Program 1: Design and Develop a program to solve simple computational problems using arithmetic expressions and use of each operator leading to simulation of a commercial calculator. (No built-in math function)

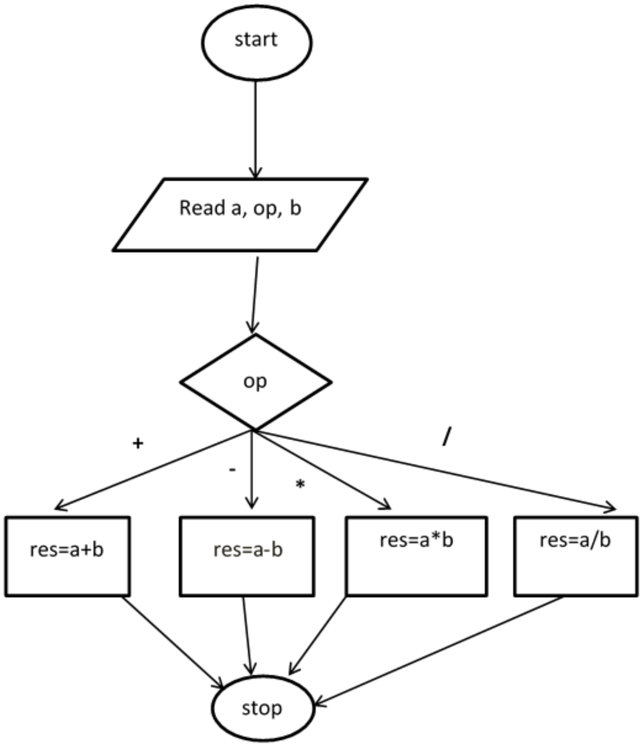
I Programming Approach

This program takes 2 numbers as input and an operator just like expression typed in the calculator, computation is performed based on the operator using case statements and the result is displayed

II Algorithm

```
Input : Two numbers and a character

Output : Result

Step 1: Start the program

Step 2: Read the two numbers and the operator

Step 3: Evaluate option based on the operator  with case statements

        case '+' : res = a + b

        case '-' : res = a -  b

        case '*' : res = a * b

        case '/' : res = a/b

Step 4: if the entered case option is invalid code the print "Wrong Choice"

Step 5: Print the res

Step 6: Stop the program
```

III Flowchart



(IV) Program

```
#include<stdio.h>

#include<conio.h>

void main()

{      float a,b,res;

       char op;

       printf("Enter The Expression in form of op1 operator op2\n");

       scanf("%f%c%f",&a,&op,&b);

       switch(op)
```

```
    {   case '+':        res = a + b;      break;

        case '-‘:        res = a-b;         break;

        case '*‘:        res = a*b;         break;

        case '/':         if(b!=0)         res=a/b;

                          else          printf("Divide by zero error");

                           break;

          default: printf("Illegal operator\n");        break;

    }

 printf("Result is……\n");

 printf("%f%c%f=%f",a,op,b,res);

 getch();

}
```

(V) Output :

```
Enter The Expression in form of op1 operator op2 :
4+5
Result is …….
4+5=9
```

(VI) Viva Questions

1. What is the use of Break statement ?
2. When is switch statement used ?
3. Can multiple constant be used in case statements ?

Program 2: Design and develop a C program that accepts three coefficients (a, b, and c) of a Quadratic equation (ax^2+bx+c=0) as input and compute all possible roots and print the possible roots for a given set of coefficients. Also print the message in case of Zero valued coefficient/s.

(I) Mathematical Approach

1. Quadratic equation $\mathbf{ax^2 + bx + c = 0}$, where a, b, and c are coefficients.
2. The roots are :

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}.$$

3. if discriminate

   is equal to 0, the roots are real and equal

   is greater than 0, the roots are real and distinct

   is less than 0, the roots are complex

(II) Algorithm

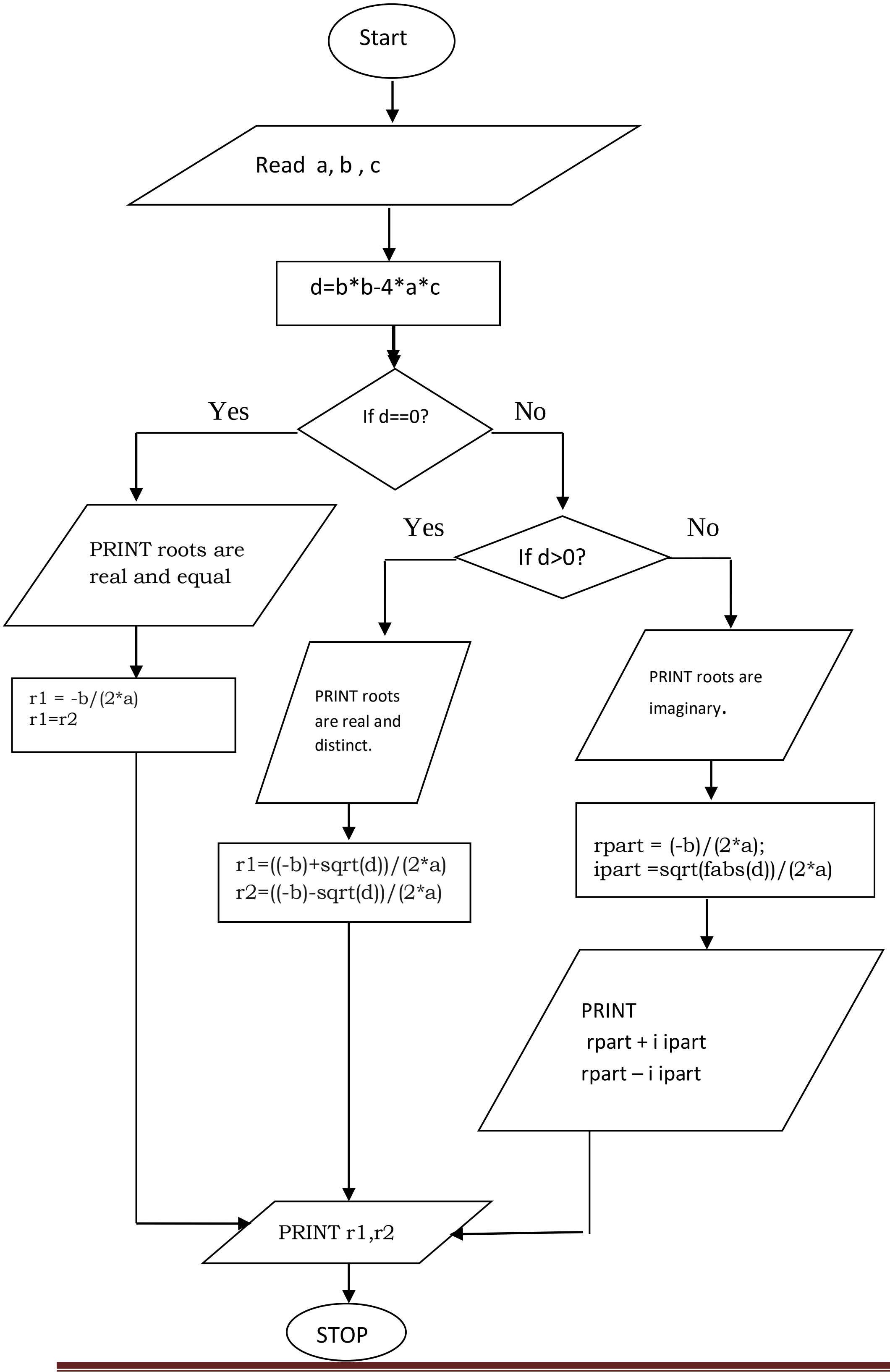
Input: Co-efficients a,b,c

Output: Root1 and Root2

```
Step 1:  Start
Step 2:  Read the coefficients a, b and c.
Step 3: Find d= (b*b)- (4*a*c)
Step 4: if (d=0)
            PRINT Roots are real and equal.
            Else got to Step6
Step 5: find r1= -b/(2*a)
            r2=r1
            PRINT r1,r2
Step 6: if (d>0)
```

```
            PRINT Roots are real and distinct.
            Else go to Step 8
Step 7: find r1= ((-b)+ sqrt(d))/(2*a)
                 R2= ((-b)- sqrt(d))/(2*a)
             PRINT  r1,r2
Step 8: PRINT Roots are imaginary.
Step 9: find rpart= -b/(2*a)
             Ipart= sqrt(fabs(d))/(2*a)
            PRINT r1= rpart + i ipart
                  r2= rpart - i ipart
Step 10: STOP
```

(III) Flowchart



(IV) **Program**

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
        float a,b,c,r1,r2,d,rpart,ipart;
        clrscr();
        printf(“Enter the non zero coefficient values of a, b and c: ”);
        scanf(“%d %d %d”,&a,&b,&c);
                d=(b*b)-(4*a*c);
         if(d==0)
        {
                printf(“\nThe roots are real and equal”);
                r1=-b/(2*a);
                 r2=r1;
                printf(“\nRoot1=%f\tRoot2=%f”,r1,r2);
        }
        else if(d>0)
        {
                printf(“\nThe roots are real and distinct”);
                r1=((-b)+sqrt(d))/(2*a);
                r2=(-b-sqrt(d))/(2*a);
                 printf(“Root1=%f\nRoot2=%f”,r1,r2);
        }
        else
         {
                 printf(“The roots are imaginary\n”);
                rpart=-b/(2*a);
                ipart=sqrt(fabs(d))/(2*a);
                printf(“Root1=%f+i%f\n”,rpart,ipart);
                printf(“Root2=%f-i%f\n”,rpart,ipart);
        }

getch();
}
```

(V) Output:

...

Run 1:

```
Enter the coefficient values of a, b and c: 1 2 1
 The roots are real and equal
 root1=-1.000 root2=-1.0000
```

Run 2:

```
Enter the coefficient values of a, b and c: 1 - 5 6
Roots are real and distinct
root1=3.0000 root2=2.0000
```

(VI) Viva Questions:

1. Write quadratic equation?
2. What is discriminate?
3. Why do you need getch(), clrscr() and exit() functions?
4. Can you enter coefficients in the same line or multiple lines?
5. What is the use for “\n” in printf?

Program 3: An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300 units Rs. 1 per unit. All users are charged a minimum of Rs. 100 as meter charge. If the total amount is more than 400, then an additional surcharge of 15% of total amount is charged. Write a program to read the name of the user, number of units consumed and print out the charges.

Programming Approach: To use multiway decision making statements to calculate charges based on different category of charges based on units consumed.

(II) Algorithm

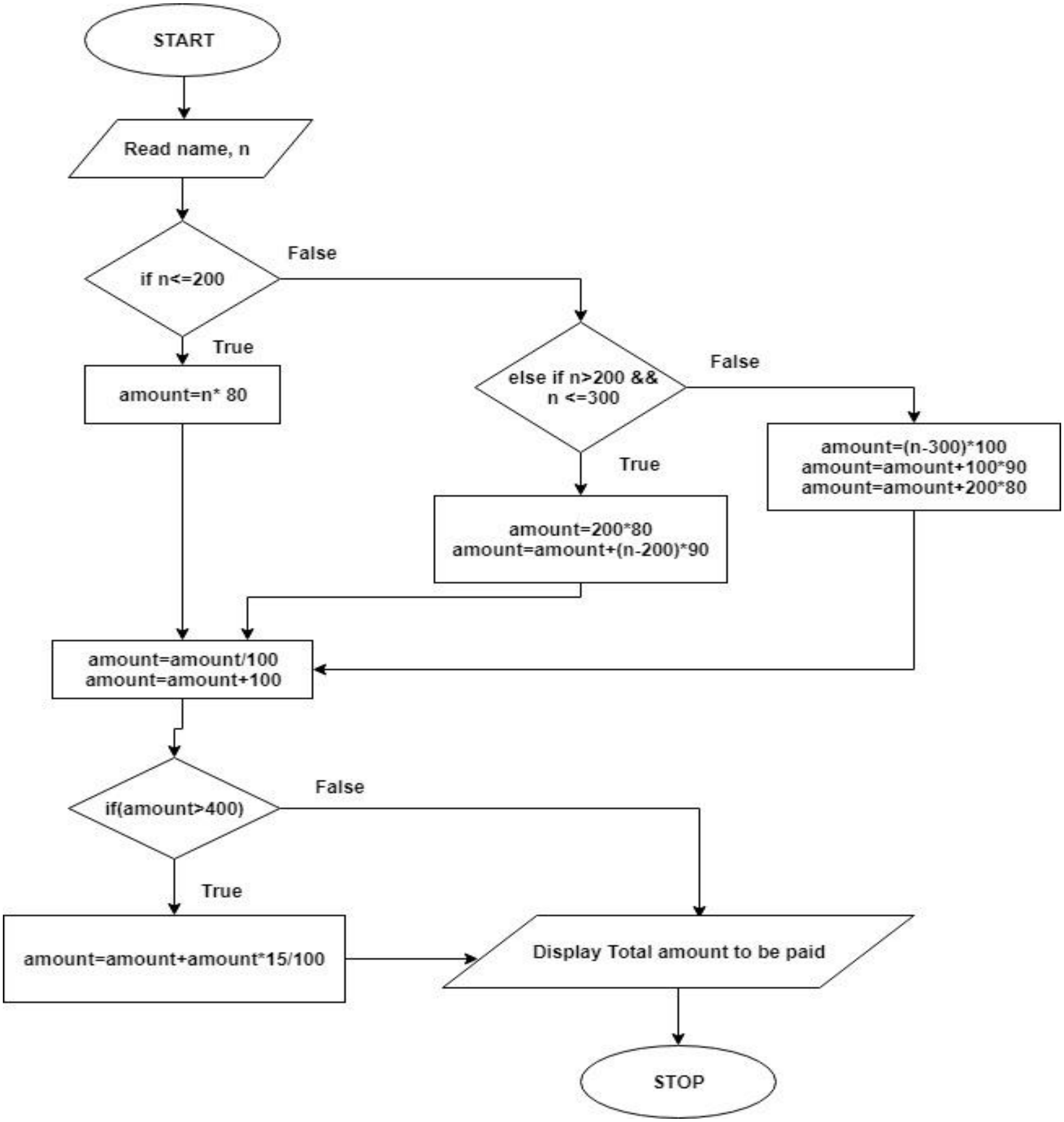
Input: Customer name & Units consumed

Output: Total amount charged for electricity consumption

```
Step 1: [Start]
        Begin
Step 2: [Input customer name]
        Read ←name
Step 3: [Input unit consumed]
        Read n
Step 4: [Check units consumed to calculate the amount]
        if n < = 200
              amount  ← n*80
        otherwise check if n > 200 and n <= 300 then calculate
              amount ← 200 * 80
              amount ← amount +(n-200) *90
        otherwise calculate
              amount ←(n-300)*100
              amount ← amount+100*90
              amount ← amount+200*80
        end if
Step 5:[Calculate the amount]
        amount ← amount/100
        amount ← amount+100
Step 6:  [Check if amount is greater than 400 then calculate additional charge of 15%]
          if amount > 400 then
```

```
                    calculate amount← amount + amount * 15/100
               end if
Step 7: [Print total amount to be paid by the customer]
             Display  amount
Step 8: [Finished]
               Stop
```

(III) Flowchart



(IV) Program

```
#include<stdio.h>

void main()
{
        char name[20];
        int n;
        float amount;

        printf("Enter the consumer name\n");
        scanf("%s", name);
        printf("Enter no. of units consumed \n");
        scanf("%d",&n);

        if (n<=200)
        {
                amount=n*80;
        }
        else if(n>200 && n<=300)
        {
                amount=200*80;
                amount=amount+(n-200)*90;
        }
        else
        {
                amount=(n-300)*100;
                amount=amount+100*90;
                amount=amount+200*80;
        }

        amount=amount/100;          //To convert into Rupees
        amount=amount+100;          // Additional 100 Rupees to be added

        if(amount>400)
        {
                amount=amount+amount*15/100;
        }

        printf("Total amount to be paid is %.2f Rs\n",amount);
}
```

(V) Output

```
Enter the consumer name :
Bhagya
Enter no. of units consumed
275
Total amount to be paid is = 327.50
```

(VI) Viva Questions

1. What is the advantages of elseif ladder ?
2. What is the difference between elseif and switch ?
3. What is explicit type casting ?

Program 4: Design and develop a C program to find the reverse of an integer number NUM and check whether it is PALINDROME or NOT. Implement a C program for the developed algorithm that takes an integer number as input and output the reverse of the same with suitable messages. Ex: Num: 1234, Reverse: 4321, Not a Palindrome.

(I) <u>Programming approach</u>

A number is a palindrome if the reverse of that number is equal to the original number. To check whether a number is palindrome or not first we reverse it and then compare the number obtained with the original; if both are same then number is palindrome otherwise not.

(II) <u>Algorithm</u>

<u>Algorithm:</u> To find whether a number is palindrome or not.

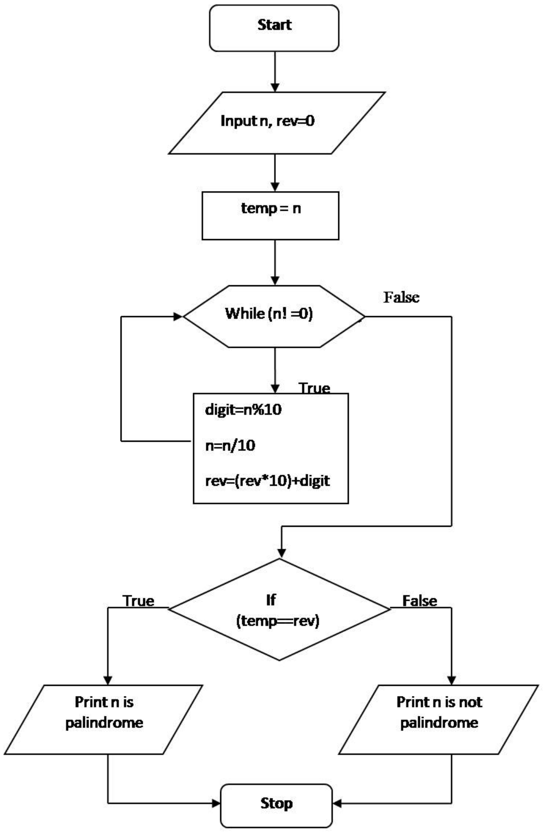
<u>Input</u>: A number.

<u>Output</u>: Reverse of the number. Also check whether the given number is palindrome or not.

```
Step1: [Initialization]
       Start
Step2: [Input]
       Read n
Step3: [Set number n to a variable temp]
```

```
          temp←n
Step4: [Iterate until n not equal to zero]
                    while (n!=0) do
                              digit←n%10
                              rev=digit+10*rev
                              n=n/10
Step5: [Print reverse number]
                    Print rev
Step6: [Check if original number and reverse number is same. If it is, number is palindrome. Otherwise, number is not palindrome]
          if (rev==temp)
                    Print palindrome
          else
                    Print not palindrome

Step7: [Finished]
          End
```

(III) Flowchart



(IV) Program

```
#include<stdio.h>
#include<conio.h>
void main()
function*/
{
        int n, rev=0,temp,digit;
        printf("Enter an integer number\n");
        scanf("%d",&n);
        temp=n;
        while(n!=0)
        {
                digit=n%10;
                rev=rev*10+digit;
                n=n/10;
        }
        printf(“\n Given number is :%d”,temp);
        printf(“\n Reverse number is:%d”,rev);
        if(rev==temp)
                printf("The number is a palindrome\n");
        else
                printf("The number  is not a palindrome\n");
}
```

(V) Output

Run1:

```
Enter an integer number
3443
Given number is: 3443
It’s reverse is 3443
The number is a palindrome
```

Run 2:

```
Enter an integer number
5678
Given number is: 5678
It’s reverse is 8765
The number is not a palindrome
```

(VI) Viva Questions

1. Define palindrome.
2. What are the applications of palindrome?
3. What is printf() and scanf()?
4. Give examples of palindrome.

Program 5. Design and Develop a C program to compute sin(x) using Taylor series approximation and compare with built-in library function and display with appropriate messages.

(I) Mathematical Approach

A Taylor Series is an expansion of a function into an infinite sum of terms.

$$\sin x \quad = \quad x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} - \dots$$

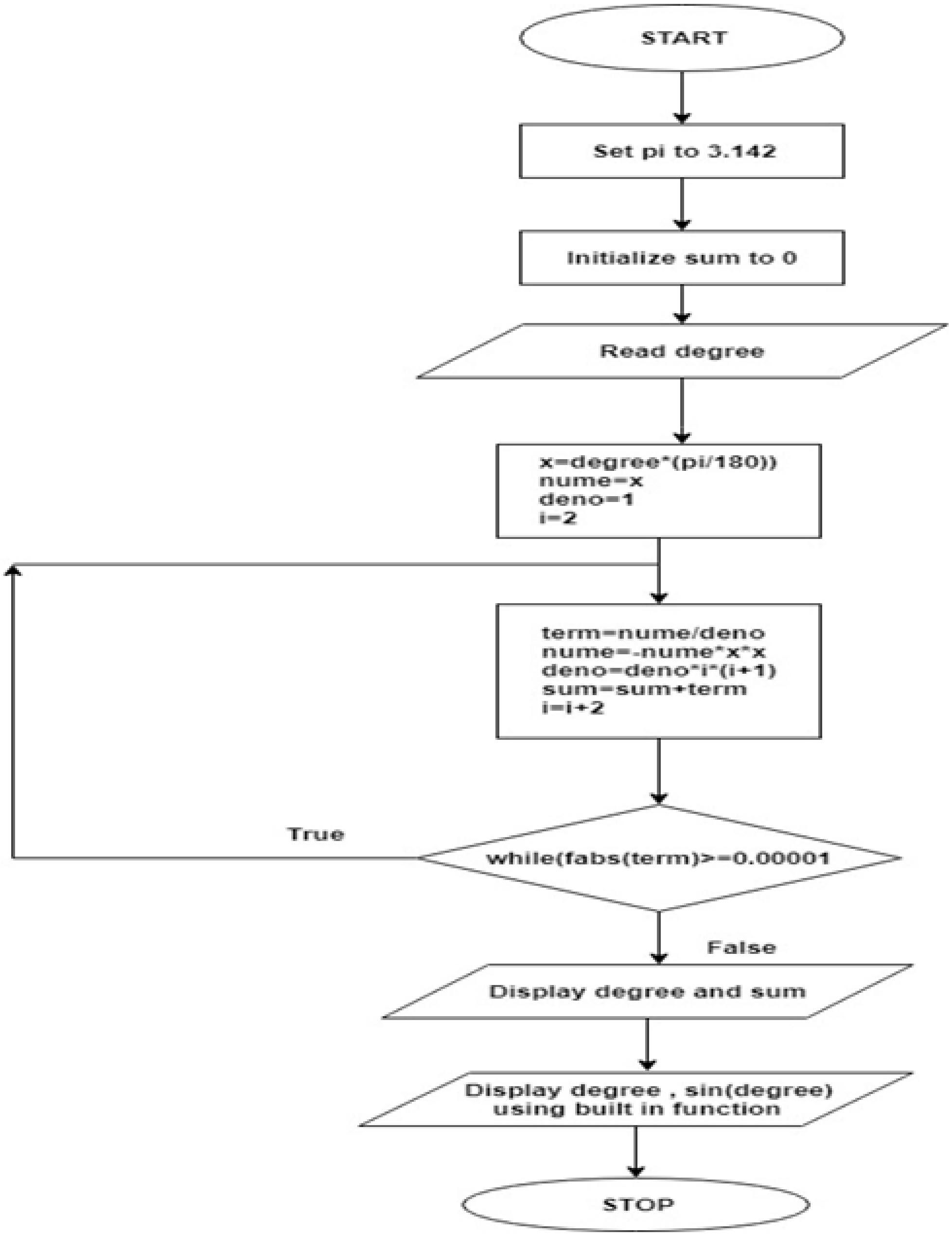
(II) Algorithm:

Input: Integer number degree

Output: Sin(x), where x is the value of degree in radian

```
Step 1: Start
Step 2: Set pi to 3.142, sum =0
Step 3:  Read degree x
Step 4: [Convert degree to radian]
            x = degree*(pi/180)
Step 6: [Assign x value to numerator]
             nume =  x
Step 7: [Set denominator to 1]
           deno = 1
Step 8: [Set i to 2]
           i =2
Step 9: [Repeat until true]
            term = nume/deno
            nume = -nume *x*x
            deno =deno * i * (i+1)
            sum =sum+term
             i = i+2
            do until while (fabs(term)>=0.00001)
Step 10: [Print sin of degree]
             Display sum
Step 11: [Print sin(degree) using built-in function for comparison]
             Display sin(x)
Step 12: [Finished]
              Stop
```

(III) Flowchart



Program :

```
#include <math.h>
#define pi 3.142
```

```
void main()
{
        int i,degree;
        float x,sum=0,term,nume,deno;
        printf("\n Enter the value of x in degree : ");
        scanf("%d", &degree);
        x=degree*(pi/180);
        nume = x ;      deno = 1;       i = 2;
                do
                {       term=nume/deno;
                        nume=-nume*x*x;
                        deno=deno*i*(i+1);
                        sum=sum+term;
                        i=i+2;
                }while(fabs(term)>=0.00001);
        printf("\n The sine value =  %f",sum);
        printf("\n The sine value using inbuilt function  = %f",sin(x));
        getch();
}
```

(V) Output

```
Enter the value x in degrees: 45
The sine value is 0.706825
The sine value using built-in function is 0.706825
```

(VI) Viva Questions

1. What is define constant ?
2. What is the difference between while and do-while ?
3. What is taylor series for sin ?

Program 6. Design and develop a C program that read N integer numbers and arrange them in ascending order using Bubble Sort.
Answer:

(I) Mathematical approach

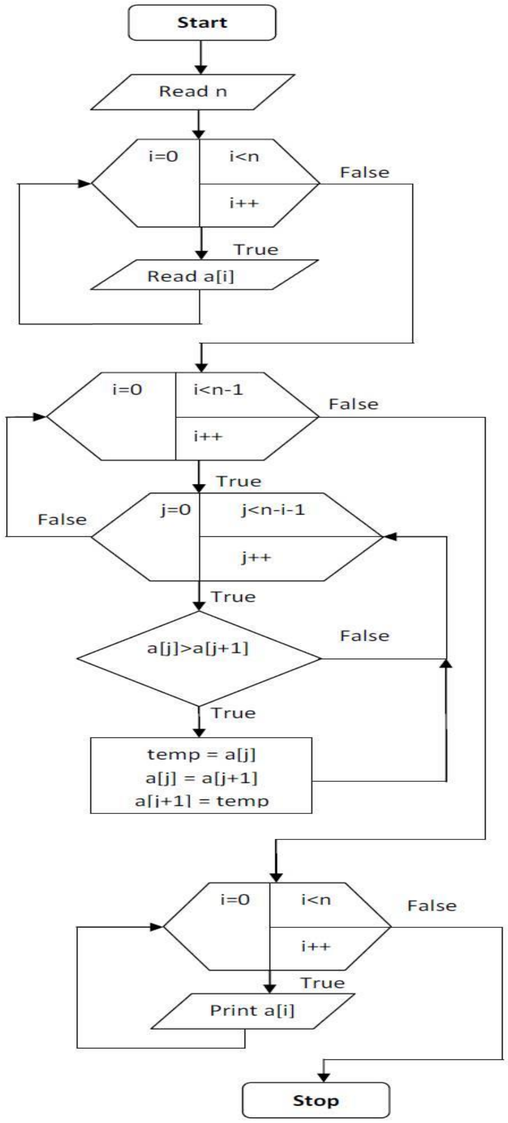
Bubble sort algorithm takes the array of numbers as input and compares the adjacent numbers and performs swapping of numbers which are not in order and prints the array of numbers in ascending order.

(II) Algorithm

Algorithm Bubble Sort [This algorithm takes a list of unsorted numbers and arranges them in ascending order using Bubble Sort Method].

```
Step 1: [Initialize]
            Start
Step 2: [Input]
            Read n
Step 3: [Input Unsorted array]
            Read elements to array a[]
Step 4: Print elements of array a[]
Step 5: [Iterate array a[] in two loops. Outer loop gives number of pass. Inner loop does
        Swap task. In each pass, compare each pair of adjacent items. If formal
        elements are greater than latter one, swap them.]
            for each value I in array a[i] to n do

                for each value j in array a[j] to n-1 do

                    [Compare each pair of adjacent elements]

                        If (a[j]>a[j+1]) then

                        [Swap these elements using temp variable]

                            temp ← a[j]

                            a[j] ← a[j+1]

                                 ←
                        a[j+1          temp
                    End if
                End for
              End for
Step 6:  Print array with sorted elements
Step 7:  [Finished]
             End.
```

(III) Flow Chart



(IV) Program

```
#include<stdio.h>
void main()
{
   int a[100], n, i , j, temp;
  clrscr();
   printf("Enter the number of elements\n");
   scanf("%d",&n);

   printf("Enter the %d elements of array\n",n);
            for(i=0;i<n;i++)
            {
            scanf("%d",&a[i]);
            }

   for(i=0;i<n-1;i++)
   {        for(j=0;j<n-i-1;j++)
         {
               if(a[j]>a[j+1])              /*Compares each pair of adjacent elements*/
                     {
                        temp=a[j];               /*Swap the elements using temp variable*/
                        a[j]=a[j+1];
                        a[j+1]=temp;
                     }
           }
   }

   printf("The sorted array is\n");          /*Prints the sorted elements*/
   for(i=0;i<n;i++)
   printf("%d\n",a[i]);
   getch();
}
```

(V) Output

```
Enter the number of elements
4
Enter 4 elements of array
87 100 20 3
The input array is
87
100
20
3
The sorted array is
3
20
87
100
```

(VI) Viva questions

1. What is bubble sort?
2. What is an array?
3. How to declare an array variable?
4. What is prinft() and scanf() function?
5. What is the application of bubble sort?

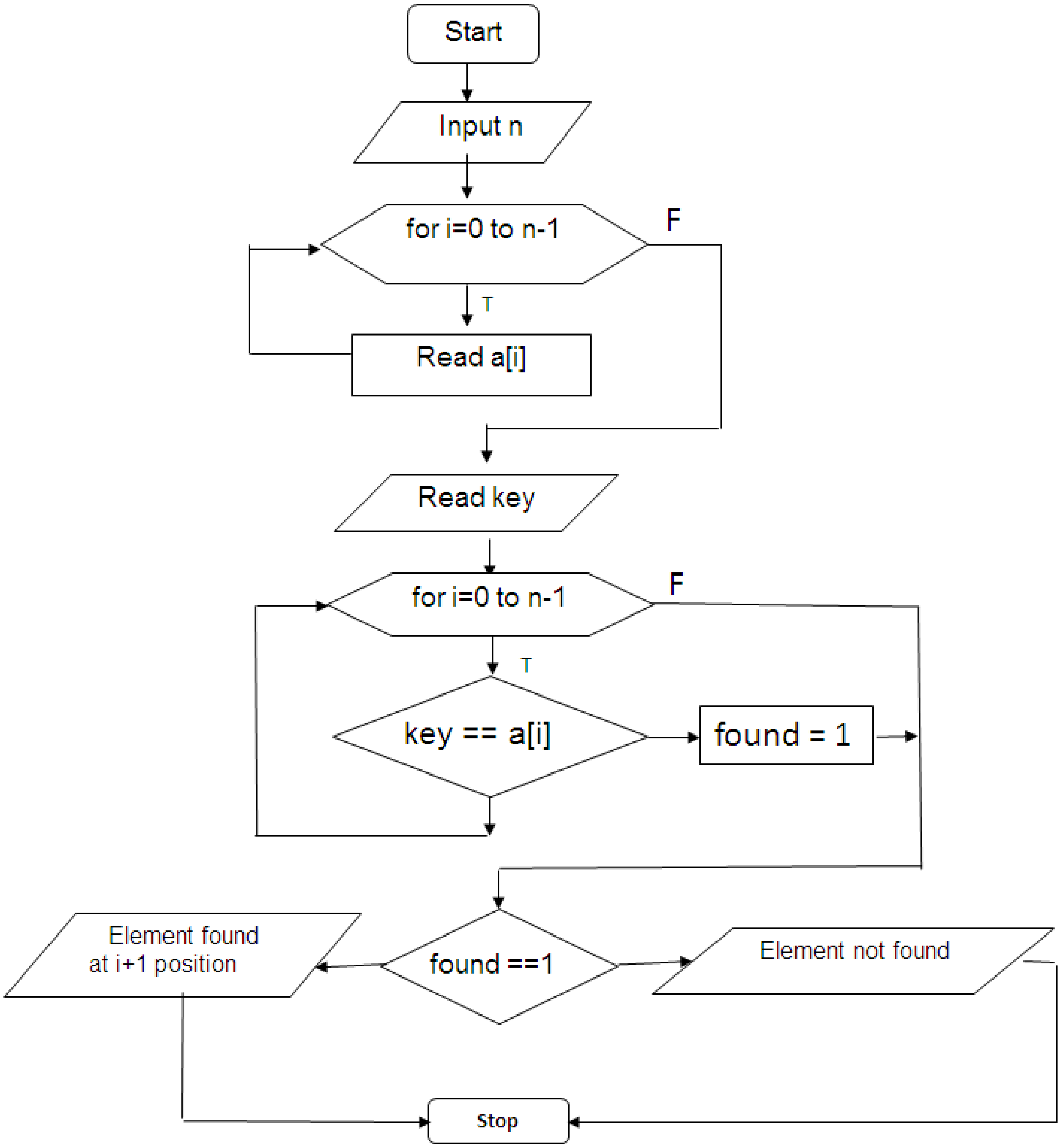
Program 7: Design and Develop a C program to input N numbers and store them in an array and perform a linear search for a given key and report success or failure.

(I) Programming approach

(II) Algorithm

