

### Exercise - 7

⇒ Problem statement :-

Accept an array of 'n' elements and display its reverse.

⇒ Algorithm:

Step 1: Start

Step 2: declare an array

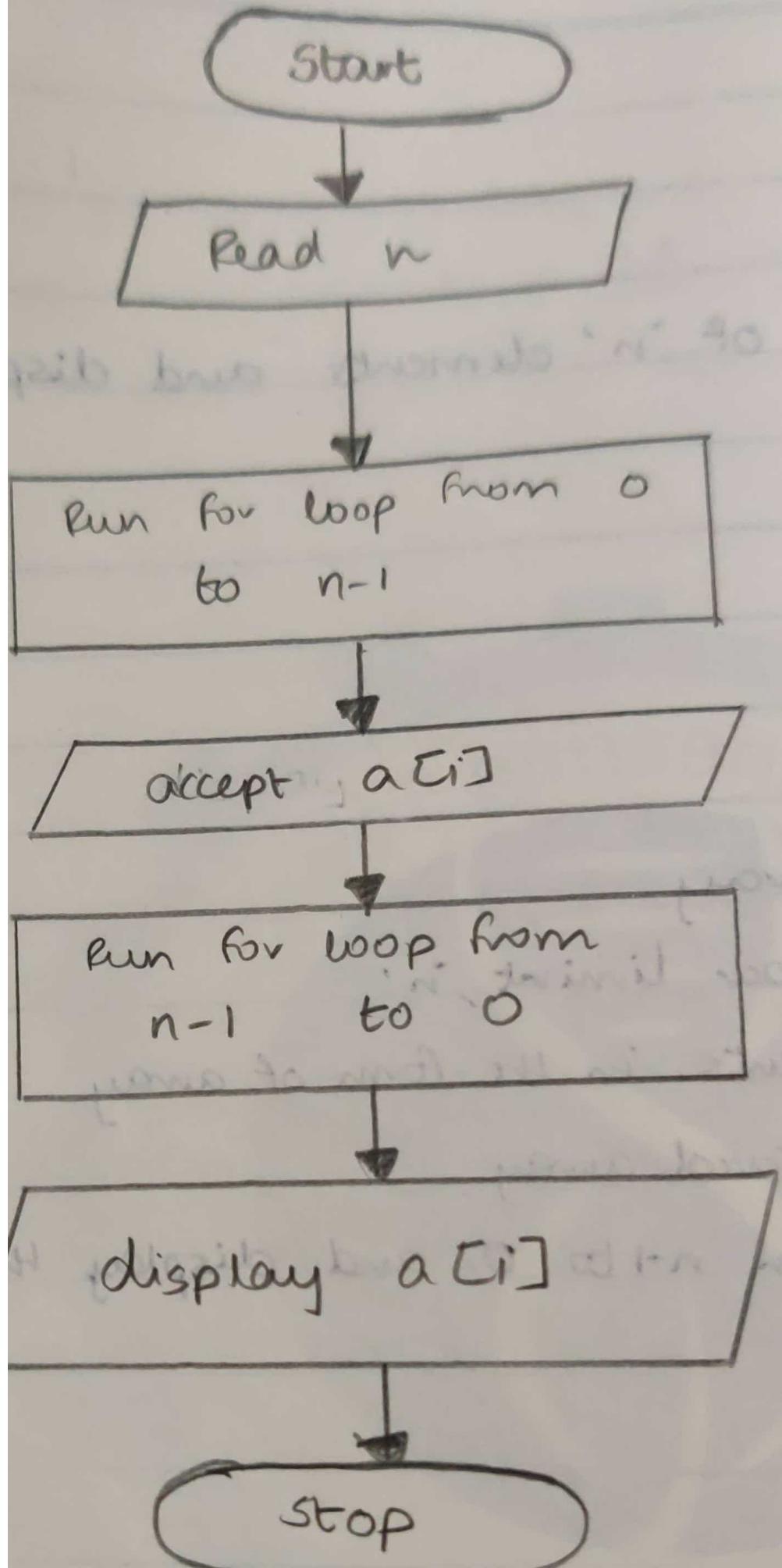
Step 3: Accept the upper limit,  $n$

Step 4: Accept  $n$  elements in the form of array

Step 5: Display the entered array

Step 6: Run for loop from  $n-1$  to 0 and display the array.

