is not Palindrome

**d. Write a C program to find all prime numbers, strong numbers and Armstrong numbers between the given two numbers.**

```
#include <stdio.h>
#include <math.h>
void armstrong(int n1, int n2)
{
  int i, sum = 0, d, x, r;
  printf("\nArmstrong numbers between %d and %d are: ", n1, n2);
  for (i = n1; i <= n2; i++)
  {
    x = i;
    d = (int)log10(i) + 1;
    sum = 0;
    while (x > 0)
    {
      r = x % 10;
      sum += pow(r, d);
      x = x / 10;
    }
    if (sum == i)
      printf("%d ", sum);
  }
}
void prime(int n1, int n2)
```

```
{

   int i, j, flag;
   printf("\nPrime numbers between %d and %d are: ", n1, n2);
   for (i = n1; i <= n2; i++)
   {
      if (i == 1 || i == 0)
         continue;
      flag = 1;
      for (j = 2; j < i; j++)
      {
         if (i % j == 0)
         {
            flag = 0;
            break;
         }
      }

      if (flag == 1)
         printf("%d ", i);
   }
}
void strong(int n1, int n2)
{
   int j, i, c = 1, r, sum = 0;
   for (i = n1; i <= n2; i++)
   {
      int x = i;
      sum = 0;
      while (x > 0)
      {
            r = x % 10;
            c = 1;
            for (j = 1; j <= r; j++)
                c = c * j;
            x = x / 10;
            sum = sum + c;
        }
        if (sum == i)
            printf("%d\n", i);
    }
}

void main()
{
    int n1, n2, c;
    printf("Enter n1,n2:  ");
    scanf("%d%d", &n1, &n2);
    printf("Enter '1' for prime//'2' for armstrong//'3' for strong:");
    scanf("%d", &c);
    switch (c)
    {
    case 1:
        prime(n1, n2);
        break;
    case 2:
        armstrong(n1, n2);
        break;
    case 3:
        strong(n1, n2);
        break;
    default:
        printf("Invalid Character");
    }
}
```

**INPUT 1:** Enter n1,n2: 1 10000

Enter '1' for prime//'2' for armstrong//'3' for strong:2

**OUTPUT 1:** Armstrong numbers between 1 and 10000 are: 1 2 3 4 5 6 7 8 9 370 371 407 1634 8208 9474

**INPUT 2:** Enter n1,n2: 1 10000

Enter '1' for prime//'2' for armstrong//'3' for strong:3

**OUTPUT 2:** 1 2 145

**e. Convert the given rupees into text**

**Ex: 123 One Hundred and Twenty-Three Only**

```
#include <stdlib.h>

#include <stdio.h>

void main()

{

    long num, div, n1;

    int flag, digit, pos, tot_dig;

    printf("\nEnter a number: ");

    scanf("%ld", &num);

    if (num == 0)

    {

        printf("Zeron\n");

        exit(0);
```

```
}
if (num > 99999)
{
    printf("please enter a number between 0 and 100000\n\n");
    exit(0);
}
tot_dig = 0;
div = 1;
n1 = num;
while (n1 > 9)
{
    n1 = n1 / 10;
    div = div * 10;
    tot_dig++;
}
tot_dig++;
pos = tot_dig;
while (num != 0)
{
    digit = num / div;
    num = num % div;
    div = div / 10;
    switch (pos)
```

```
{
case 2:
case 5:
   if (digit == 1)
      flag = 1;
   else
   {
      flag = 0;
      switch (digit)
      {
      case 2:
         printf("twenty ");
         break;
      case 3:
         printf("thirty ");
         break;
      case 4:
         printf("forty ");
         break;
      case 5:
         printf("fifty ");
         break;
      case 6:
            printf("sixty ");

            break;

        case 7:

            printf("seventy ");

            break;

        case 8:

            printf("eighty ");

            break;

        case 9:

            printf("ninety ");

        }

    }

    break;

case 1:

case 4:

case 3:

    if (flag == 1)

    {

        flag = 0;

        switch (digit)

        {

        case 0:

            printf("ten ");
            break;

        case 1:

            printf("eleven ");

            break;

        case 2:

            printf("twelve ");

            break;

        case 3:

            printf("thirteen ");

            break;

        case 4:

            printf("fourteen ");

            break;

        case 5:

            printf("fifteen ");

            break;

        case 6:

            printf("sixteen ");

            break;

        case 7:

            printf("seventeen ");

            break;

        case 8:
            printf("eighteen ");
            break;
        case 9:
            printf("nineteen ");
        }
    }
    else
    {
        switch (digit)
        {
        case 1:
            printf("one ");
            break;
        case 2:
            printf("two ");
            break;
        case 3:
            printf("three ");
            break;
        case 4:
            printf("four ");
            break;
        case 5:
            printf("five ");
            break;
        case 6:
            printf("six ");
            break;
        case 7:
            printf("seven ");
            break;
        case 8:
            printf("eight ");
            break;
        case 9:
            printf("nine ");
        }
    }
    if (pos == 3)
    {
        printf("hundred ");
        break;
    }
    if (pos == 4)
        printf("thousand ");
    break;
```

```
        }
        pos--;
    }
    if (pos == 4 && flag == 0)
        printf("thousand");
    else if (pos == 4 && flag == 1)
        printf("ten thousand");
    if (pos == 1 && flag == 1)
        printf("ten ");
    printf("rupees only");
}
```

