a.

i. check whether the year is Leap year

```
#include <stdio.h>
void year();
void main()
{
   checkYear();
}
void checkYear()
{
  int year;
  printf("Enter a year: ");
  scanf("%d", &year);
  if (year % 400 == 0)
  {
    printf("%d is a leap year.", year);
  }
  else if (year % 100 == 0)
  {
    printf("%d is not a leap year.", year);
  }
  else if (year % 4 == 0) {
    printf("%d is a leap year.", year);
  }
  else {
    printf("%d is not a leap year.", year);
  }
```

}

```
Enter a year: 2014

2014 is not a leap year.
```

ii. convert binary to hexadecimal

```
#include <stdio.h>

void f();

void main()

{

   convertNum();

}

void convertNum()

{

   long int bv, hv = 0, i = 1, rem;


   printf("Enter the binary number: ");

   scanf("%ld", &bv);

   while (bv != 0)

   {

      rem = bv % 10;

      hv = hv + rem * i;

      i = i * 2;

      bv = bv / 10;

   }

   printf("Equivalent hexadecimal value: %lX", hv);

}
```

```
Enter the binary number: 110

Equivalent hexadecimal value: 6
```

iii. count number of digits in a number

```
#include <stdio.h>

void num();

void main()

{

   cont();

}

void cont()

{

   long long n;

   int count = 0;

   printf("Enter an integer: ");

   scanf("%lld", &n);

   while (n != 0)

   {

      n /= 10;

      ++count;

   }

   printf("Number of digits: %d", count);

}
```

```
Enter an integer: 1234

Number of digits: 4
```

b.

i. Check Armstrong number or not.

```
#include<stdio.h>
int an()
{
   int n,r,p=0,o;
   printf("enter the number=");
   scanf("%d",&n);
   o=n;
   while(n>0)
   {
      r=(n%10)*(n%10)*(n%10);
      p=p+r;
      n=n*0.1;
   }
   if(o==p)
   return p;
   else
   return 0;
}
int main()
{
   int q;
   q=an();
   if(q==0)
   printf("its not an armstrong  number ");
   else
```

```
   printf("its an armstrong number");
   return 0;
}
```

```
enter the number=123

its not an armstrong  number
```

ii. to evaluate the following using loops x + x^3 / 3! + x^5 / 5! +... upto 5 terms

```
#include <stdio.h>
#include <math.h>
int num();
void main()
{
   int res;
   res=num();
   printf("\nThe sum = %d\n",res);
}
int num()
{
        int x,sum,ctr;
        int i,n,m,mm,nn;
        printf("Input the value of x :");
        scanf("%d",&x);
        printf("Input number of terms : ");
        scanf("%d",&n);
        sum =x; m=-1;
        printf("The values of the series: \n");
        printf("%d\n",x);
   for (i = 1; i < n; i++)
```

```
    {
        ctr = (2 * i + 1);
        mm = pow(x, ctr);
        nn = mm * m;
        printf("%d \n",nn);
        sum = sum + nn;
        m = m * (-1);
            }
            return sum;
}
```

```
Input the value of x :2

Input number of terms : 5

The values of the series:

2

-
8

32

-
128

512

The sum = 410
```

iii. Convert temperature Fahrenheit to Celsius

```
#include <stdio.h>
float tc()
{
   float f;
   printf("Enter the Fahrenheit degree ");
   scanf("%f",&f);
   f=(f-32)*5/9;
```

```
    return f;
}
float main()
{
    printf("The Centigrade value of given Fahrenheit degree is \n %.2f Degree",tc());
    return 0;
}
```

```
Enter the Fahrenheit degree 100

The Centigrade value of given Fahrenheit degree is

 37.78 Degree
```

c.