```
Step 1: [input the number of items]
        read n
Step 2: [input N elements from keyboard]
        for i=0 to n-1
        read a[i]
        end for
Step 3: [input the item to be searched]
        read key
Step 4: [ search the key]
        for i=0 to n-1
        if( key == a[i]) found =1
        end for
Step 5: [ Display the position ]
        If ( found =1)
            Print "Element found at position i+1"
        Else
         print "element not found"
Step 6: Stop
```

(III) Flowchart



(IV) Program

```
#include <stdio.h>
#include<conio.h>
void main()
{
   int a[10];
   int i, n, key, found = 0;
```

```
    printf("\nEnter the no of elements:");
    scanf("%d", &n);

    printf("Enter the elements  \n");
    for (i = 0; i < n; i++)
    {
           scanf("%d", &a[i]);
   }

    printf("Enter the element to be searched \n");
    scanf("%d", &key);

    /*  Linear search begins */
    for (i = 0; i < n ; i++)
    {
       if (a[i] == key )
       {
          found = 1;
          break;
       }
    }

    if (found == 1)
       printf("Element is present in the array at %d position\n", i+1);
    else
       printf("Element is not present in the array\n");

getch();
}
```

(V) Output

Run1

```
Enter the no of elements
5
Enter the elements
45 78 23 90 12
Enter the element to be searched
90
```

```
Element is present in the array at 5 position
```

Run2

```
Enter the no of elements
7
Enter the elements
7 9 3 5 6 2 4
Enter the element to be searched
8
Element is not present in the array
```

(VI) Viva Quesions

1. What is the staring index value of an array ?
2. What are the different ways of initialization of array ?
3. What is the advantages of arrays ?

Program 8: Design and Develop a C program that reads N integer numbers and search a key element using *Binary search* technique.

Answer:

(I) Mathematical approach

Binary Search starts with the middle element of the sorted list.

1. [low, high] denotes the range in which element has to be present and [mid] denotes the middle element.
2. Initially low = 0, high = number_of_elements and mid = floor((low+high)/2).
3. If the middle element of the list is equal to the ‘key’ then we have found the position the specified value.
4. Else if the ‘key’ is greater than the middle element then the ‘key’ has to be present in the last half of the list.
5. Or if the ‘key’ is lesser than the middle element then the ‘input key’ has to be present in the first half of the list.
6. Hence, the search list gets reduced by half after each iteration.

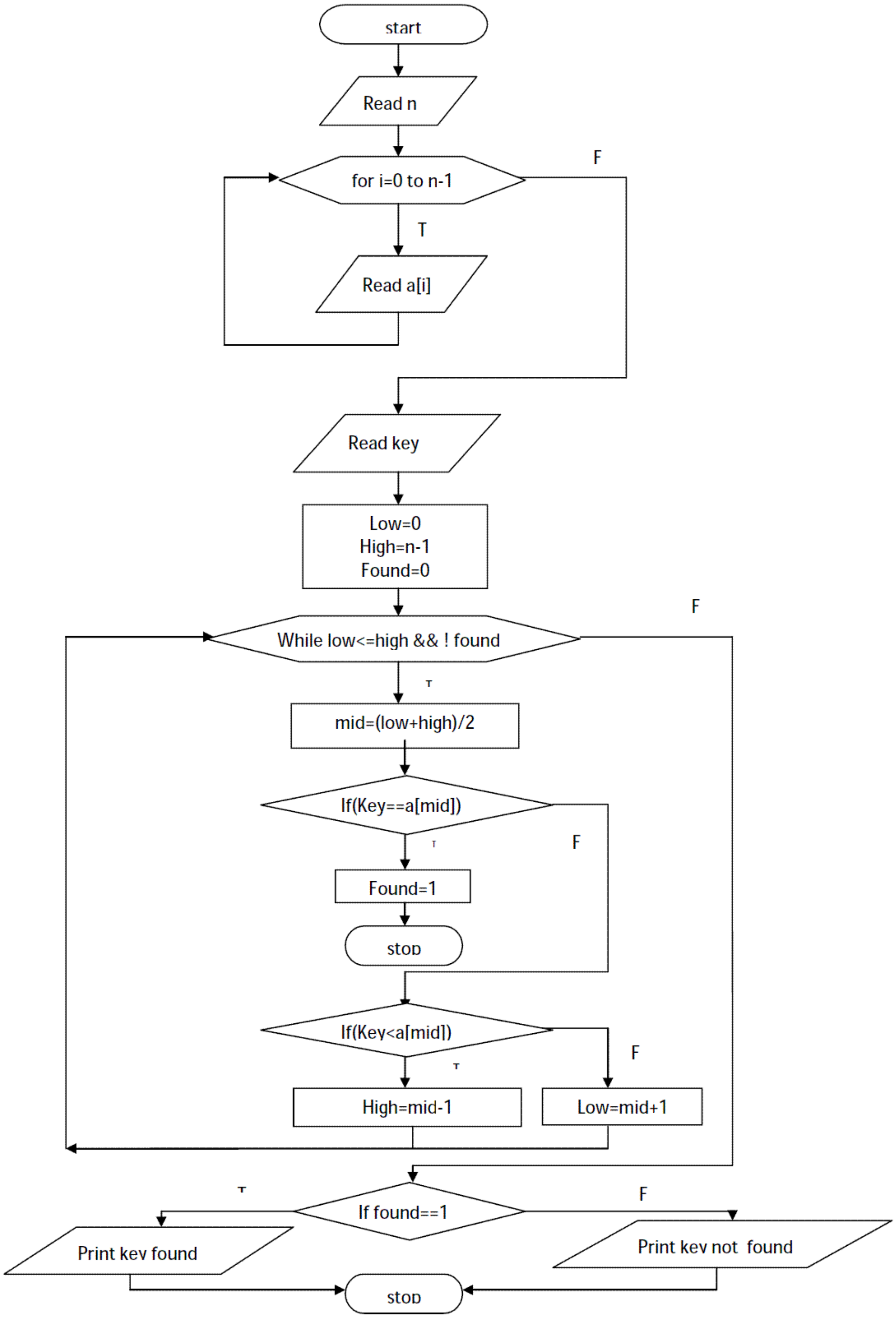
(II) Algorithm

```
Step 1: [input the number of items]
        read n
Step 2: [input N elements from keyboard]
        for i=0 to n
        read a[i]
        end for
Step 3: [input the item to be searched]
        read key
Step 4: [initialization]
        low=0
        high=n-1
        found=0
Step 5: [search using binary search method]
        while (low <=high && !found)
        mid=(low+high)/2
        if(key==a[mid])
        found=1
        else if (key<a[mid])
        high=mid-1
        else
        low=mid+1
        end while
Step 6: if(found==1)
        Print "key found at mid+1 position"
        Else
```

Print “key not found”

Step 7: Stop

(III) Flowchart



(IV) Program

```
#include<stdio.h>
#include<conio.h>
void main()
{
        int a[100], n, i, key, low, mid, high, found=0;
        clrscr();
        printf(“enter the number of elements\n”);
        scanf(“%d”,&n);
        printf(“enter %d elements in ascending order\n”,n);
        for(i=0;i<n;i++)
        scanf(“%d”,&a[i]);                  //read n array elements in ascending order
        printf(“enter an element to search\n”);
        scanf(“%d”,&key);                               //read key element to search
        low=0;
        high=n-1;                                       //initialize range of elements
        while(low<=high && !found)
        {
                mid= (low+high)/2;                      //compute mid element position
                if(key==a[mid])
                found=1;                                        //key found at mid position
                else if (key<a[mid])        // If the middle element is greater than key then key
                                     has to be present in the range [low,mid-1]
                high=mid-1;
                else                        //If the middle element (mid) is less than key then key
                                     has to present in range [mid+1 , high],
                low=mid+1;
        }
        if(found==1)
        printf(“key found at position %d\n”,mid+1);
        else
        printf(“key not found\n”);
        getch();
}
```

(V) Output

Run 1:

```
Enter the number of elements:
6
Enter 6 elements in ascending order
4 7 9 11 15 18
Enter an element to search
15
Key found at position 5
```

Run 2:

```
Enter the number of elements:
4
Enter 4 elements in ascending order
2 5 7 9
Enter an element t search
10
Key not found
```

(VI) Viva questions

1. Explain the logic of binary search.
2. What is the difference between linear search and binary search?
3. Will Binary search works for non ascending array also? If no why?

Program 9: Design and develop a C program that reads two matrices A (m x n) and B (p x q) and Compute product of matrices A and B. Read matrix A and matrix B in row major order. Print both the input matrices and resultant matrix appropriately.

Answer:

(I) Programming Approach

1. Read matrix A of size mxn
2. Read matrix B of size pxq
3. Check for compatibility (n==p)
4. Perform multiplication AXB, resultant matrix size will be mXq

   Where m & p are the number of rows

   n & q are the number of columns

(II) Algorithm

Step 1: Start

Step 2: Declare variables and initialize necessary variables

Step 3: Enter the element of matrices by row wise using loops

Step 4: Check the number of rows and column of first and second matrices

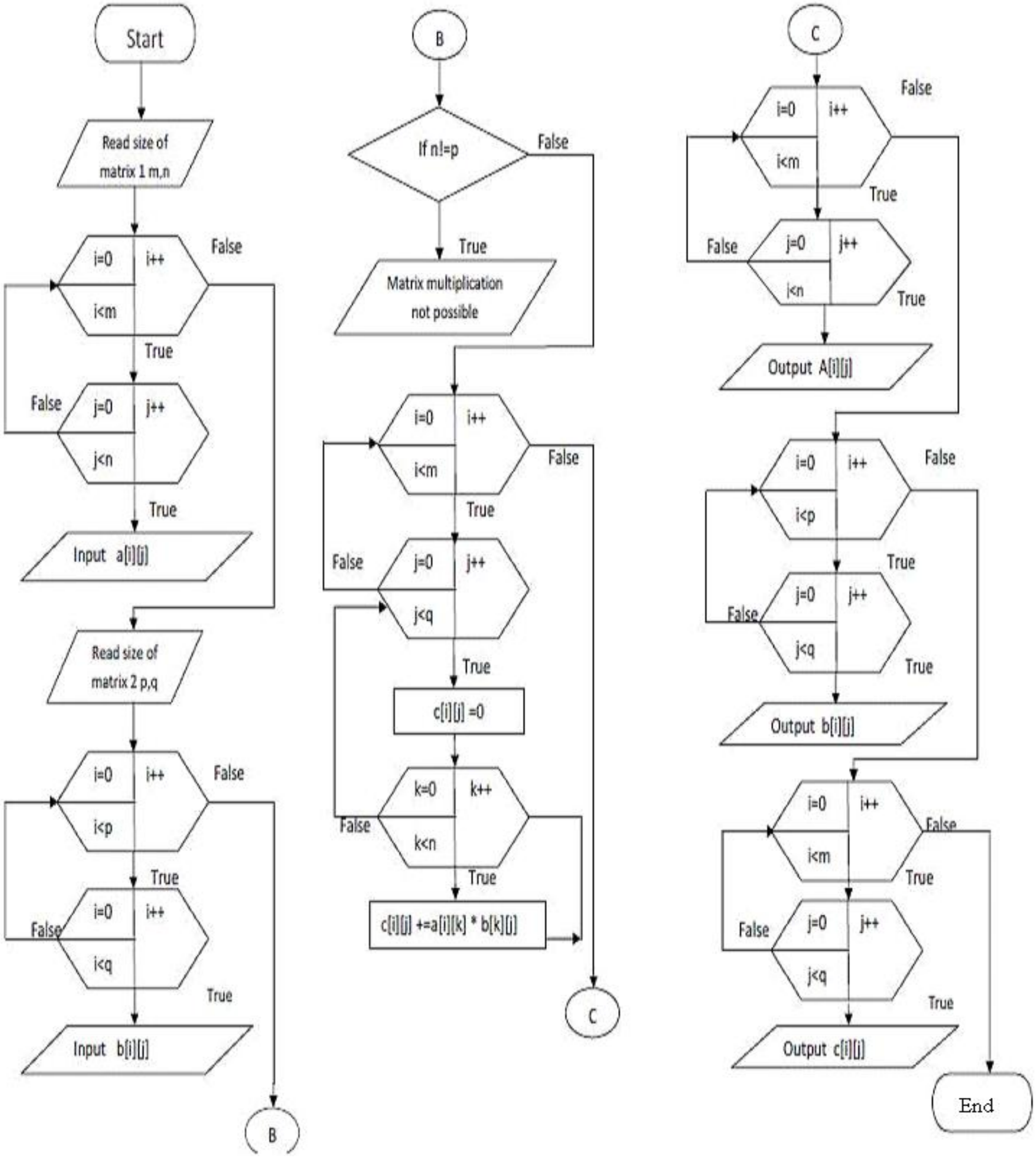
Step 5: If number of rows of first matrix is equal to the number of columns of second matrix, go to step 6. Otherwise, print matrix multiplication is not possible and go to step 8.

Step 6: Multiply the matrices using nested loops.

Step 7: Print the product in matrix form as console output.

Step 8: Stop

(III) Flowchart



(IV) <u>Program</u>

```
#include<stdio.h>
#include<conio.h>
void  main()
{
    int a[10][10],b[10][10],c[10][10],i,j,k,m,n,p,q;
    clrscr();  // clear screen

    printf("enter the size of matrixA");
          scanf("%d%d",&m,&n);// read size of matrix a
    printf("enter the size of matrixB");
          scanf("%d%d",&p,&q); // read size of matrix b

    if(n!=p)// check for compatability
          printf("matrix multiplication is not possible");
    else
    {
          printf("enter elements of matrixA\n");
          for(i=0;i<m;i++)
                for(j=0;j<n;j++)
                      scanf("%d",&a[i][j]);                    //read matrix a



          printf("enter elements of matrixB\n");
          for(i=0;i<p;i++)
             for(j=0;j<q;j++)
                      scanf("%d",&b[i][j]);                //read matrix b
```

```
            for(i=0;i<m;i++)
                for(j=0;j<q;j++)
                 {       c[i][j]=0;
                            for(k=0;k<n;k++)
                             c[i][j] = c[i][j] +   a[i][k] * b[k][j] ;      // multiply a and b matrix
                  }



            printf("\nresultant-matrix is \n");
             for(i=0;i<m;i++)
             {       for(j=0;j<q;j++)
                         printf("%5d",c[i][j]);
                       printf("\n");
              }
          } // end of else
         getch ( );
       }
```

(V) Output

Run 1

```
Enter the size of matrix a
2  2

Enter the size of matrix b
2  3

Enter the elements of matrix a
2  4
1  3

Enter the elements of matrix b
4  1
2  3
```

```
The resultant matrix c
16  14
10  10
```

Run 2

```
Enter the size of matrix a
3  2

Enter the size of matrix b
4  3

matrix multiplication is not possible
```

(VI) Viva questions

1. What is an Array?
2. What is 2D-Array?
3. How to declare 2D-Array?
4. How to initialize 2D-Array?
5. What is disadvantage of an array?
6. What is the output of the following code?

```
int main( )
{
  int a[3][3]={10,20,30,1,2,3,21,22,23};
  printf("%d",a[2][2]);
}
```

7. Differentiate between an array and an ordinary variable.

Program 10: Design and develop a C program to implement the following operations without using library functions. Display the results after every operation.

a. **Read STRING S1 = "Dayananda"**
b. **Read STRING S2 = "Sagar"**
c. **Output the concatenated string (s1+s2) STIRNG s3 = "Dayananda Sagar"**

Answer:

(I) Programming Approach

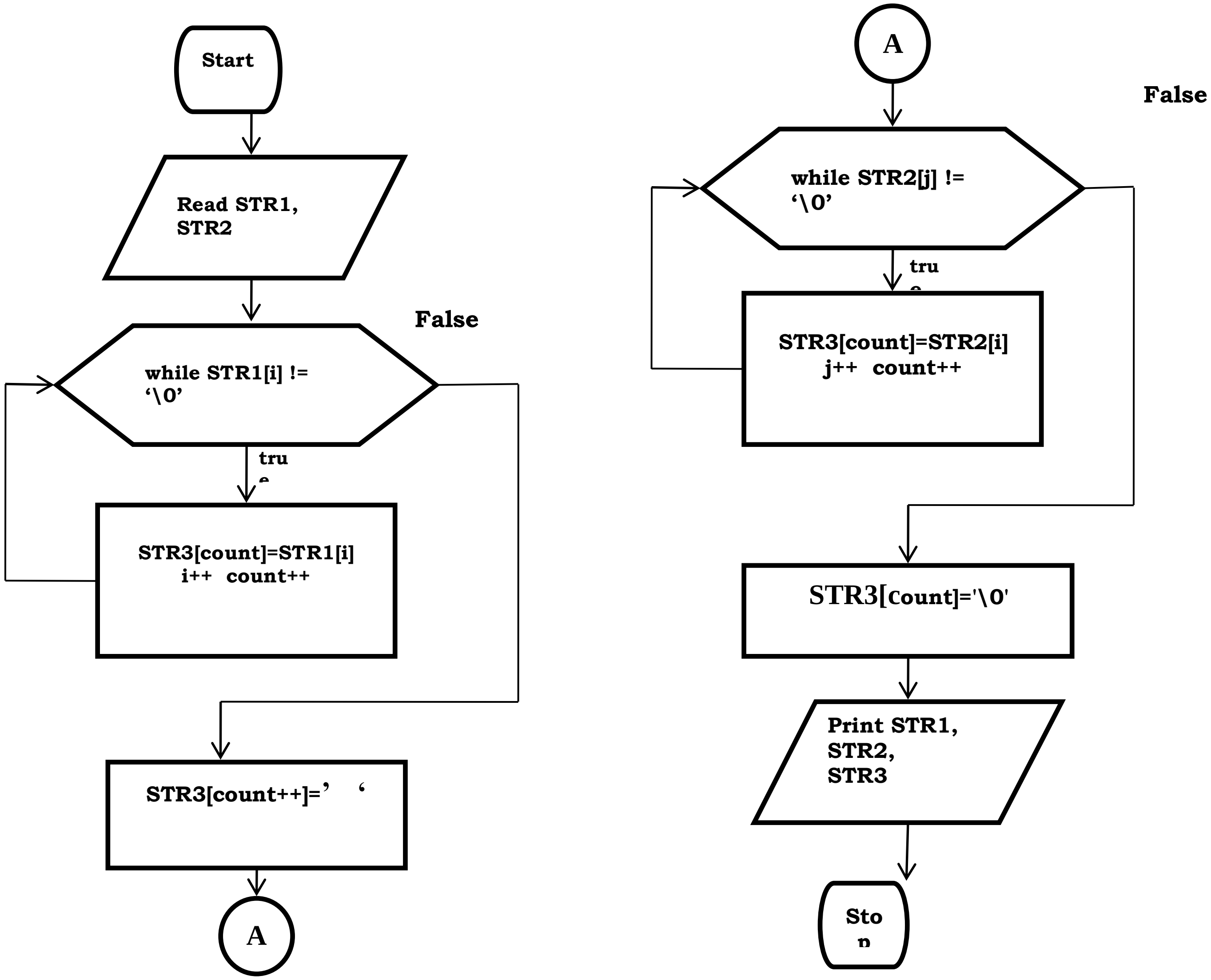
The program is used to read 2 strings STR1 & STR2 and compute the resulting string STR3 which consists of STR1 and STR2 without using library function. The program includes the concept of Strings and looping.

(II) Algorithm

```
Step 1: START
        [Input 2 strings STR1 & STR2]
        Read STR1, STR2
Step 2: [Initialize i, j and count to Zero]
        i = 0;   j = 0;     count = 0;
Step 3: [Copy the string STR1 to STR3]
        while STR1[ i ] != '\0'
            BEGIN
                STR3[ count ]   =  STR1[ i ]
                count ++        [Track length of string STR3]
                i++;
            END
Step 4: [Increment the value of Count to insert space between 2 strings]
        STR3[count++]=' '
Step 5: [Copy the string STR2 to STR3]
        While STR2[ j ] != '\0'
            BEGIN
                STR3[ count ]   =  STR1[ i ]
                count ++
                j++;
            END
Step 6: [Terminate the String with end of Character]
                STR3[count] = '\0'
Step 7: [Print the String STR1, STR2 and STR3]
                Print STR1, STR2, STR3
```

Step 8: END

(III) Flow Chart



(IV) Program