**INPUT 1:** Enter a number: 123

**OUTPUT 1:** one hundred twenty three rupees only

**INPUT 2:** Enter a number: 99990

**OUTPUT 2:** ninety nine thousand nine hundred ninety rupees only

# Lab4 1D Array manipulation

**a. Write a C program for the following operations on a 1d array**

**i. Insert at a position**

**ii.Delete an element**

**iii.Find the sum of all elements**

**iv.Count of positive, negative, odd and even numbers**

**v.Search for an element**

**vi. Interchange two position values**

```
#include <stdio.h>
int read(int a[10], int n)
{
    int i;
    for (i = 0; i < n; i++)
        scanf("%d", &a[i]);
    return a[10];
}
void write(int a[10], int n)
{
    int i;
    for (i = 0; i < n; i++)
        printf("%d ", a[i]);
}
void main()
{
    int i, n, pos, ele, sum = 0, a[10], choice, flag = 0;
    int posi = 0, neg = 0, odd = 0, even = 0;
    int t, pos1, pos2;
    printf("Enter the size:");
    scanf("%d", &n);
    printf("Enter the elements:\n");
    read(a, n);
    printf("Given array is:\n");
    write(a, n);
    do
    {
        printf("\nEnter your choice:\n1.Insert at a position\n2.Delete an element\n3.Sum of all elements\n4.count of positive,negative,odd and even Nos.\n5.Search for an element\n6.Interchange two position values\n7.Exit\n");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            printf("Enter the position and element to insert:");
            scanf("%d %d", &pos, &ele);
            for (i = n; i >= pos; i--)
                a[i] = a[i - 1];
            a[pos] = ele;
            n += 1;
            printf("The array after Insertion is:\n");
            write(a, n);
            break;
        case 2:
            printf("Enter the position to delete:");
            scanf("%d", &pos);
            for (i = pos; i < n; i++)
                a[i - 1] = a[i];
            n -= 1;
            printf("The array after deletion:\n");
            write(a, n);
            break;
        case 3:
```

```
    for (i = 0; i < n; i++)
        sum += a[i];
    printf("Sum of elements is %d", sum);
    break;
case 4:
    for (i = 0; i < n; i++)
    {
        if (a[i] > 0)
            posi++;
        else
            neg++;
        if (a[i] % 2 == 0)
            even++;
        else
            odd++;
    }
    printf("No of positive Nos. are %d.\n", posi);
    printf("No of negative Nos. are %d.\n", neg);
    printf("No of even Nos. are %d.\n", even);
    printf("No of odd Nos. are %d.\n", odd);
    break;
case 5:
    printf("Enter the element to search:");
    scanf("%d", &ele);
    for (i = 0; i < n; i++)
    {
        if (ele == a[i])
        {
            flag = 1;
            break;
        }
```

```
      }
      if (flag == 1)
        printf("Data is found");
      else
        printf("Data not found");
      break;
    case 6:
      printf("Array before swapping:\n");
      write(a, n);
      printf("Enter the positions to swap:");
      scanf("%d %d", &pos1, &pos2);
      t = a[pos1];
      a[pos1] = a[pos2];
      a[pos2] = t;
      printf("Array after swapping:\n");
      write(a, n);
      break;
    }
  } while (choice < 7);
}
```

**INPUT 1:**

```
Enter the size: 4
Enter the elements: 2 3 4 6
Given array is: 2 3 4 6
Enter your choice:
1.Insert at a position
2.Delete an element
3.Sum of all elements
4.count of positive,negative,odd and even Nos.
5.Search for an element
6.Interchange two position values
7.Exit
```

```
3
```

**OUTPUT 1:**

```
Sum of elements is 15
Enter your choice:
1.Insert at a position
2.Delete an element
3.Sum of all elements
4.count of positive,negative,odd and even Nos.
5.Search for an element
6.Interchange two position values
7.Exit
7
```

## Lab5 Problems on 2D arrays and Strings

**a. Write a C program to**

**i. Read a Matrix, and find the transpose of the matrix**

```
#include<stdio.h>
void read(int a[10][10],int m,int n)
{
   int i,j;
   for(i=0;i<m;i++)
      for(j=0;j<n;j++)
      scanf("%d",&a[i][j]);
}
void write(int a[10][10],int m,int n)
{
   int i,j;
   for(i=0;i<m;i++){
      printf("\n");
      for(j=0;j<m;j++){
         printf("%d ",a[i][j]);
      }
   }
}
void main()
{
    int choice;
    int a[10][10],b[10][10],c[10][10],i,j,k;
    int m,n,m1,n1,m2,n2;

        printf("Enter the sizes:");
        scanf("%d %d",&m,&n);
        printf("Enter the elements:\n");
        read(a,m,n);
        printf("\nEntered matrix is:\n");
        write(a,m,n);
        for(i=0;i<m;i++)
        for(j=0;j<n;j++){
          b[i][j]=a[j][i];
        }
        printf("\nThe transpose of the given matrix is:\n");
        write(b,m,n);
}
```

**INPUT 1:**

```
Enter the sizes:3 3
Enter the elements:
1 2 3 4 5 6 7 8 9
```

**OUTPUT 1:**

```
Entered matrix is:
1 2 3
4 5 6
7 8 9
The transpose of the given matrix is:
1 4 7
2 5 8
3 6 9
```

### ii. Addition of two matrices

```
#include <stdio.h>
void read(int a[10][10], int m, int n)
{
    int i, j;
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
```

```
         scanf("%d", &a[i][j]);
}
void write(int a[10][10], int m, int n)
{
   int i, j;
   for (i = 0; i < m; i++)
   {
      printf("\n");
      for (j = 0; j < m; j++)
      {
         printf("%d ", a[i][j]);
      }
   }
}
void main()
{
   int choice;
   int a[10][10], b[10][10], c[10][10], i, j, k;
   int m, n, m1, n1, m2, n2;
   printf("Enter the sizes of matrix:");
   scanf("%d %d", &m, &n);
   printf("Enter the elements of a :\n");
   read(a, m, n);
   printf("Enter the elements of b :\n");
   read(b, m, n);
   printf("The entered matrices are:\n");
   write(a, m, n);
   printf("\n");
   write(b, m, n);
   printf("\nThe addition of two matrices are:\n");
   for (i = 0; i < m; i++)
      for (j = 0; j < n; j++)
         c[i][j] = a[i][j] + b[i][j];
   write(c, m, n);
}
```

```
INPUT 1:     Enter the sizes of matrix:2 2

             Enter the elements of a :

             1 2 3 4

             Enter the elements of b :

             1 2 3 4
```