i. check prime number or not

```
#include <stdio.h>
int pn(int a)
{
    int g=0;
    for(int e=2;e<a/2;e++)
    {
        if( a%e==0 )
        {
            g=1;
            break;
        }
    }
    if(g==0)
        printf("given number is prime number.");
    else
        printf("given number is not prime number.");
```

```
}
int main()
{
   int a;
   printf("enter number:");
   scanf("%d",&a);
   pn(a);
   return 0;
}
```

```
enter number:234

given number is not prime number.
```

ii. find all roots of the quadratic equation

```
#include <math.h>
#include <stdio.h>
void root(double,double,double);
void main()
{
   double a, b, c, d, r1, r2, rp, ip;
   printf("Enter coefficients a, b and c: ");
   scanf("%lf %lf %lf", &a, &b, &c);
   root(a,b,c);
}
void root(double a,double b,double c)
{
   double d, r1, r2, rp, ip;
   d = b * b - 4 * a * c;
   if (d > 0)
```

```
    {
        r1 = (-b + sqrt(d)) / (2 * a);
        r2 = (-b - sqrt(d)) / (2 * a);
        printf("root1 = %.2lf and root2 = %.2lf", r1, r2);
    }
    else if (d == 0)
    {
        r1 = r2 = -b / (2 * a);
        printf("root1 = root2 = %.2lf;", r1);
    }
    else
    {
        rp = -b / (2 * a);
        ip = sqrt(-d) / (2 * a);
        printf("root1 = %.2lf+%.2lfi and root2 = %.2f-%.2fi", rp, ip, rp, ip);
    }
}
```

```
Enter coefficients a, b and c: 2.3
4
5.6
root1 = -0.87+1.30i and root2 = -0.87-1.30i
```

iii. find ASCII number to character and character to ASCII number

```
#include <stdio.h>
int con(char a)
{
    printf("the asci value of given char is : %i",a);
}
int main()
{
    char t;
```

```
    printf("Enter the character value :");
    scanf("%c",&t);
    con(t);
    return 0;
}
```

```
Enter the character value :w

the asci value of given char is : 119
```

d.

i. check perfect or abundant or deficient number

ii. calculate factorial of a number

```
#include <stdio.h>
int factorial(int,int);
void main()
{
    int n, i,res;
    unsigned long long fact = 1;
    printf("Enter an integer: ");
    scanf("%d", &n);
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else {
        for (i = 1; i <= n; ++i)
        {
            res=factorial(fact,i);
        }
```

```
      printf("Factorial of %d = %llu", n, fact);
   }
}
int factorial(int fact,int i)
{
  return fact *= i;
}
```

```
Enter an integer: 4

Factorial of 4 = 24
```

iii. count number of digits in a number

```
#include <stdio.h>
int num(int);
void main()
{
   int n;
   int count = 0,res;
   printf("Enter an integer: ");
   scanf("%d", &n);
   while (n != 0)
   {
     res=num(n);
      ++count;
   }
   printf("Number of digits: %d", count);
}
int num(int n)
{
```

```
    return n /= 10;
}
Enter an integer: 1234
Number of Digits: 4
```

e.

i. Largest and Smallest of five numbers

```
#include <stdio.h>
int comp(int a[5],int* i,int* y)
{
    int f,q=a[0],e=a[0];
    for (f=0;f<5;f++)
    {
        if(q<a[f])
        q=a[f];
        if(e>a[f])
        e=a[f];
    }
    *i=q;
    *y=e;
}
int main()
{
    int a[5],f,u;
    printf("Enter the value oa array :");
    for(f=0;f<5;f++)
    {
        scanf("%d",&a[f]);
    }
```

```
   f=0;
   comp(a,&f,&u);
   printf("\nthe largest no. : %d \n the smallest no. : %d",f,u);
   return 0;
}
```

```
Enter the value oa array :2

3

4

5

6
the largest no. : 6

the smallest no. : 2
```

ii. Find Simple interest and compound interest

```
#include<stdio.h>
#include<math.h>
int intrest(int p,int t,int r,int *si,int *amount, int *ci )
{
   *si=(p*t*r)/100;
   *amount=p*pow((1 +r/100),t);
   *ci=amount-p;
}
int main()
{
      int p,t,r,si,amount,ci;
      printf("Please enter principal,time and rate of interest:");
      scanf("%d%d%d",&p,&t,&r);
      intrest(p,t,r,&si,&amount,&ci);
      printf("\nSimple interest = %d",si);
      printf("\nCompound interest = %d",ci);
```

```
    return 0;
}
```

```
Please enter principal,time and rate of interest:10

20

30

Simple interest = 60

Compound interest = 94597600
```