Step 7: Stop



Q] Problem statement: write a code to accept an array and display its reverse.  
Code: #include<stdio.h>

```
int main()
{
int n, i;
int a[n];
printf("\nEnter a number : ");
scanf("%d", &n);
printf("\nEnter %d nos : ", n);
for(i=0;i<n;i++)
{
scanf("%d", &a[i]);
}
printf("\nThe entered array is : ");
for(i=0;i<n;i++)
{
printf("\n%d", a[i]);
}
printf("\nThe reverse of the entered array is : ");
for(i=n-1;i>=0;i--)
{
printf("\n%d", a[i]);
}
return 0;
}
```

Enter a number : 5

enter 5 nos : 2

4

5

8

9

1

The entered array is :

2

4

5

8

9

The reverse of the entered array is :

9

8

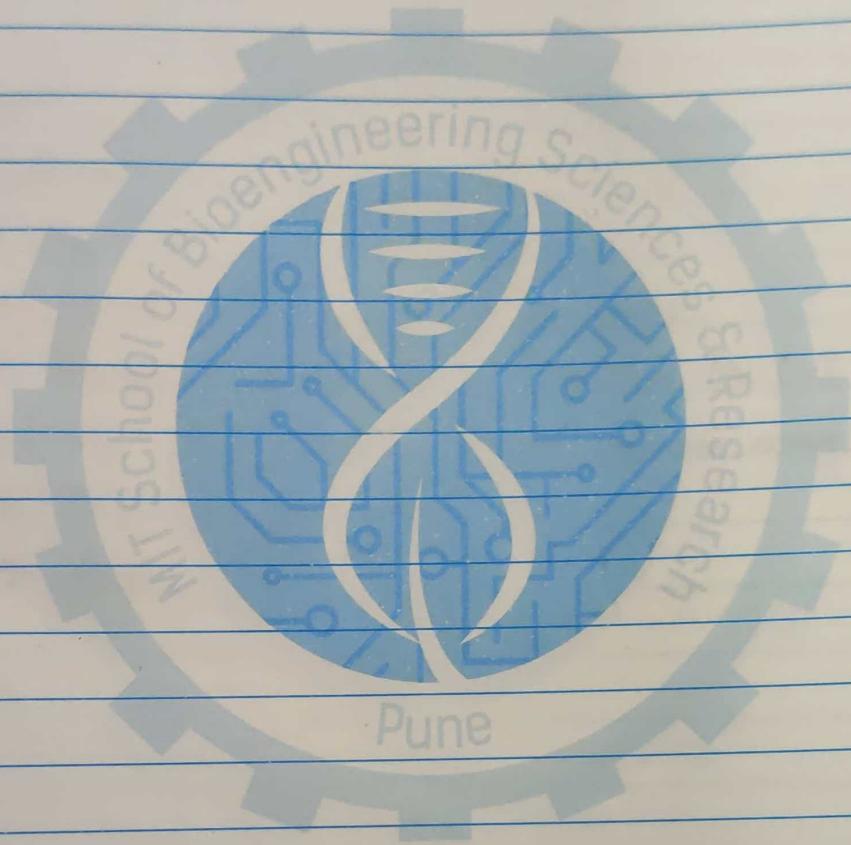
5

4

2

⇒ Conclusion:

- Reverse of accepted array was successfully displayed.
- for loop was used to accept and display the entered array and its reverse.





### Exercise - 8

⇒ Problem statement:

Write a code to search and display the number of occurrences of a particular value in an array.

⇒ Algorithm:

Step 1: Start

Step 2: Read n

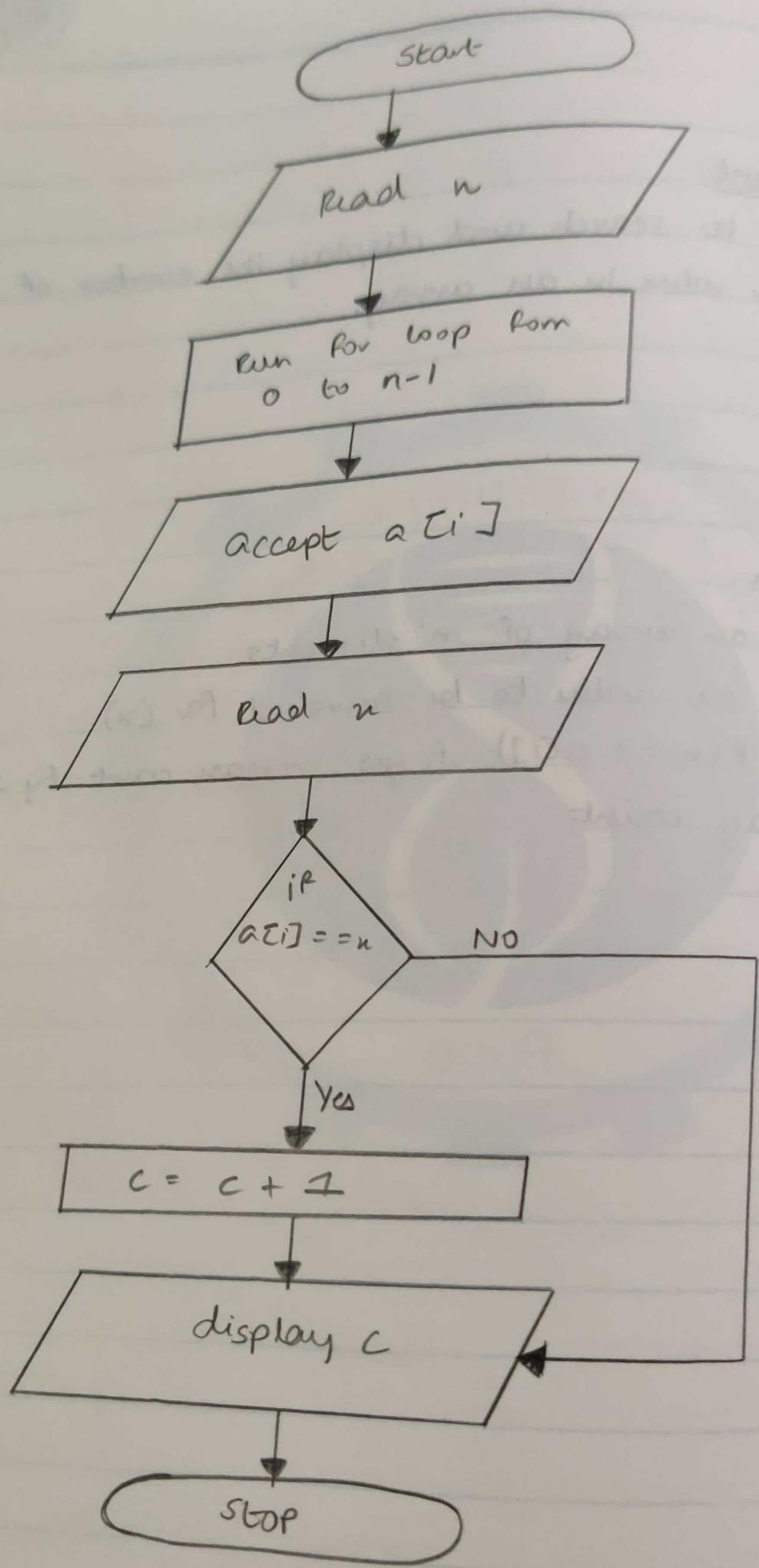
Step 3: Accept an array of 'n' elements

Step 4: Accept a value to be searched for (x)

Step 5: check if ( $x == a[i]$ ) if yes increase count by 1.

Step 6: display count.

Step 7: Stop



Q] Problem statement: Write a code to accept an array and search and display the number of occurrences of a particular value in an array

Code:

```
#include<stdio.h>

int main()

{
int n , i , x , c=0;
int a[n];
printf("\nEnter a number : ");
scanf("%d" , &n);
printf("\nEnter %d nos : " , n);
for(i=0;i<n;i++)
{
scanf("%d" , &a[i]);
}
printf("\nThe entered array is : ");
for(i=0;i<n;i++)
{
printf("\n%d" , a[i]);
}
printf("\nEnter a number : ");
scanf("%d" , &x);
for(i=0;i<n;i++)
{
if(a[i]==x)
{
c++;
}
printf("\n%d is the location where %d is present" , i , x);
```

```
}

}

printf("\nthe number of times the entered value occured in the array entered is %d", c);

}
```

```
Enter a number : 5
enter 5 nos : 4
5
8
6
6
The entered array is :
4
5
8
6
6
Enter a number : 6
3 is the location where 6 is present
4 is the location where 6 is present
the number of times the entered value occured in the array entered is 2.
```



⇒ Conclusion:-

- A program to search and display the number of occurrences of a particular value in an array was successfully compiled and executed.
- for loop was used to accept and display the an array of n elements.
- if conditional statement was used to check if the element of the array at a particular location is equal to the entered value by the user.
- Increment operator is used to increase the value of count if the conditional statement is true.