**OUTPUT 1:**

```
The entered matrices are:
1 2
3 4


1 2
3 4
The addition of two matrices are:
2 4
6 8
```

### iii. Multiplication of two matrices with prompting of rules

```
#include <stdio.h>
void read(int a[10][10], int m, int n)
{
   int i, j;
   for (i = 0; i < m; i++)
      for (j = 0; j < n; j++)
         scanf("%d", &a[i][j]);
}
void write(int a[10][10], int m, int n)
{
   int i, j;
   for (i = 0; i < m; i++)
   {
      printf("\n");
      for (j = 0; j < m; j++)
      {
         printf("%d ", a[i][j]);
      }
   }
}
void main()
{
   int choice;
```

```
    int a[10][10], b[10][10], c[10][10], i, j, k;
    int m, n, m1, n1, m2, n2;

    printf("Enter the size of both matrices:");
    scanf("%d %d %d %d", &m1, &n1, &m2, &n2);
    if (n1 == m2)
    {
        printf("Enter the elements of Matrix a:\n");
        read(a, m1, n1);
        printf("Enter the elements of Matrix b:\n");
        read(b, m2, n2);
        printf("The entered matrices are:\n");
        write(a, m1, n1);
        printf("\n");
        write(b, m2, n2);
        for (i = 0; i < m1; i++)
            for (j = 0; j < n2; j++)
            {
                c[i][j] = 0;
                for (k = 0; k < n1; k++)
                {
                    c[i][j] += a[i][k] * b[k][j];
                }
            }
        printf("\nThe multiplication of two matrices is:\n");
        write(c, m1, n2);
    }
    else
        printf("The matrix multiplication is not possible.");
}
```

**INPUT 1:**

```
Enter the size of both matrices:3 3 3 3

Enter the elements of Matrix a:

1 2 3 4 5 6 7 8 9

Enter the elements of Matrix b:

9 8 7 6 5 4 3 2 1
```

**OUTPUT 1:**

```
The entered matrices are:

1 2 3

4 5 6

7 8 9
```

```
9 8 7
6 5 4
3 2 1
The multiplication of two matrices is:

30 24 18
84 69 54
138 114 90
```

**INPUT 2:** Enter the size of both matrices:3 2 3 2

**OUTPUT 2:** The matrix multiplication is not possible.

**b. Write a C program that uses functions to perform the following operations:**

i. **To insert a sub string into a given main string from a given position.**

```
#include <stdio.h>
#include <string.h>
void string(char str[], int pos, int l)
{
  int i = 0, n = strlen(str);

  for (int j = 0; j < l; j++)
  {
    for (i = pos - 1; i < n; i++)
    {
      str[i] = str[i + 1];
    }
    printf("%c ", str[i]);
    n--;
  }
```

```
    str[n] = '\0';
    printf("%s", str);
}
void main()
{
    char str[100];
    int pos, l;
    printf("enter string    ");
    gets(str);
    printf("enter the position  ");
    scanf("%d", &pos);
    printf("enter the number of characters ");
    scanf("%d", &l);
    string(str, pos, l);
}
```

```
INPUT 1:      enter string    sai mani kumar
              enter the position  5
              enter the new word   mani
OUTPUT 1:     sai manimani kumar
```

**ii. To delete n characters from a given position in a given string.**

```
#include <stdio.h>
#include <string.h>
void string(char str[], int pos, int l)
{
    int i = 0, n = strlen(str);

    for (int j = 0; j < l; j++)
    {
        for (i = pos - 1; i < n; i++)
        {
            str[i] = str[i + 1];
        }
        printf("%c ", str[i]);
        n--;
    }
    str[n] = '\0';
    printf("the string after deleting %s", str);
}
void main()
{
    char str[100];
    int pos, l;
    printf("enter string    ");
    gets(str);
    printf("enter the position  ");
    scanf("%d", &pos);
    printf("enter the number of characters ");
    scanf("%d", &l);
    string(str, pos, l);
}
```