iii. simple calculator (add, sub, mul, div, mod)

```
#include <stdio.h>

int comp(int x,int z,int* a,int* s, int* m,int* d,int* mo)

{

  *a=x+z;

  *s=x-z;

  *m=x*z;

  *d=x/z;

  *mo=x%z;

}

int main()

{

int x,z,a,s,m,d,mo;

   printf("enter the value if x & z : \n");

   scanf("%d %d",&x,&z);

   comp(x,z,&a,&s,&m,&d,&mo);

   printf("\n the addition is : %d \n the subtraction : %d \n the multiplication is : %d ",a,s,m);

   printf("\n the division is : %d \n the module is : %d",d,mo);

   return 0;

}
```

```
enter the value if x & z :

3

4
 the addition is : 7

 the subtraction : -
1

 the multiplication is : 12

 the division is : 0

 the module is : 3
```

f.

i. Print the sum of series 1 + 1/2 + 1/3 + 1/4 + ... + 1/N.

```
#include <stdio.h>

int main()

{

   float a;

   auto float input(){

   float z,c;

   printf("enter the value of N :");

   scanf("%f",&c);

   float proc(float n)

   {

      float o,b=0;

      for (o=1;o<n;o++)

      {

         b=b+1/o;

      }

      printf(" the value of the series is : %f",b);

   }

   proc(c);
```

```
}
   input();
   return 0;
}
```

```
enter the value of N :4
 the value of the series is : 1.833333
```

ii. Find GCD and LCM of numbers

iii. reverse a number

```
#include <stdio.h>
int main()
{
   int a;
   auto int input()
   {
      int z,c;
      printf("enter the value of N :");
      scanf("%d",&c);
      int proc(int n)
      {
         int o,b=0;
         for (o=1;n!=0;o++)
         {
            b=b+n%10;
            b=b*10;
            n=n/10;
         }
         b=b/10;
```

```
        printf(" the reverce of N is : %d",b);
      }
      proc(c);
   }
   input();
   return 0;
}
```

```
enter the value of N :76548
 the reverce of N is : 84567
```

g.

i. to Print Fibonacci Series

```
#include<stdio.h>
int fibo(int);
int main()
{
  int count, c = 0, i;
  printf("Enter number of terms:");
  scanf("%d",&count);
  printf("\nFibonacci series:\n");
  for ( i = 1 ; i <= count ; i++ )
  {
    printf("%d\n", fibo(c));
    c++;
  }

  return 0;
}
```

```
int fibo(int num)

{

  if ( num == 0 )

   return 0;

  else if ( num == 1 )

   return 1;

  else

   return ( fibo(num-1) + fibo(num-2) );

}
```

```
Enter number of terms:4
Fibonacci series:

0

1

1

2
```

ii. to print even or odd numbers in given range

```
#include <stdio.h>

void EvenAndOdd(int stVal, int n);

int main()

{

   int n;

          printf("Print even or odd numbers in a given range :\n");

   printf(" Input the range to print starting from 1 : ");

   scanf("%d", &n);

   printf("\n All even numbers from 1 to %d are : ", n);

   EvenAndOdd(2, n);

   printf("All odd numbers from 1 to %d are : ", n);

   EvenAndOdd(1, n);
```

```
    printf("\n\n");
    return 0;
}
void EvenAndOdd(int stVal, int n)
{
    if(stVal > n)
        return;
    printf("%d  ", stVal);
    EvenAndOdd(stVal+2, n);
}
```

```
Print even or odd numbers in a given range :
 Input the range to print starting from 1 : 10
 All even numbers from 1 to 10 are : 2  4  6  8  10
All odd numbers from 1 to 10 are : 1  3  5  7  9
```

iii. to convert a decimal number to binary

```
#include <stdio.h>
int find(int dnum)
{
    if (dnum == 0)
        return 0;
    else
        return (dnum % 2 + 10 * find(dnum / 2));
}
int main()
{
    int dnum;
    printf("Enter the decimal number:");
```

```
    scanf("%d",&dnum);
    printf("%d", find(dnum));
    return 0;
}
```

```
Enter the decimal number:11

1011
```

h.