```
/* The program to Concatenate 2 Strings */
#include<stdio.h>
#include<conio.h>
void main()
{
    char STR1[100],STR2[100],STR3[100];
    int i=0,j=0,count=0;
    clrscr( );

    //READING OF STRINGS STR1 AND STR2
    printf("Enter the String 1\n");
    gets(STR1);
    printf("Enter the String 2\n");
    gets(STR2);

    //COPY STRING STR1 TO STR3
    while(STR1[i]!='\0')
    {
        STR3[count]=STR1[i];
        count++;
        i++;
    }

    //INSERT BLANK SPACE BETWEEN STR1 AND STR2
    STR3[count++]=' ';

    //COPY STRING STR2 TO STR3 FROM THE LOCATION COUNT
    while(STR2[j]!='\0')
```

```
        {
           STR3[count]=STR2[j];
           count++;
           j++;
        }

        //END OF STRING
         STR3[count]='\0';
         //PRINTING OF STRINGS STR1, STR2, STR3
         printf("\n String STR1=\t");
         puts(STR1);
         printf("\n String STR2=\t");
         puts(STR2);
         printf("\n String STR3=\t");
         puts(STR3);
         getch();
     }
```

(V) Output

Run1:

```
String S1 =         Dayananda
String S2 =         Sagar
String S3 =         Dayananda Sagar
```

Run2:

```
String S1 =         Engineering
String S2 =         Department
String S3 =         Engineering Department
```

(VI) Viva Questions

1. What is String?
2. Write C code snippet to find the length of a string without using library function?

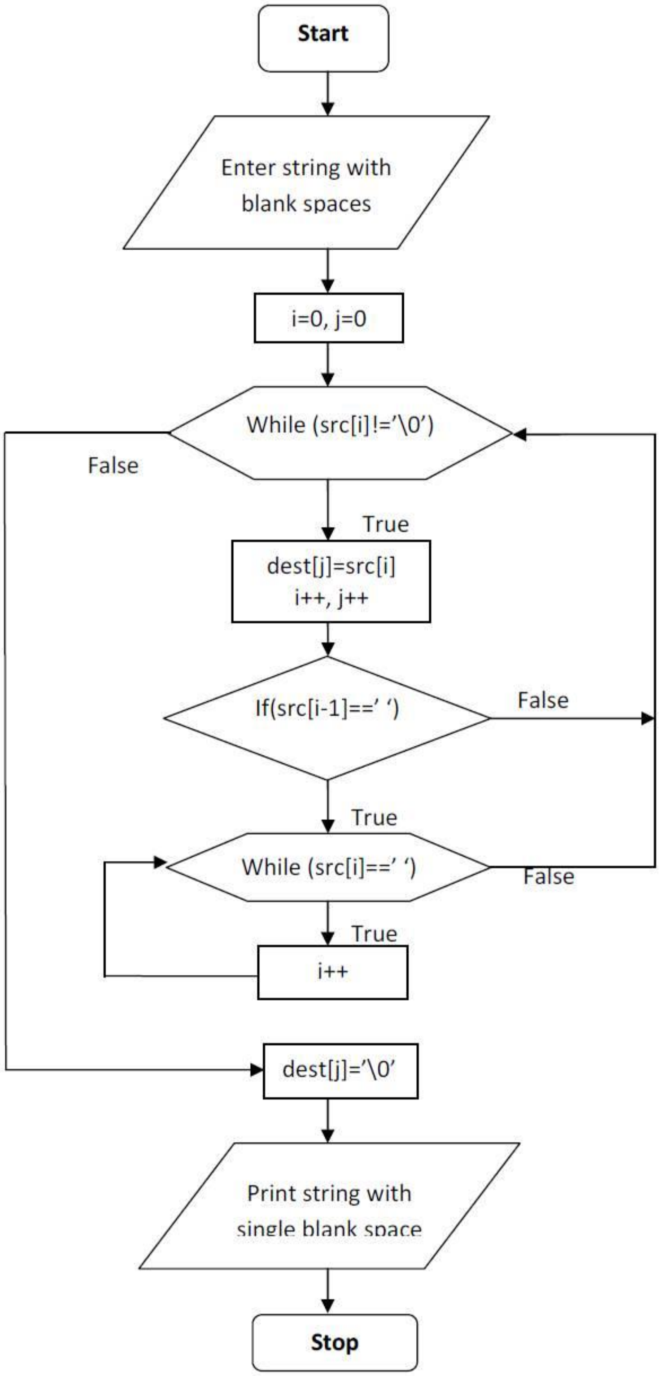
Program 11: Design and Develop a program in C to copy its input to its output, replacing each string of one or more blanks by a single blank.

(I)

(II) Algorithm:

```
Step 1: start
Step 2:  Read source array ‘src’
Step 3: i=0
        J=0
Step 4: while src[i]!=null string
        dest[j]=src[i]
        j++
        i++
                if src[i-1]=’ ‘             /* space */
                        while src[i]=’ ‘
                        i++
                        end while
                end if
        end while
Step 5: dest[j]=null string
Step 6: print “dest”

Step 7: stop
```

(III) Flowchart



(IV) Program:

```
# include<stdio.h>
# include<conio.h>
void main ()
{
char src[100], dest[100];
            int i,j;
            clrscr();
            printf("Enter the string with Blanks\n\n");
            gets(src);
            i=0;                                    //first index of source string
            j=0;                                    //first index of destination string
            while(src[i]!='\0')
            {
                        dest[j]=src[i];              //copy char from src to dest
                        j++;
                        i++;
                     if(src[i-1]==’  ’)
                        {
                                    while(src[i]==’  ’)
                                    i++;
                        }
             }
             dest[j]='\0';
             printf("The string with single blank space \n");
             puts(dest);
           getch();
}
```

8. OUTPUT:

```
Enter the string with Blanks
welcome_ _ _ _ to_ _ _DSCE_DSI

The string with single blank space
welcome_to_DSCE_DSI
```

NOTE: Number of space is represented by no. of _(Underscore)

(VI) Viva Questions

1. What is the output of the code snippet given below

```
void main()
{
char arr[8]=”Computer”
printf(“%s”,arr);
}
```

2. What is the output of the code snippet given below

```
void main()
{
char arr[6]=”Network Programming”
printf(“%s”,arr);
}
```

3. Write a code snippet to compare 2 stings?

4. List out various String handling functions?

5.What is the difference between strings and character arrays ?

Program 12: Develop and develop a C function isprime(num) that accepts an integer argument and returns 1 if the argument is prime, 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.

Answer:

(I) Mathematical Approach

The range between which we have to find the prime number should be integers.

The mathematical approach is:

a. Take each number within the specified range.
b. Perform number % i where i=1 to number and keep track of count where we can store how many numbers completely divide the given number.
c. If count ==2 then the number is prime number otherwise it is not prime number.

(II) Algorithm

Algorithm: *To find the prime numbers between the given range.*

Input: *Integer range to find prime numbers.*

Output: *Prime numbers with in the given range.*

```
Step 1: Start
Step 2: Input Integer Range n1 and n2
Step 3: For i=n1 until n2
Step 4: Call isprime(i);
Step 5: If return value is 1
Step 6: Print the prime number
Step 7: End
```

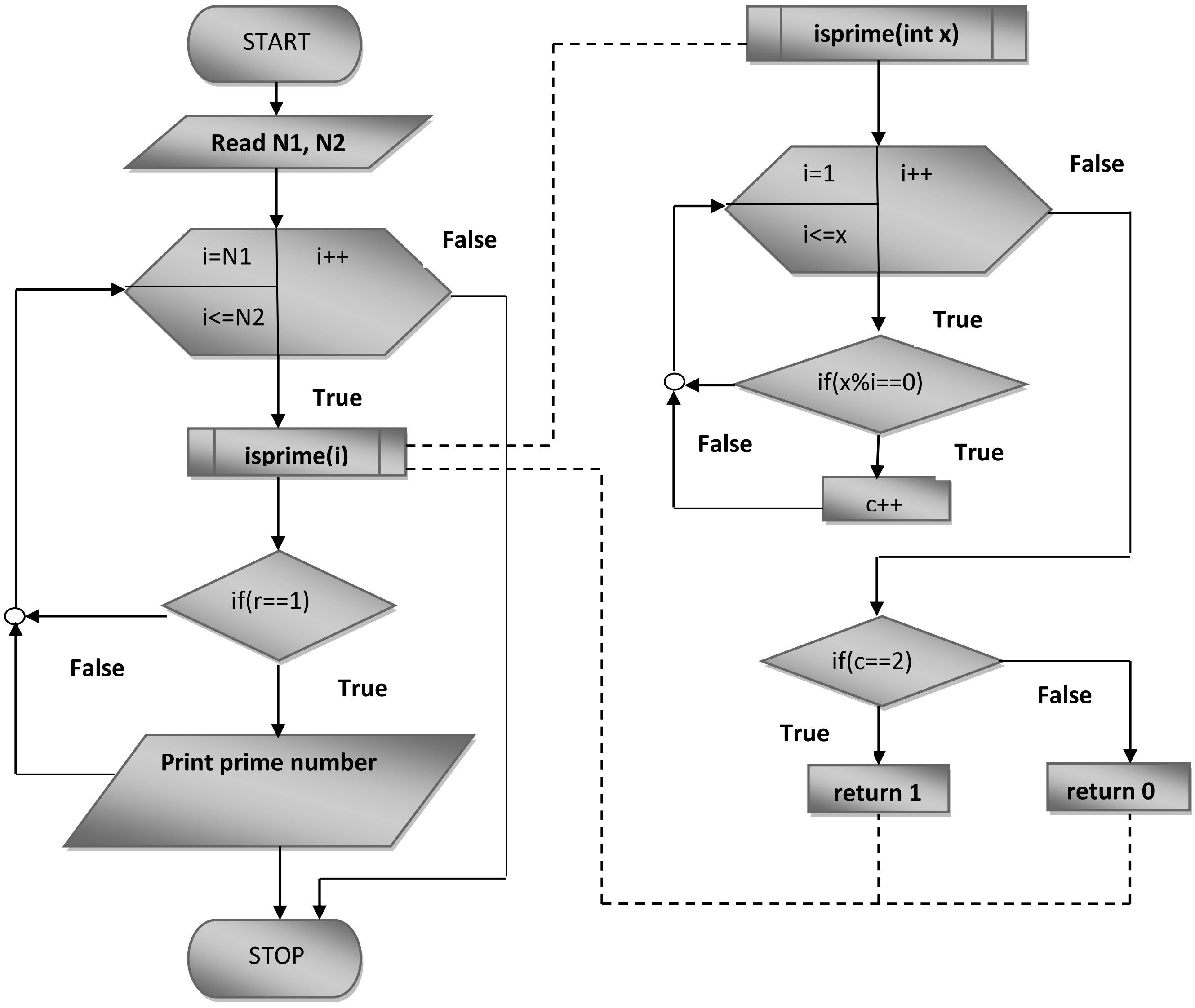
Algorithm for isprime(x)

```
Step 1: Start
Step 2: initialize count=0;
Step 3: For i=1 until x
Step 4: if(x%i==0) then
            Begin
        Count++;
            End if
 End For