**INPUT 1:**

```
enter string    sai mani kumar
enter the position  4
enter the number of characters 6
```

**OUTPUT 1:**

```
the string after deleting saikumar
```

**INPUT 2:**

```
enter string    hi this is the the world
enter the position  11
enter the number of characters 4
```

**OUTPUT 2:**

```
 hi this is the world
```

**c. Write a C program to find a string within a sentence and replace it with another string.**

```
#include <stdio.h>
#include <string.h>

void replaceSubstring(char[], char[], char[]);

void main()
{
    char string[100], sub[100], new_str[100];
    printf("\nEnter a string: ");
    gets(string);
    printf("\nEnter the substring: ");
    gets(sub);
    printf("\nEnter the new substring: ");
    gets(new_str);
    replaceSubstring(string, sub, new_str);
}

void replaceSubstring(char string[], char sub[], char new_str[])
{
    int stringLen, subLen, newLen;
    int i = 0, j, k;
    int flag = 0, start, end;
    stringLen = strlen(string);
    subLen = strlen(sub);
    newLen = strlen(new_str);

    for (i = 0; i < stringLen; i++)
    {
        k = 0;
        for (j = i; j < i + subLen; j++)
        {
            if (string[j] == sub[k])
                k++;
            else
                break;
        }
        if (k == subLen)
        {
            start = i;
            end = j;
            flag = 1;
            break;
        }
    }

    if (flag == 0)
    {
        printf("SUBSTRING WAS NOT FOUND!!!");
    }
    else
    {
        for (j = start; j < end; j++)
        {
            for (k = start; k < stringLen; k++)
                string[k] = string[k + 1];
            stringLen--;
            i--;
        }
        for (j = start; j < start + newLen; j++)
        {
            for (k = stringLen; k >= j; k--)
                string[k + 1] = string[k];
            string[j] = new_str[j - start];
            stringLen++;
            i++;
        }
        printf("\nThe string after replacing : %s\n", string);
    }
}
```

```
INPUT 1:      Enter a string: sai mani kumar
              Enter the substring: sai
              Enter the new substring: devathi sai

OUTPUT 1:     The string after replacing : devathi sai mani kumar

INPUT 2:      Enter a string: sai mani kumar
              Enter the substring: saikumar
              Enter the new substring: devathi sai
OUTPUT 2:     SUBSTRING WAS NOT FOUND!!!
```

**d. Write a C program that reads a line of text and counts all occurrences of a particular word.**

```
#include <stdio.h>
#include <string.h>
void main()
{
    char str[100] = {0}, word[100], substr[100][100] = {0};
    printf("Enter the sentence:  ");
    gets(str);
    printf("Enter the word  ");
    gets(word);
    int i = 0, j = 0, k = 0,count=0;
    char c;
    while (str[k] != '\0') //for splitting sentence into words
    {
        j = 0;
        while (str[k] != ' ' && str[k] != '\0' && str[k] != '.')
        {
            substr[i][j] = str[k];
            k++;
            j++;
        }
        substr[i][j] = '\0';
        if (str[k] != '\0')
        {
            k++;
        }
        if (strcmp(substr[i], word) == 0)
        {
            count++;
        }
        i++;
    }
        printf("Number of occurrences of the word %s is %d", word, count);
}
```

**INPUT 1:** Enter the sentence:  SAI MANI KUMAR

Enter the word  SAI

**OUTPUT 1:** Number of occurrences of the word SAI is 1

**INPUT 2:** Enter the sentence: hello everyone hello world

Enter the word hello

**OUTPUT 2:** Number of occurrences of the word hello is 2

**e. Write a C program for the following:**

**Ram wanted to increase his typing speed to participate in programming contests. His friend suggested that type the sentence "The quick brown fox jumps over the lazy dog" repeatedly. This sentence is known as a pangram because it contains every letter of the alphabet. After typing the sentence several times, Ram became bored with it so he started to look for other pangrams. For this task, read a sentence from the user and store it in a character array 's' Hint: Allocate memory for the string using dynamic memory allocation and determine whether the given string is a pangram( or not. Ignore upper or lower cases.**

```
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

int main()

{

   char *str, arr[26] = {0};

   int i;



   str = (char *)malloc(100 * (sizeof(char)));

   if (str == NULL)

   {

      printf("memory is not allocated");

      return 0;
```

```
}
printf("Enter the string:   ");
gets(str);
strlwr(str);
int k;
for (i = 0; str[i] != '\0'; i++)
{
    if (str[i] >= 'a' || str[i] <= 'z')
    {
        k = (int)str[i];
        (arr[k - 97])++;
    }
}
int flag = 0;
for (i = 0; i < 26; i++)
{
    if (arr[i] >= 1)
        flag = 1;
    else
    {
        flag = 0;
        break;
    }
    }
    if (flag == 1)
    {
        printf("The given string in Pangram");
    }
    else
    {
        printf("The given string is Not Pangram");
    }
    free(str);
    return 0;
}
```

**INPUT 1:** Enter the string: The quick brown fox jumps over the lazy dog

**OUTPUT 1:** The given string in Pangram

**INPUT 2:** Enter the string: How vexingly quick daft zebras jump!

**OUTPUT 2:** The given string in Pangram

# Lab6 Function calling mechanisms (Call by value)

**a. Write a C program to find prime Fibonacci numbers using functions.**

```
#include<stdio.h>
void prime(int x)
{
    int i,y=0;
    for(i=1;i<x;i++)
    {
        if(x%i==0)
        {
            y++;
        }
    }
    if(y==1)
    {
        printf("%d\t",x);
    }
}
void fibonacciSeries(int n)
{
   int a=0, b=1, c;
   for(int i=0; i<n; i++)
   {
     prime(a);
     c = a+b;

     a = b;
     b = c;
   }
}

int main()
{
   int term;

   printf("Enter the term: ");//no fibonacci prime numbers do you need
   scanf("%d", &term);

   printf("The prime fibonacci series is: \n");
```

```
    fibonacciSeries(term);

    return 0;
  }
```