### Exercise - 9

⇒ Problem statement:-

Write a code to accept an array of  $n$  elements and to display it in ascending order.

⇒ Algorithm:-

Step 1: Start

Step 2: Read  $n$

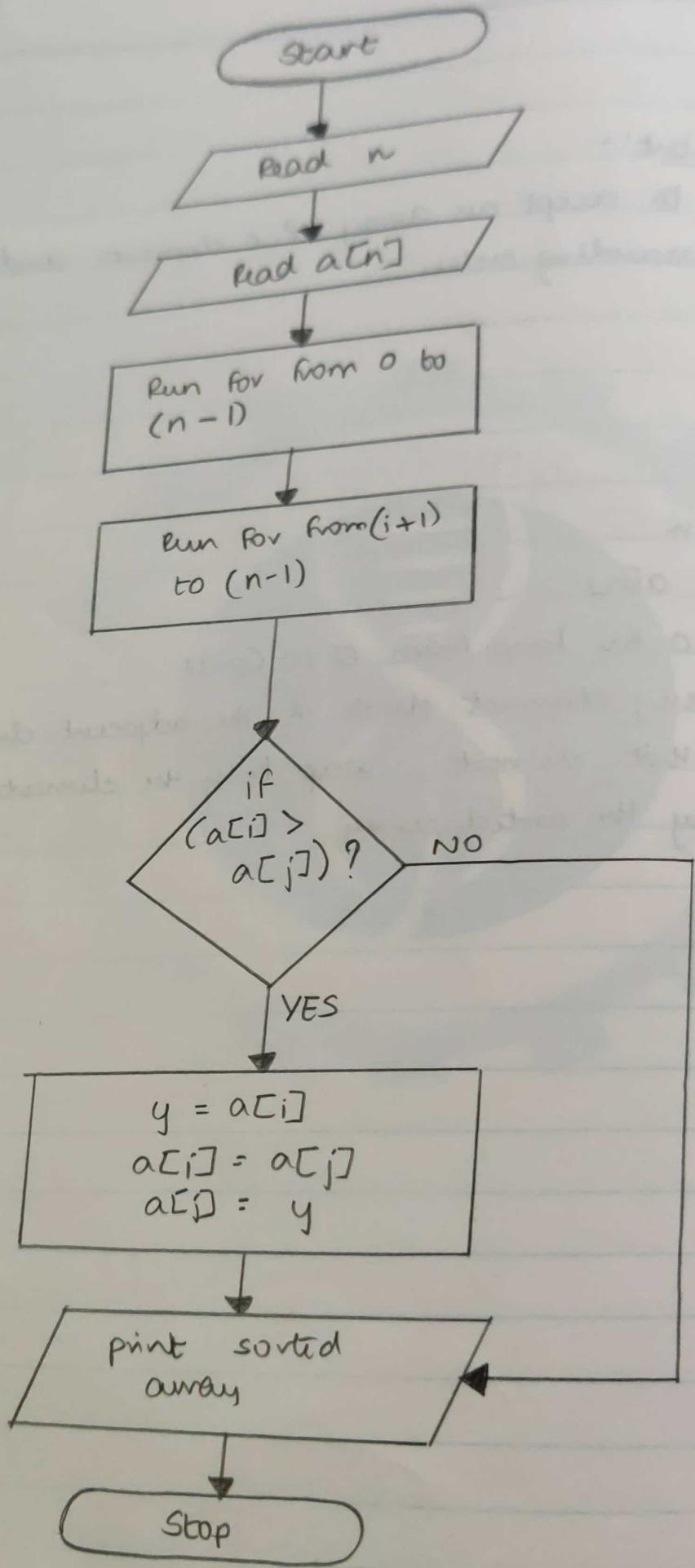
Step 3: Read  $a[n]$

Step 4: Run a for loop from 0 to  $(n-1)$

Step 5: for every element check if the adjacent element is greater than that element, swap both the elements.