i. Reverse the elements of an array

```
#include<stdio.h>
void rvereseArray(int arr[], int start, int end)
{
   int temp;
   while (start < end)
   {
      temp = arr[start];
      arr[start] = arr[end];
      arr[end] = temp;
      start++;
      end--;
   }
}
void printArray(int arr[], int size)
{
  int i;
  for (i=0; i < size; i++)
    printf("%d ", arr[i]);
```

```
  printf("\n");
}
int main()
{
   int arr[] = {1, 2, 3, 4, 5, 6};
   int n = sizeof(arr) / sizeof(arr[0]);
   printArray(arr, n);
   rvereseArray(arr, 0, n-1);
   printf("Reversed array is \n");
   printArray(arr, n);
   return 0;
}
```

```
1 2 3 4 5 6
Reversed array is
6 5 4 3 2 1
```

ii. Find the fourth largest and Third smallest element in an array

iii. Find Mean, Median, Mode, Variance, Standard Deviation, and Range of 'n' elements in an array

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

float mean1(float[],int);
float median1(float[],int);
float mode1(float[],int);
double sd1(float[],int);
int main()
{
```

```
int i,n,choice;
float array[100],mean,median,mode;
double sd;
printf("Enter No of Elements\n");
  scanf("%d",&n);
printf("Enter Elements\n");
for(i=0;i<=n-1;i++)
   scanf("%f",&array[i]);
do
{
   printf("\n\tEnter Choice\n\t1.Mean\n\t2.Median\n\t3.Mode\n\t4.Standard
deviation\n\t5.Exit\n");
   scanf("%d",&choice);

   switch(choice)
   {
      case 1: mean=mean1(array,n);
         printf("\n\tMean = %f\n",mean);
         break;
      case 2: median=median1(array,n);
         printf("\n\tMedian = \n",median);
         break;
      case 3: mode=mode1(array,n);
         printf("\n\tMode = %f\n",mode);
         break;
      case 4: sd=sd1(array,n);
         printf("\n\tStandard deviation = %f\n",sd);
         break;
```

```
        case 5: break;
        default:printf("Wrong Option");
            break;
    }

}while(choice!=5);
  getchar();

return 0;
}

float mean1(float array[],int n) {
    int i;
    float sum=0;
    for(i=0;i<=n;i++)
    sum=sum+array[i];
return (sum/n);
}
float median1(float array[],int n) {
float temp;
int i,j;
for(i=n-1;i>=0;i--)
    for(j=0;j<=i;j++)
        if(array[j]>=array[j+1])
        {
            temp=array[j];
            array[j]=array[j+1];
            array[j+1]=temp;
```

```
    }

if(n%2==0)
   return (array[n/2]+array[n/2-1])/2;
else
   return array[n/2];
}
float mode1(float array[],int n) {
return (3*median1(array,n)-2*mean1(array,n));
}
double sd1(float array[],int n) {
int j;
double max[100],sum,variance,mean;
mean=mean1(array,n);
sum=0;
for(j=0;j<=n;j++)
{
   max[j]=pow((array[j]-mean),2);
   sum+=max[j];
}
variance=sum/(j-1);
return sqrt(variance);
}
```

```
Enter No of Elements
4
Enter Elements
1
2
```

```
3

4

        Enter Choice

        1.Mean

        2.Median

        3.Mode

        4.Standard deviation

        5.Exit

3
Mode = -
0.500000

        Enter Choice

        1.Mean

        2.Median

        3.Mode

        4.Standard deviation

        5.Exit

5
```

i.

i. Sum of upper triangular and lower triangular elements of mxm array

```
#include <stdio.h>

void sum(int mat[3][3], int r, int c)

{

   int i, j;

   int usum = 0;

   int lsum = 0;

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

      for (j = 0; j < c; j++)
```

```
        {
            if (i <= j)
            {
                usum += mat[i][j];
            }
        }
        printf("Upper sum is %d\n", usum);
        for (i = 0; i < r; i++)
        for (j = 0; j < c; j++)
        {
            if (j <= i)
            {
                lsum += mat[i][j];
            }
        }
    printf("Lower sum is %d", lsum);
}
int main()
{
    int r = 3;
    int c = 3;
    int mat[3][3] = {{ 6, 5, 4 },
                     { 1, 2, 5 },
                     { 7, 9, 7 }};
    sum(mat, r, c);
    return 0;
}
```

```
Upper sum is 29

Lower sum is 32
```

ii. Find the maximum & minimum element in each row and each column of mxm array