Step 5: if(count==2) then
```

```
            Begin
            return 1;
    else
            return 0;
            End if
```

Step 6: Stop

(III) Flow Chart



(IV) Program

```
#include<stdio.h>

#include<conio.h>


int isprime(int); //function prototype


void main()

{

        int n1,n2,r,i;

        clrscr();

        printf("Enter the range n1 and n2 to find prime numbers\n");

        scanf("%d%d",&n1,&n2);  // read the inputs

        printf("The prime numbers between %d and %d are\n",n1,n2);

        for(i=n1;i<=n2;i++)

        {

               r=isprime(i);  //function call

                if(r==1)

               printf("%d\t",i);

        }

 getch();

}


int isprime(int  x)        // function definition

{

  int i,c=0;

  for(i=1;i<=x;i++)

  {

  if(x%i==0)             //check number is completely divisible

  c++;

   }

  if(c==2)         //check number is divisible by 1 and itself
 return(1);
 else
 return(0);
}
```

(V) Output

```
Enter the range n1 and n2 to find prime numbers
10 20
The prime numbers between 10 and 20 are
11 13 17 19

Enter the range n1 and n2 to find prime numbers
1 10
The prime numbers between 10 and 20 are
2  3 5 7
```

(VI) Viva Questions

1. What are prime numbers.

2. What are functions.

3. What are the elements of user defined functions.

4. What is function prototype.

5. What is function call.

6. Differentiate between call by value and call by reference.

7. What are actual and formal parameters.

8. What is recursion.

9. What is base condition in recursion and why it is required.

10. Which data structure is used in recursion.

Program 13: Design and develop a C program to create a structure called Employee to maintain a record of details using an array of structures with four fields (Emp_name, Emp_id, Emp_sal, Emp_age). Assume appropriate data type for each field. Print the Employee details in Tabular Format.

Answer:

(I) Programming Approach

Solution to the given problem can be given in various ways like with/without the usage of functions. To make it simple, here we have implemented without using functions.

First we need to create a structure template with the necessary fields followed with creation of array of structure variables as needed. Then as per the user choice of number of employees, their details will be read and stored in the array of employee structure using dot operator in a loop. The same will be displayed in the tabular format with necessary formatting to the user.

(II) Algorithm

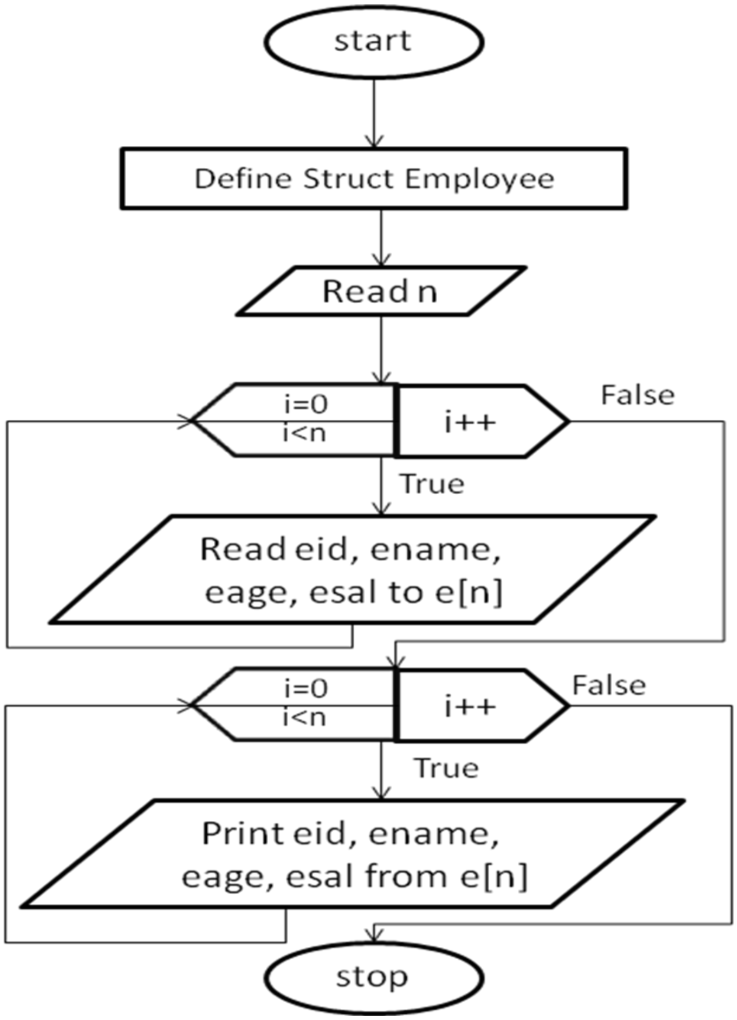
Algorithm: *To read the details of 'n' employees and display the same.*

Input: *Number of Employees (n), for each Employee – his eid, ename, esal, eage.*

Output: *Details of the 'n' employees (Each employee's eid, ename, esal, eage).*

```
Step 1: Start
Step 2: Definition of the structure template.
Step 3: Create aliasing name for the structure using typedef. //for ease of use
Step 4: Declare the array of structure variable as per the need.
Step 5: Read the number of Employees - 'n' value.
Step 6: For i=0 until i=n do
            Read the details of each employee (eid, ename, eage, esal)
            and store the same in the array of structure created.
Step 7: Print the header details of employees
Step 8: For i=0 until i=n do
            Print the details of each employee (eid, ename, eage, esal) in
            a formatted way using the array of structure in which the
            details were stored.
Step 9: End
```

(III) Flow Chart


start
Define Struct Employee
Read n
i=0
i<n
i++
False
True
Read eid, ename, eage, esal to e[n]
i=0
i<n
i++
False
True
Print eid, ename, eage, esal from e[n]
stop

(IV) Program

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

//Definition of an Employee structure with the necessary fields
struct employee
{
         int eid,eage;
         char ename[20];
        float esal;
}

//typed definition gives aliasing for "struct employee" as "emp"
typedef struct employee emp;

void main()
{
         emp e[50]; //e - Array of structure employee
         int n,i;  // n - for number of employees and i - a loop iterator
          clrscr();
   // Read the number of Employees


          printf("\n Enter the number of Employees: ");
          scanf("%d",&n);
  // Read the details of n -  employees
         for(i=0;i<n;i++)
        {
                 printf("\n Enter the details of Employee : %d\n",i+1);
```

```
                        printf("\n Enter eid:");
                     scanf("%d",&e[i].eid);
                     printf("\nEnter ename:");
                     scanf("%s",e[i].ename);
                      printf("\nEnter eage:");
                     scanf("%d",&e[i].eage);
                     printf("\nEnter esal:");
                     scanf("%f",&e[i].esal);
                    }
   //Display the details of n = eployees with proper formatting
   printf("\n***********************************************\n");
   printf("\nDetails of %d Employees are as follows\n",n);
   printf("\n***********************************************\n");
   printf("\nEmpid\tEmpname\tEmpage\tEmpsal\n");
   printf("\n---------------------------------------------\n");
   for(i=0;i<n;i++)
   {
     printf("\n%d\t%s\t%d\t%f",e[i].eid,e[i].ename,e[i].eage,e[i].esal);
   }
   printf("\n---------------------------------------------\n");
   getch();
   }
```

(V) Output

```
Enter the number of Employees: 3
Enter the details of Employee : 1
Enter eid: 111
Enter ename: Akash
Enter eage: 21
Enter esal: 20000

Enter the details of Employee : 2
Enter eid: 222
Enter ename: Banu
Enter eage: 22
Enter esal: 30000

Enter the details of Employee : 3
Enter eid: 333
Enter ename: Chaithra
Enter eage: 23
Enter esal: 32000




****************************************************************
Details of 3 Employees are as follows
****************************************************************

Empid      Empname               Empage       Empsal
----------------------------------------------------------------------
111        Akash                 21           20000.000000
222        Banu                  22           30000.000000
333        Chaithra              23           32000.000000
-----------------------------------------------------------------------
```

(VI) Viva Questions

1.Define Structure.

2.Differentiate between array of structures and a structure array.

3.Elaborate the usage of typedef in the program.

4.Differentiate between inbuilt data types and user defined data types.

5.Explain how an individual member of the structure can be accessed through a structure variable.

6. Describe the usage of structure within a structure.

7. Can the structure variables access the values of the local variables?

8. Can a structure variable be used as a function parameter/ argument?

9. Is it possible to create a structure variable without a structure name?

10.Explain how pointers can be used to access the fields of the structure.

Program 14: Write a C program using pointers to compute the sum, mean and standard deviation of all elements stored in an array of n real numbers.

Answer:

(I) Mathematical approach

Elements of an array: x_1, x_2, x_3….., x_N

Sum=x_1+x_2+x_3…..+x_N

Mean=µ= (x_1+x_2+x_3…..+x_n)/N=Sum/N

Standard deviation:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\mu)^2}$$

(II) Algorithm

Algorithm-Computation of sum, mean, standard deviation of array elements [This algorithm takes array elements as input and outputs sum, mean, standard deviation for the given elements]

Step 1. Start

Step 2. Enter the number of elements 'n'.

Step 3. In a loop enter the elements.

Step 4. Calculate the sum in for loop.