Step 6: Display the sorted array

Step 7: Stop



Q) Problem statement : Write a code to display the array in ascending order.

Code:

```
#include<stdio.h>

int main()

{
    int n, i, y;
    int a[n];
    printf("\nEnter a number : ");
    scanf("%d", &n);
    printf("\nEnter %d nos : ", n);
    for(i=0;i<n;i++)
    {
        scanf("%d", &a[i]);
    }
    printf("\nThe entered array is : ");
    for(i=0;i<n;i++)
    {
        printf("\n%d", a[i]);
    }
    for(int i=0;i<n;i++)
    {
        for(int j=i+1; j<n;j++)
        {
            if(a[i]>a[j])
            {
                y=a[i];
                a[i]=a[j];
                a[j]=y;
            }
        }
    }
}
```

```
a[i]=a[j];  
a[j]=y;  
}  
}  
}  
printf("\nThe sorted array is : ");  
for(i=0;i<n;i++)  
{  
printf("\n%d", a[i]);  
}  
return 0;  
}
```

Enter a number : 5

enter 5 nos : 4

8  
6  
1  
2

The entered array is :

4  
8  
6  
1  
2

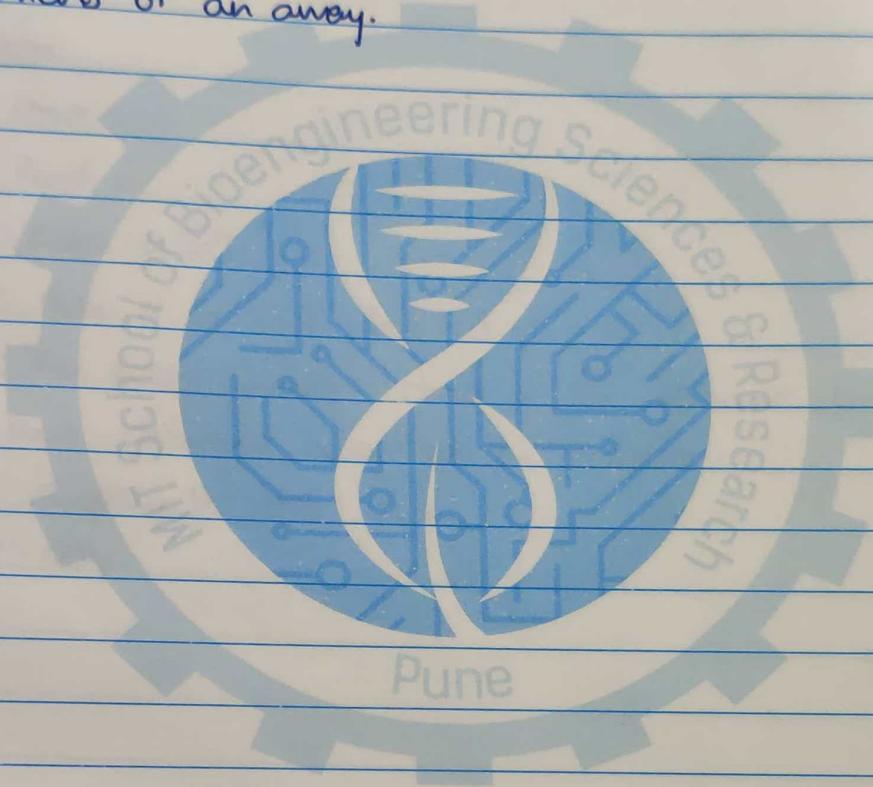
The sorted array is :

1  
2  
4  
6  
8



### Conclusion:

- A program to accept array of 'n' elements and display it in ascending order was successfully compiled and executed.
- for loop was used to accept and display and compare the elements of an array.
- If conditional statement was used to compare the adjacent elements of an array.