**INPUT 1:** Enter the term: 10

**OUTPUT 1:** The prime fibonacci series is:

```
2	3	5	13
```

**INPUT 2:** Enter the term: 20

**OUTPUT 2:** The prime fibonacci series is:

```
2	3	5	13	89	233	1597
```

**b. Write a C program to find npr, ncr values by writing factorial function.**

```
#include <stdio.h>
double fact(double);
double find_npr(double, double);
double find_ncr(double, double);

int main()
{

    double n, r;
    double ncr, npr;

    printf("Enter the value of n and r\n");
    scanf("%lf%lf", &n, &r);
    if (n < r)
    {
        printf("n should be greater than or equal to r");
        return 0;
    }
    ncr = find_ncr(n, r);
    npr = find_npr(n, r);

    printf("the combination %.lfC%.lf = %.lf\n", n, r, ncr);
    printf("the permutation %.lfP%.lf = %.lf\n", n, r, npr);

    return 0;
}

double find_ncr(double n, double r)
{
    double result;

    result =(double) fact(n) / (double) (fact(r) * fact(n - r));

    return result;
}

double find_npr(double n, double r)
{
    double  result;

    result = (long)fact(n) / (long)fact(n - r);

    return result;
```

```
}


double fact(double n)
{
    double c;
    double result = 1;

    for (c = 1; c <= n; c++)
        result = result * c;

    return result;
}
```

```
INPUT 1:      Enter the value of n and r
              5
              4
OUTPUT 1:     the combination 5C4 = 5
              the permutation 5P4 = 120

INPUT 2:      Enter the value of n and r
              4
              5
OUTPUT 2:     n should be greater than or equal to r
```

# Lab7 Function calling mechanisms (Call by reference)

**a. Write a C program to perform all operations on strings using functions**

**i. Find the length of a string**

```
#include<stdio.h>
int stringlen(char *str)
{
   int i;
   for(i=0;*(str+i)!='\0';i++);
   return i;
}
void main()
{
   char str[100];
   int l;
   printf("Enter the string:   ");
   gets(str);
   l=stringlen(str);
   printf("The length of the string is %d",l);
}
```

**INPUT 1:** Enter the string:   sai mani kumar

**OUTPUT 1:** The length of the string is 14

**INPUT 2:** Enter the string:   bengal

**OUTPUT 2:** The length of the string is 5

**ii. Find the substring when position and length is given**

```
#include<stdio.h>

#include<stdlib.h>

char *substring(char *str,int pos,int len)

{

   char *substr;
```

```
    substr = malloc(len);

   if (substr == NULL)
   {
      printf("Unable to allocate memory.\n");
      return 0;
   }
   int c;
    for (c = 0; c < len; c++)
   {
      *(substr+c) = *(str+pos-1);
      str++;
   }

   *(substr+c) = '\0';
    return substr;
}
void main()
{
    char str[100],*substr;
    int pos,len;
    printf("Enter the string:   ");
    gets(str);
    printf("Enter the position   ");
    scanf("%d",&pos);
    printf("enter the length    ");
    scanf("%d",&len);
    substr=substring(str,pos,len);
```

```
  printf("The substring is \"%s\"",substr);
}
```

```
INPUT 1:     Enter the string:   sai mani kumar
             Enter the position   5
             enter the length    4
OUTPUT 1:    The substring is "mani"

INPUT 2:     Enter the string:   hello world this sai
             Enter the position   1
             enter the length    11
OUTPUT 2:    The substring is "hello world"
```

## Lab8 Recursive functions

**a. Write C programs that use recursive function**

**i. To find the factorial of a given integer.**

```
#include <stdio.h>

long factorial(int n)
{
   if (n == 0)
      return 1;
   else
      return (n * factorial(n - 1));
}

int main()
{
   int number;
   long fact;
```

```
    printf("Enter a number: ");
    scanf("%d", &number);

    fact = factorial(number);
    printf("Factorial of %d is %ld\n", number, fact);
    return 0;
}
```

**INPUT 1:** Enter a number: 6

**OUTPUT 1:** Factorial of 6 is 720

**INPUT 2:** Enter a number: 10

**OUTPUT 2:** Factorial of 10 is 3628800

**ii. To find the greatest common divisor of two given integers.**

```
#include <stdio.h>
int hcf(int n1, int n2);
int main()
{
    int n1, n2;
    printf("Enter two positive integers: ");
    scanf("%d %d", &n1, &n2);
    printf("G.C.D of %d and %d is %d.", n1, n2, hcf(n1, n2));
    return 0;
}

int hcf(int n1, int n2)
{
    if (n2 != 0)
        return hcf(n2, n1 % n2);
    else
    return n1;
}
```

**INPUT 1:** Enter two positive integers: 5 10

**OUTPUT 1:** G.C.D of 5 and 10 is 5.

**INPUT 2:** Enter two positive integers: 24 5

**OUTPUT 2:** G.C.D of 24 and 5 is 1.

**iii. To print Fibonacci series**.

```
#include <stdio.h>

int fibonacci(int);

void main()

{

  int n, f, n1, n2, i;

  int choice;



  printf("Enter the value:");

  scanf("%d", &n);

  printf("Fibonacci series of %d are:\n", n);

  for (i = 0; i < n; i++)

    printf("%d ", fibonacci(i));

}

int fibonacci(int n)

{

  if (n == 0)
```

```
            return 0;
        else if (n == 1)
            return 1;
        else
            return (fibonacci(n - 1) + fibonacci(n - 2));
    }
```

```
INPUT 1:     Enter the value:5
OUTPUT 1:    Fibonacci series of 5 are:
             0 1 1 2 3

INPUT 2:     Enter the value:15
OUTPUT 2:    Fibonacci series of 15 are:
             0 1 1 2 3 5 8 13 21 34 55 89 144 233 377
```