Step 5: sum=sum+*ptr

Step 6. Compute mean of the array elements

mean=sum/n

Step 7. Compute standard deviation of the array elements

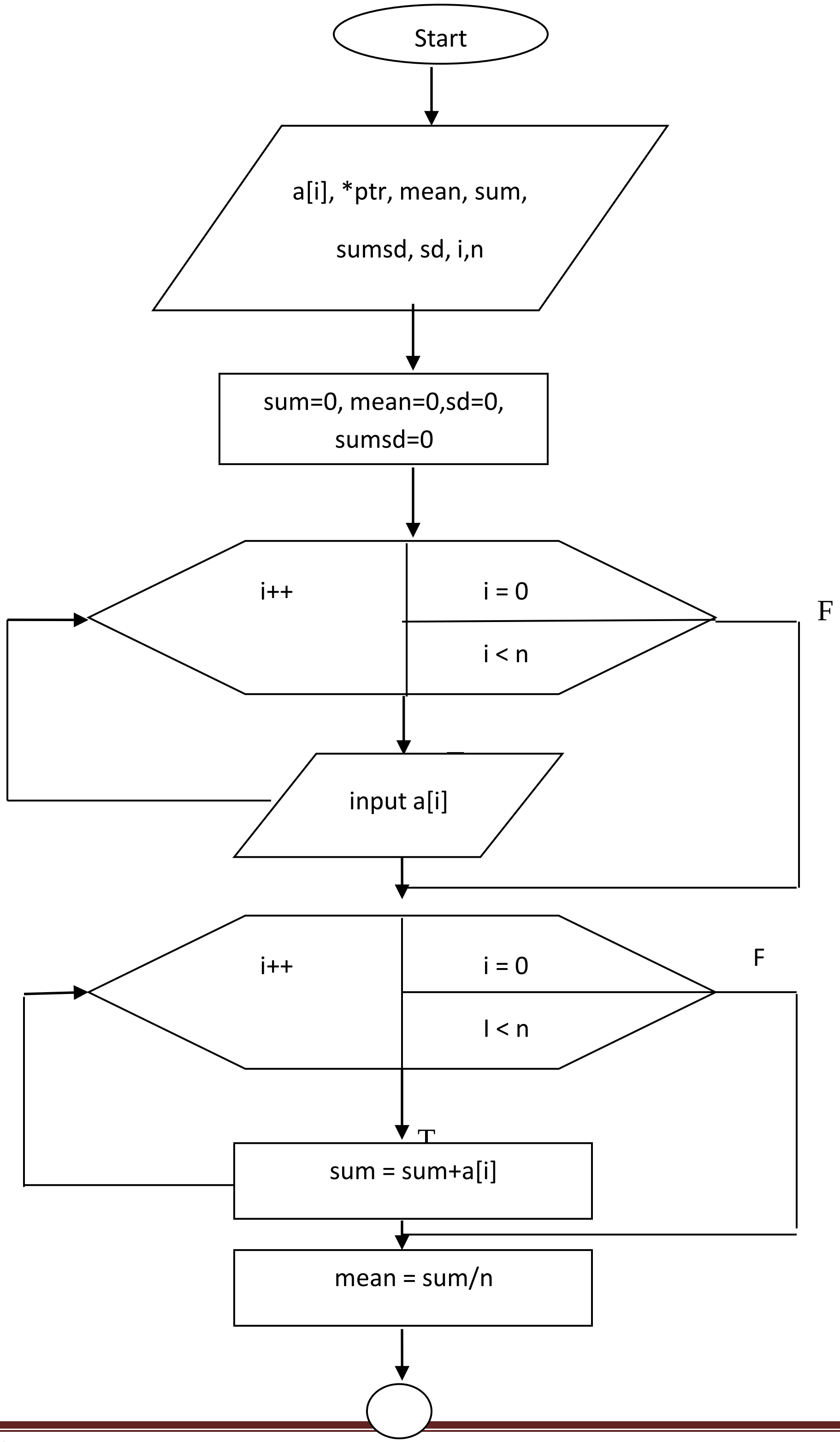
sumstd=sumstd+pow((*ptr-mean),2)

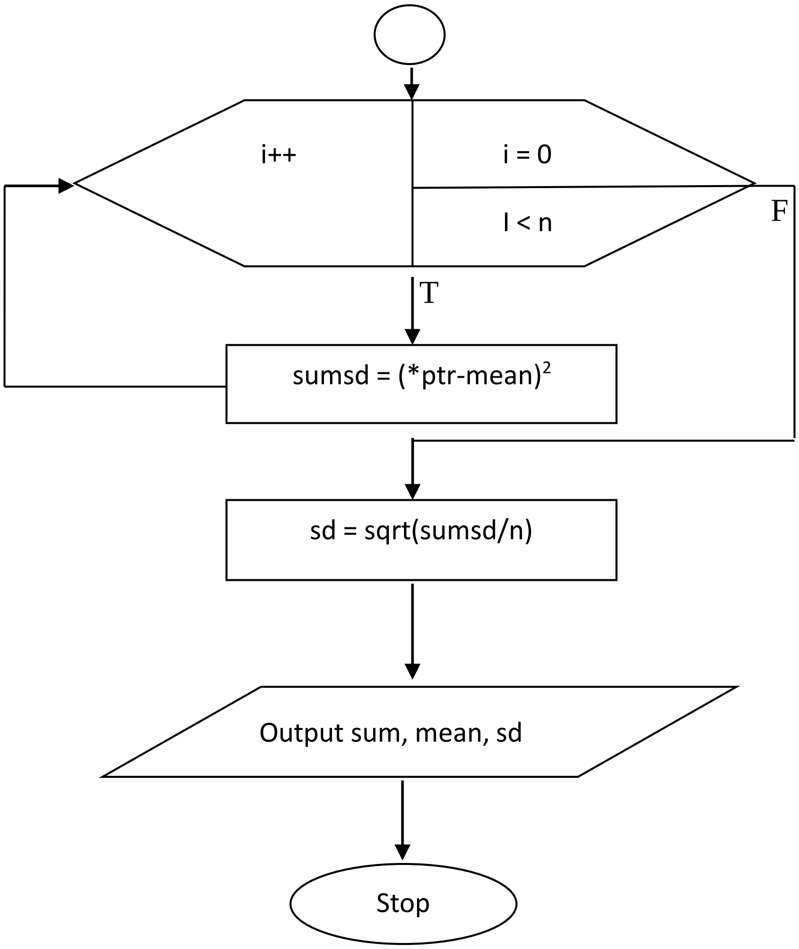
std=sqrt(sumstd/n);

Step 8. Print the values of sum, mean and standard deviation for the given array elements

Step 9. Stop

III Flowchart




i++
i = 0
I < n
F
T
sumsd = (*ptr-mean)^2
sd = sqrt(sumsd/n)
Output sum, mean, sd
Stop

(III) <u>Program</u>

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
  float a[10],*ptr,mean=0,sum=0,sd=0,sumsd=0;
   int i,n;
   printf("\nEnter the number of elements:");
   scanf("%d",&n);
   printf("\nEnter the elements: ");
   for(i=0;i<n;i++)
       scanf("%f",&a[i]);
   ptr=a;
   for(i=0;i<n;i++)
   {
       sum=sum+*ptr;
       ptr++;
     }
    mean=sum/n;
     ptr=a;
    for(i=0;i<n;i++)
     {
        sumsd=sumsd+pow((*ptr-mean),2);
        ptr++;
       }
     sd=sqrt(sumsd/n);
     printf("\nSum=%f",sum);
```

```
        printf("\nMean=%f",mean);
        printf("\nStandard deviation=%f",sd);
        getch();
    }
```

(IV) Output

....

Run 1:

```
Enter the number of elements
5
Enter array elements
1
2
3
4
5
Sum=15.000
Mean=3.000
Standard Deviation =1.414
```

Run 2:

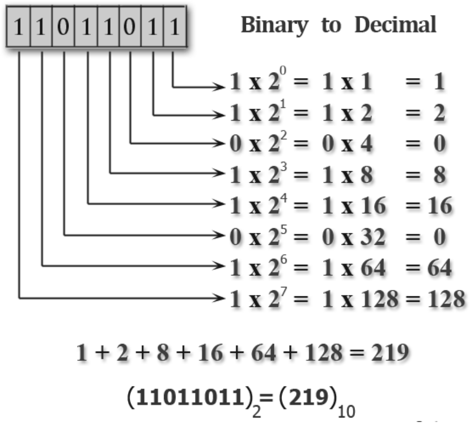
```
Enter the number of elements
4
Enter array elements
10.5
25.25
30.56
9.5
Sum=75.810
Mean=18.952

Standard Deviation =9.154
```

(V) Viva Questions

1. What is a pointer?
2. Define single dimension and multi-dimensional arrays?
3. What is the formula for calculating sum, mean and standard deviation of given elements?

Program 15. Implement recursive program to convert binary to decimal

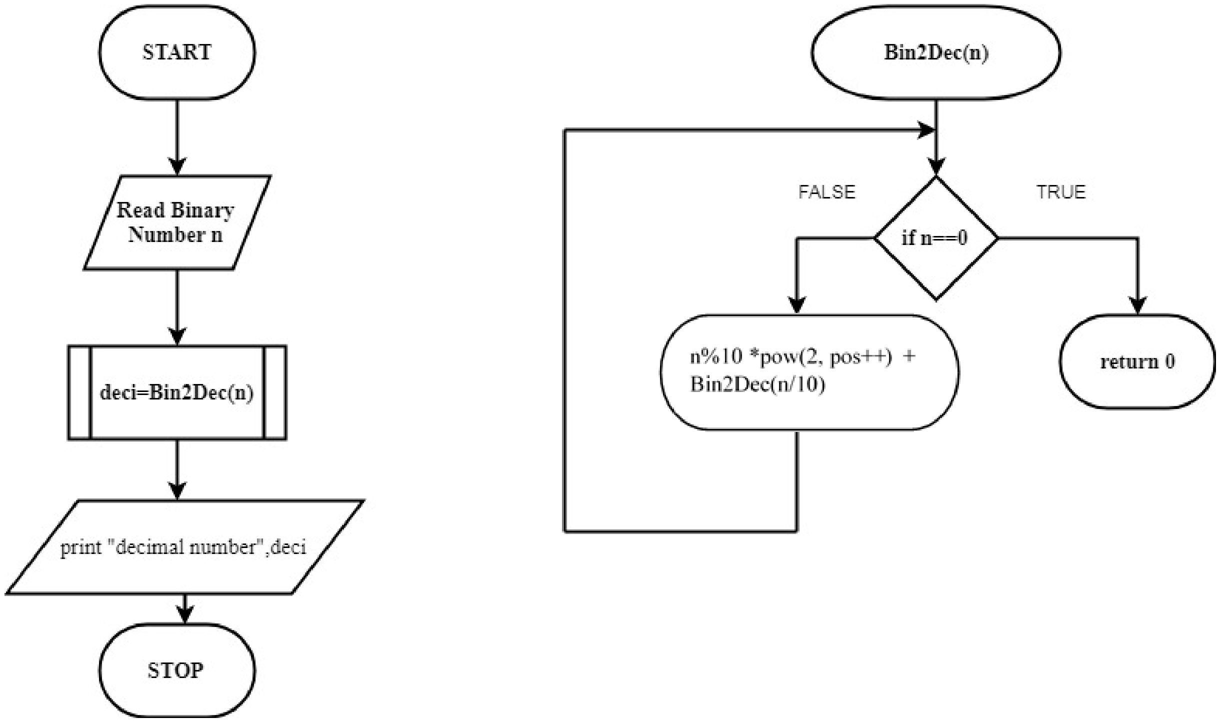
(I) Mathematical Approach



(II) Algorithm

```
Step 1: [Initialize i]
    i=0
Step 2: [Read binary number]
    Read n
Step 3: [Repeat step 2 until n =0]
    if n=0
            return 0;
    else
            digit=n%10
            n=n/10
            return digit*pow(2,i++) + Bin2Dec(n);
    endif
```

(III) Flowchart



(IV) Program

```
#include<stdio.h>
#include<math.h>
int Bin2Dec(long long int);
int pos=0;
void main()
{
    long long int num;
    int deci;
    printf("Enter Binary number\n");
    scanf("%lld",&num);
    deci=Bin2Dec(num);
```

```
     printf("Decimal number is %d\n",deci);
}

int Bin2Dec(long long int n)
{
    if (n==0)
      return 0;
    else
       return n%10 *pow(2, pos++)  + Bin2Dec(n/10);

}
```

(V) Output

```
Enter a binary number
10101001
Decimal number is  169
```

(VI) Viva Question

1. What is recursive program ?
2. What are the advantages of recursion program >
3. Difference between local and global variables ?
4. What is the size of long long int ?