### Exercise - 10

→ Problem Statement:

Write a code to accept an array of 'n' elements and  
arrange the array in descending order

→ Algorithm:

Step 1: Start

Step 2: Read n

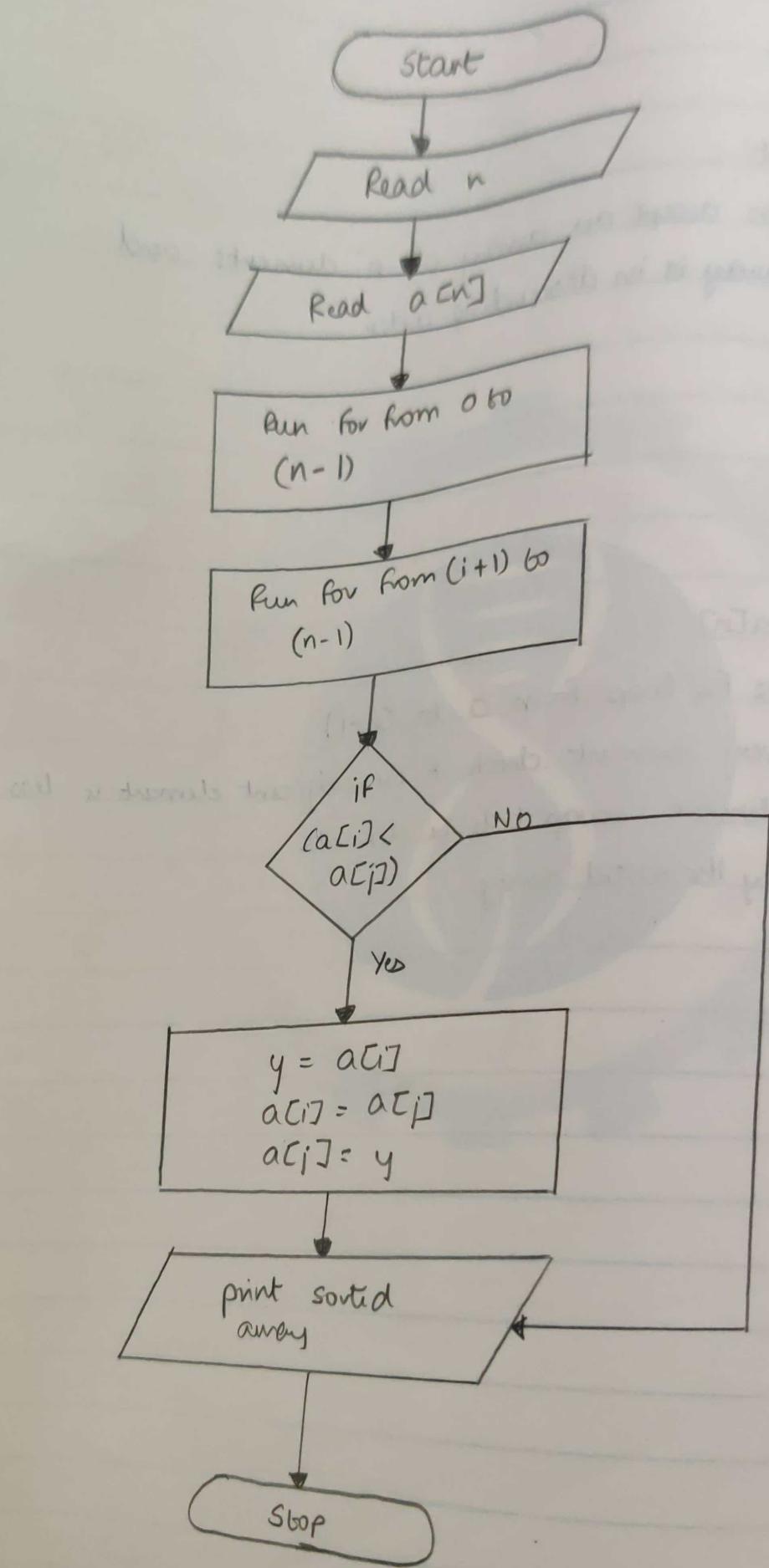
Step 3: Read a[n]

Step 4: Run a for loop from 0 to (n-1)

Step 5: For every element check if the adjacent element is less  
than that element, swap both the elements.

Step 6: display the sorted array

Step 7: Stop.