## Lab9 Dynamic memory allocation

**a. Write a C program to concatenate two strings using pointers.**

```
#include <stdio.h>
#include <stdlib.h>
int main()
{
    char *s1, *s2;
    int i, j = 0;
    s1 = (char *)malloc(50 * sizeof(char));
    if (s1 == NULL)
    {
        printf("Unable to allocate memory");
        return 0;
    }
    s2 = (char *)malloc(25 * sizeof(char));
    if (s2 == NULL)
    {
        printf("Unable to allocate memory");
        return 0;

    printf("enter the string1:   ");
    gets(s1);
    printf("enter the string2:   ");
    gets(s2);
    for (i = 0; *(s1 + i) != '\0'; i++)
        ;
    for (; *(s2 + j) != '\0'; i++, j++)
    {
        *(s1 + i) = *(s2 + j);
    }
    *(s1 + i) = '\0';
    printf("The concatenated string is %s", s1);
}
```

```
INPUT 1:      enter the string1:   sai mani
              enter the string2:    kumar devathi
OUTPUT 1:     The concatenated string is sai mani kumar devathi

INPUT 2:      enter the string1:   mani
              enter the string2:   kumar
OUTPUT 2:     The concatenated string is manikumar
```

**b. Write a C program to find the length of string using pointers.**

```
#include <stdio.h>
#include <malloc.h>
int main()
{
    char *s1;
    int i, j = 0;
    s1 = (char *)malloc(50 * sizeof(char));
    if (s1 == NULL)
    {
        printf("Unable to allocate memory");
        return 0;
    }
    printf("enter the string1:   ");
    gets(s1);
    for (i = 0; *(s1 + i) != '\0'; i++)
        ;
    printf("length of the string is %d", i);
}
```

```
INPUT 1:     enter the string1:   SAI MANI KUMAR
OUTPUT 1:    length of the string is 14

INPUT 2:     enter the string1:   mumbai
OUTPUT 2:    length of the string is 6
```

**c. Write a C program to compare two strings using pointers.**

```
#include <stdio.h>
#include<stdlib.h>
int main()
{
    char *s1, *s2;
    int s1len = 0, s2len = 0;
    s1 = (char *)malloc(50 * sizeof(char));
    if (s1 == NULL)
    {
        printf("Unable to allocate memory");
        return 0;
    }
    s2 = (char *)malloc(50 * sizeof(char));
    if (s2 == NULL)
    {
        printf("Unable to allocate memory");
        return 0;
    }
    printf("enter the string1:   ");
    gets(s1);
    printf("enter the string2:   ");
    gets(s2);
    for (s1len = 0; *(s1 + s1len) != '\0'; s1len++)
        ;
    for (s2len = 0; *(s2 + s2len) != '\0'; s2len++)
        ;
    int flag = 1, i;
    if (s1len = s2len)
    {
        for (i = 0; *(s1 + i) != '\0'; i++)
        {
            if (*(s1 + i) == *(s2 + i))
            {
                flag == 1;
            }
            else
            {
                flag = 0;
                break;
            }
        }
        if (flag == 1)
        {
            printf("Both the strings are same");
        }
        else
        {
            printf("Both the strings are not equal");
        }
    }
    else
    {
        printf("Both the strings are not equal");
    }
}
```

**INPUT 1:** enter the string1: sai mani kumar
enter the string2: sai mani kumar

**OUTPUT 1:** Both the strings are same

```
INPUT 2:     enter the string1:   sai mani kumar
             enter the string2:   sai mani kumar
OUTPUT 2:    Both the strings are not equal
```

**d. Write a C program to copy a string from source to destination using pointers.**

```
#include <stdio.h>

#include <stdlib.h>

int main()

{

   char *s1, *s2;

   s1 = (char *)malloc(50 * sizeof(char));

   if (s1 == NULL)

   {

      printf("Unable to allocate memory");

      return 0;

   }

   s2 = (char *)malloc(50 * sizeof(char));

   if (s2 == NULL)

   {

      printf("Unable to allocate memory");

      return 0;

   }

   printf("enter the string1:   ");

   gets(s1);
```

```
    int i;

    for (i = 0; *(s1 + i) != '\0'; i++)

    {

        *(s2 + i) = *(s1 + i);

    }

    *(s2 + i) = '\0';

    printf("the second string which was copied is : %s", s2);

}
```

**INPUT 1:** enter the string1: sai mani

**OUTPUT 1:** the second string which was copied is : sai mani

**INPUT 2:** enter the string1: teja

**OUTPUT 2:** the second string which was copied is : teja

## Lab10 Structures and unions

a. **Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.**

```
#include <stdio.h>
struct book_detail
{
    int id;
    float price;
    char author[10], title[10];
};
struct book_detail detail();
```

```
void main()
{
   struct book_detail book;
   book = detail();

   printf("Title of the book      :  %s\n", book.title);
   printf("Author of the book    :  %s\n", book.author);
   printf("The id of the book    :  %d\n", book.id);
   printf("The price of the book :  %.2f\n", book.price);
}
struct book_detail detail()
{
   struct book_detail book;
   printf("enter the title of the book\n");
   scanf(" %s", &book.title);
   fflush(stdin);
   printf("enter the author name of the book\n");
   scanf(" %s", &book.author);
   fflush(stdin);
   printf("enter the id of the book\n");
   scanf("%d", &book.id);
   fflush(stdin);
   printf("enter the price of the book\n");
   scanf("%f", &book.price);
   return book;
}
```

**INPUT 1:**

```
enter the title of the book
Let Us C
enter the author name of the book
Yeswanth kanetkar
enter the id of the book
```

```
23456
enter the price of the book
239
```

**OUTPUT 1:**

```
Title of the book        :  Let Us C
Author of the book       :  Yeswanth kanetkar
The id of the book       :  23456
The price of the book :  239.00
```