```
#include<stdio.h>

const int MAX = 100;

void smallestInRow(int mat[MAX][MAX], int n, int m)

{

   printf(" { ");

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

   {

      int minm = mat[i][0];

      for (int j = 1; j < m; j++)

      {

         if (mat[i][j] < minm)

            minm = mat[i][j];

      }

      printf(minm,", ");

   }

   printf("}");

}

void smallestInCol(int mat[MAX][MAX], int n, int m)

{

   printf(" { ");

   for (int i = 0; i < m; i++)

   {

      int minm = mat[0][i];

      for (int j = 1; j < n; j++)
```

```
        {
            if (mat[j][i] < minm)
                minm = mat[j][i];
        }
        printf( minm,", ");
    }
    printf("}");
}
int main()
{
    int n = 3, m = 3;
    int mat[MAX][MAX] = { { 2, 1, 7 },
                          { 3, 7, 2 },
                          { 5, 4, 9 } };

    printf("Minimum element of each row is ");
    smallestInRow(mat, n, m);
    printf("\nMinimum element of each column is ");
    smallestInCol(mat, n, m);
    return 0;
}
```

iii. Perform matrix multiplication between two mxn array

```
#include <stdio.h>
#define N 4
void multiply(int mat1[][N], int mat2[][N], int res[][N])
{
    int i, j, k;
    for (i = 0; i < N; i++)
```

```
    {
        for (j = 0; j < N; j++)
        {
            res[i][j] = 0;
            for (k = 0; k < N; k++)
                res[i][j] += mat1[i][k] * mat2[k][j];
        }
    }
}
int main()
{
    int mat1[N][N] = { { 1, 1, 1, 1 },
                       { 2, 2, 2, 2 },
                       { 3, 3, 3, 3 },
                       { 4, 4, 4, 4 } };
    int mat2[N][N] = { { 1, 1, 1, 1 },
                       { 2, 2, 2, 2 },
                       { 3, 3, 3, 3 },
                       { 4, 4, 4, 4 } };
    int res[N][N];
    int i, j;
    multiply(mat1, mat2, res);
    printf("Result matrix is \n");
    for (i = 0; i < N; i++)
    {
        for (j = 0; j < N; j++)
            printf("%d ", res[i][j]);
```

```
        printf("\n");
    }
    return 0;
}
```

j.

i. to perform Substring Extraction (With and Without String Handling Functions).

ii. to read a string and prints if it is a palindrome or not.

```
#include <stdio.h>
#include <string.h>
void isPalindrome(char str[])
{
    int l = 0;
    int h = strlen(str) - 1;
    while (h > l)
    {
        if (str[l++] != str[h--])
        {
            printf("%s is Not Palindrome", str);
            return;
        }
    }
    printf("%s is palindrome", str);
}
int main()
{
```

```
    isPalindrome("Gopal Abdul");
    isPalindrome("\n Welcome");
    isPalindrome("\n devil");
    return 0;
}
```

```
Gopal Abdul is Not Palindrome

Welcome is Not Palindrome

devil is Not Palindrome
```

iii. to replace a particular word by word character in a line of text.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
char* replaceWord(const char* s, const char* oldW,const char* newW)
{
    char* result;
    int i, cnt = 0;
    int newWlen = strlen(newW);
    int oldWlen = strlen(oldW);
    for (i = 0; s[i] != '\0'; i++)
    {
        if (strstr(&s[i], oldW) == &s[i])
        {
            cnt++;
            i += oldWlen - 1;
        }
    }
    result = (char*)malloc(i + cnt * (newWlen - oldWlen) + 1);
```

```
    i = 0;
    while (*s)
    {
        if (strstr(s, oldW) == s)
        {
            strcpy(&result[i], newW);
            i += newWlen;
            s += oldWlen;
        }
        else
            result[i++] = *s++;
    }
    result[i] = '\0';
    return result;
}
int main() {
    char str[] = "Welcome";
    char c[] = "To";
    char d[] = "My_World";
    char* result = NULL;
    printf("Old string: %s\n", str);
    result = replaceWord(str, c, d);
    printf("New String: %s\n", result);
    free(result);
    return 0;
}
```

```
Old string: Welcome
New String: Weldome
```