**b. Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.**

```
#include <stdio.h>
union info
{
   char name[20], home_address[50], hostel_address[50], city[20], state[20], zip[10];
};

void main()
{
   union info s;
   printf("Enter NAME: ");
   gets(s.name);
   printf("NAME                    :   %s\n", s.name);
   printf("Enter CITY: ");
   gets(s.city);
   printf("CITY                    :   %s\n", s.city);
   printf("Enter STATE: ");
   gets(s.state);
   printf("STATE                   :   %s\n", s.state);
   printf("Enter ZIP: ");
```

```
    gets(s.zip);
    printf("ZIP                    :   %s\n", s.zip);
    int n;
    printf("Enter HOSTEL ADDRESS : ");
    gets(s.hostel_address);
    printf("HOSTEL ADDRESS              :  %s\n", s.hostel_address);
    printf("Enter HOME ADDRESS : ");
    gets(s.home_address);
    printf("HOME ADDRESS                : %s\n", s.home_address);
}
```

**INPUT 1:**

```
sai mani kumar
Tenali
Andhra Pradesh
522201
GUNTUR
TENALI
```

**OUTPUT 1:**

```
NAME                 :   sai mani kumar
CITY                 :   Tenali
STATE                :   Andhra Pradesh
ZIP                  :   522201
HOSTEL ADDRESS   :  GUNTUR
HOME ADDRESS       : TENALI
```

c. **Write a C program to define a structure named DOB, which contains name, day, month and year. Using the concept of nested structures display your name and date of birth.**

```
#include <stdio.h>
struct details
```

```
{
    char name[50];
    struct dob
    {
        int date;
        int month;
        int year;
    } date_of_birth;

} s;

void main()
{
    printf("Enter the name:    ");
    gets(s.name);
    fflush(stdin);
    printf("enter the date month year(dd):  ");
    scanf("%d", &s.date_of_birth.date);
    printf("enter the date month year(mm):  ");
    scanf("%d", &s.date_of_birth.month);
    printf("enter the date month year(yyyy):  ");
    scanf("%d", &s.date_of_birth.year);
    printf("NAME            : %s\n", s.name);
    printf("DATE OF BIRTH   : %d-%d-%d", s.date_of_birth.date, s.date_of_birth.month,
s.date_of_birth.year);
}
```

**INPUT 1:**

```
Enter the name:    sai mani kumar
enter the date month year(dd):  07
enter the date month year(mm):  10
enter the date month year(yyyy):  2003
```

**OUTPUT 1:**

```
NAME                  : sai mani kumar
DATE OF BIRTH   : 7-10-2003
```

## Lab11 File operations

**a. Write a C program to display the contents of a file.**

```
#include <stdio.h>
int main()
{
   char ch, fname[30];
   FILE *fp;
   printf("Enter the filename with extension: ");
   gets(fname);
   fp = fopen(fname, "r");
   if (fp == NULL)
   {
      printf("unable to open the file");
      return (0);
   }
   ch = fgetc(fp);
```

```
    while (ch != EOF)
    {
        printf("%c", ch);
        ch = fgetc(fp);
    }
    fclose(fp);
}
```

Contents in my.txt is c programming in my new laptop

**INPUT 1:** my.txt

**OUTPUT 1:** c programming in my new laptop

**b. Write a C program to copy the contents of one file to another.**

```
#include<stdio.h>
int main()
{
    FILE *fp1,*fp2;
    char ch,f1name[30],f2name[30];
    printf("Enter the filename with extension: ");
    gets(f1name);
    fp1=fopen(f1name,"r");
    if (fp1 == NULL)
    {
        printf("unable to open the file");
        return (0);
    }
    printf("Enter the filename with extension: ");
    gets(f2name);
    fp2=fopen(f2name,"w");
    if (fp2 == NULL)
    {
        printf("unable to open the file");
        return (0);
    }
    ch=fgetc(fp1);
    while(ch!=EOF)
    {
        fputc(ch,fp2);
        ch=fgetc(fp1);
    }
    printf("\n\nCONTENT OF THE FILE IS COPIED SUCCESSFULLY\n ");
    fclose(fp1);
    fclose(fp2);
}
```

Contents in file1.txt - Hi this is sai mani kumar

**INPUT 1:** file1.txt
file2.txt

**OUTPUT 1:** CONTENT OF THE FILE IS COPIED SUCCESSFULLY

**c. Write a C program to reverse the first n characters in a file, where n is given by the user.**

```
#include <stdio.h>
int main()
{
    FILE *fp;
    char ch, fname[30], newch[500];
    int i = 0, j, COUNT = 0,n;
    printf("Enter the filename with extension: ");
    gets(fname);
    printf("Enter number of characters that should be reversed: ");
    scanf("%d",&n);
    fp = fopen(fname, "r+");
    if (fp == NULL)
    {
        printf("Unable to open the file");
        return 0;
    }
    printf("\nThe original content is:\n\n");
    ch = fgetc(fp);
    while (COUNT!=n)
    {
        COUNT++;
        putchar(ch);
        newch[i] = ch;
        i++;
        ch = fgetc(fp);
    }
    printf("\n\n\n");
    printf("The content in reverse order of first n characters is:\n\n");
    rewind(fp);
    for (j = n-1; j >= 0; j--)
    {
        ch = newch[j];
        fputc(ch, fp);
        printf("%c", ch);
    }
        printf("\n\nFIRST \"%d\" CHARACTERS OF THE FILE IS REVERSED
SUCCESSFULLY\n ",n);
    fclose(fp);
    return 0;
}
```

Contents of file1.txt abcdefghijklmnopqrstuvwxyz

```
INPUT 1:      file1.txt
              5

OUTPUT 1:     The content in reverse order of first n characters is:
              edcba
              FIRST "5" CHARACTERS OF THE FILE IS REVERSED
              SUCCESSFULLY
```

Contents of file1.txt after executing program edbcafghijklmnopqrstuvwxyz

**d. Two files DATA1 and DATA2 contain sorted lists of integers. Write a C program to merge the contents of two files into a third file DATA i.e., the contents of the first file followed by those of the second are put in the third file.**

```
#include <stdio.h>
int main()
{
    //int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, b[] = {2, 4, 6, 8, 10, 12, 14};
    int a[100],b[100];
    FILE *data1, *data2,*data3;
    data1 = fopen("data1.txt", "a+");
    if (data1 == NULL)
    {
        printf("unable to open the file");
        return (0);
    }
    data2 = fopen("data2.txt", "a+");
```

```
if (data2 == NULL)
{
    printf("unable to open the file");
    return (0);
}
data3 = fopen("data3.txt", "a+");
if (data3 == NULL)
{
    printf("unable to open the file");
    return (0);
}
//fwrite(a, sizeof(a), 1, data1);
//fwrite(b, sizeof(b), 1, data2);
rewind(data1);

fread(a, sizeof(a), 1, data1);
printf("%d",sizeof(a));

// int n = sizeof(a) / sizeof(a[0]);
// printf("%d",n);
for (int i = 0; i < 10; i++)
{
    printf("%d\n", a[i]);
    fputwc(a[i],data3);
}

rewind(data2);
printf("\n\n data 2 is\n");

fread(b, sizeof(b), 1, data2);
printf("%d",sizeof(b));
```

```
    // n = sizeof(b) / sizeof(b[0]);
     printf("\n");
    for (int i = 0; i < 7; i++)
    {
        printf("%d\n", b[i]);
        fputwc(a[i],data3);
    }

    fclose(data1);
    fclose(data2);
    fclose(data3);
}
```

**e. Write a C program to count the no. of characters present in the file.**

```
#include<stdio.h>
int main()
{
        char filename[15],ch;
        FILE *fp;
        int ccount;
        printf("\n Enter an existing file name to be read:");
        scanf("%s",filename);
        fp=fopen(filename,"r"); //open an existing file in read mode
        if(fp==NULL)
        {
                printf("\n Unable to open file");
                return 0;
        }
        ccount=0; //initially, all counts are 0
        ch=fgetc(fp); //read a character from file
        while(ch!=EOF)
        {
                ccount++; //increment character count by 1
                ch=fgetc(fp); //again, read a character from file
        }
        fclose(fp); //close the file
        printf("\n No.of characters=%d\n",ccount);

}
```

The content in the file in sai.txt is - abcdefghi jklmnopqrst uvwxyz

The content in the file in mani.txt is - sai mani kumar

tenali

**INPUT 1:** sai.txt

**OUTPUT 1:** No.of characters=28

**INPUT 2:** mani.txt

**OUTPUT 2:** No.of characters=21

**f. Write a C program to find whether the given word is present in the given file or not. Example: The content of the file is "Computer programming. Computer can do computations". Input: Computer Output: 'Computer' is found at two locations.**

```
#include<stdio.h>

#include<string.h>

int main()

{

     FILE *fp;
```

```
char ch,temp[15],word[15],filename[15];
int count=0;

printf("\n Enter file name:");
scanf("%s",filename);
fp=fopen(filename,"w");
fflush(stdin);
printf("\n Enter the text (end with \n):");
ch=getchar();
while(ch!='\n')
{
        fputc(ch,fp);
        ch=getchar();
}
fclose(fp);
printf("\n Enter the word to be searched:");
scanf("%s",word);
fp=fopen(filename,"r");
while(!feof(fp))
{
        fscanf(fp,"%s",temp);
        if(strcmp(temp,word)==0)
        count++;
```

```
          }

          fclose(fp);

          printf("\n \'%s\' is found at %d locations",word,count);

    }
```

The content in the file is - Computer programming. Computer can do computations

**INPUT 1:** Computer

**OUTPUT 1:** ‘Computer’ is found at 2 locations

**INPUT 2:** can

**OUTPUT 2:** ‘can’ is found at 1 locations

## Lab12 Command line arguments.

**a. Write a C program to print all arguments given through the command line.**

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
    printf("the total no of arguments are:  %d\n", argc);
    printf("arguments are : ");
    for (int i = 0; i < argc; i++)
    {
        printf("%s\n", argv[i]);
    }
    return 0;
}
```

**INPUT 1:**

```
$./a.out sum.c master.txt sai  (0r)  .\sum.exe master.txt sai
```

**OUTPUT 1:**

```
the total no of arguments are:  3
sum.c (or) location
Master.txt
Sai
```

**INPUT 2:**

```
$./a.out sum.c 345 mani kumar  (0r)  .\sum.exe 345 mani kumar
```

**OUTPUT 2:**

```
the total no of arguments are:  4
sum.c (or) location
345
mani
kumar
```

**b. Write a C program to find the sum of N integer numbers using command line arguments.**

```
#include <stdio.h>
int main(int argc, char const *argv[])
{
   printf("the total no of arguments are:  %d\n", argc);
   int x;
   x=atoi(argv[1]);//converts strings to integer
   printf("the number given by the user through command line argument is %d",x);
   printf("The sum of first %d integers is : ",x);
   int sum=0;
   for (int i = 1; i <= x; i++)
   {
      sum=sum+i;
   }
```

```
    printf("    %d",sum);
    return 0;
 }
```

**INPUT 1:** $./a.out sum.c 5  (0r)  .\sum.exe 5

**OUTPUT 1:** The sum of first 5 integers is: 15

**INPUT 2:** $./a.out sum.c 10 (0r)  .\sum.exe 10

**OUTPUT 2:** The sum of first 10 integers is: 45