Q) problem statement : Write a code to accept an array and display it in descending order

Code:

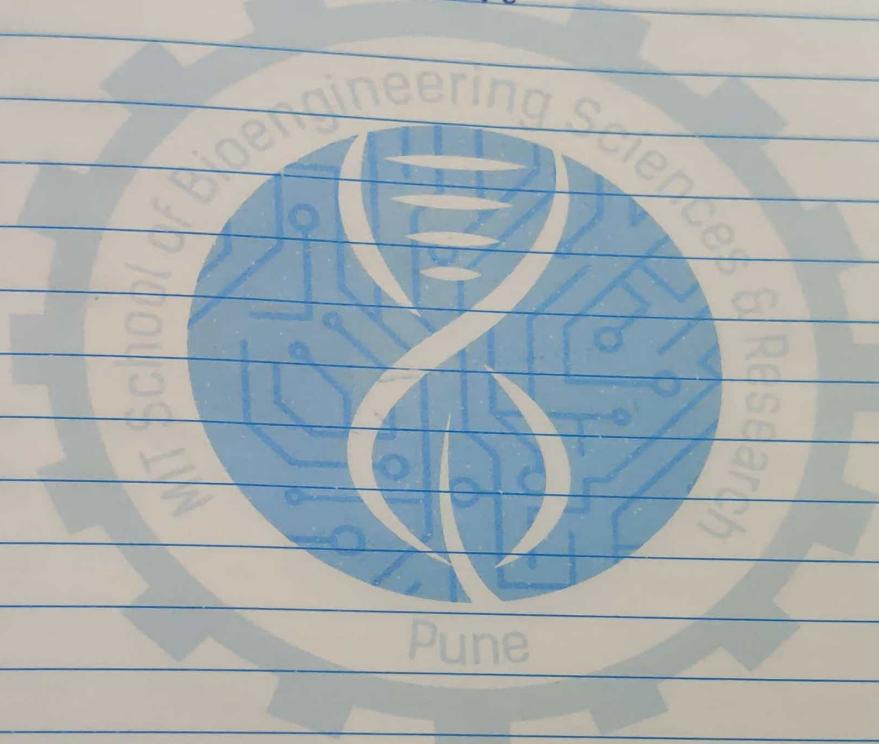
```
#include<stdio.h>
int main()
{
    int n, i, y;
    int a[n];
    printf("\nEnter a number : ");
    scanf("%d", &n);
    printf("\nEnter %d nos : ", n);
    for(i=0;i<n;i++)
    {
        scanf("%d", &a[i]);
    }
    printf("\nThe entered array is : ");
    for(i=0;i<n;i++)
    {
        printf("\n%d", a[i]);
    }
    for(int i=0;i<n;i++)
    {
        for(int j=i+1; j<n;j++)
        {
            if(a[i]<a[j])
            {
                y=a[i];
                a[i]=a[j];
                a[j]=y;
            }
        }
    }
}
```

```
)  
)  
)  
)  
printf("\nThe sorted array is : ");  
for(i=0;i<n;i++)  
{  
    printf("\n%d", a[i]);  
}  
return 0;  
}
```

```
Enter a number : 5  
enter 5 nos : 4  
6  
8  
2  
1  
The entered array is :  
4  
6  
8  
2  
1  
The sorted array is :  
8  
6  
4  
2  
1
```

→ Conclusion:

- A program was successfully compiled and executed to accept an array of 'n' elements and arrange it in descending order.
- For loop was used to accept and display the entered array in descending order.
- IF conditional statement was used to check if the adjacent number is less than that particular element.



### Exercise - 11

⇒ Problem statement:

Write a code to accept 3 strings or names and find the length of the strings and display it.

⇒ Algorithm:

Step 1: Start

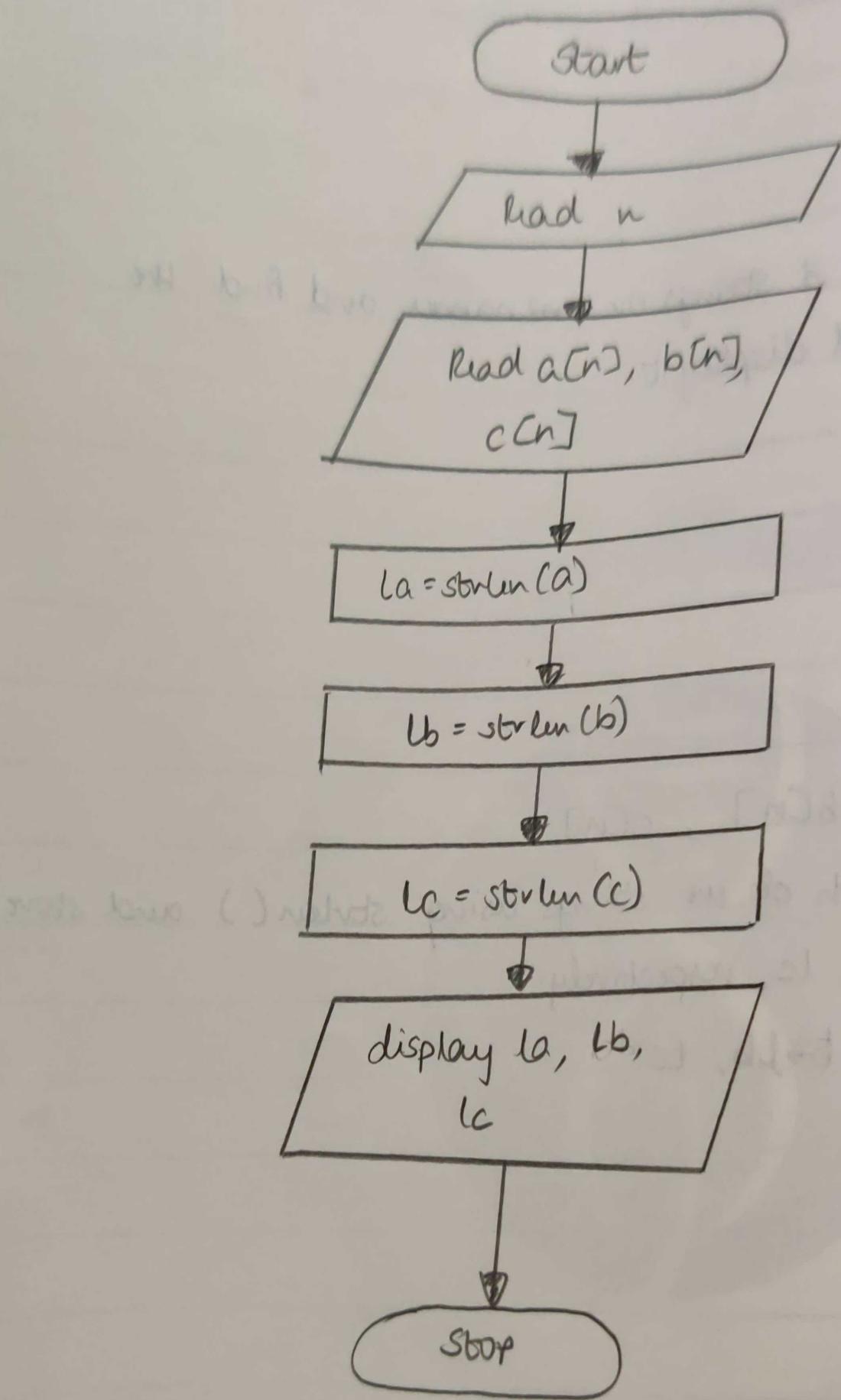
Step 2: Read n

Step 3: Read a[n], b[n], c[n]

Step 4: Find the length of the strings using strlen() and store in la, lb, lc respectively.

Step 5: display la, lb, lc.

Step 6: Stop



Q1 Problem Statement: Write a code to accept 3 strings and find their length.

Code:

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

int main()

{
    printf("\nAnushka surdas");
    printf("\nMITU22BTB10016");

    char c1[20];
    char c2[20];
    char c3[20];
    printf("\nEnter your name : ");
    fgets(c1 , sizeof(c1) , stdin);
    c1[strcspn(c1, "\n")]=0;
    printf("\nEnter your name : ");
    fgets(c2 , sizeof(c2) , stdin);
    c2[strcspn(c2, "\n")]=0;
    printf("\nEnter your name : ");
    fgets(c3 , sizeof(c3) , stdin);
    c3[strcspn(c3, "\n")]=0;

    int l1,l2,l3;
    l1=strlen(c1);
    l2=strlen(c2);
    l3=strlen(c3);

    printf("\nthe length of %s is : %lu" , c1 , l1);
    printf("\nthe length of %s is : %lu" , c2 , l2);
    printf("\nthe length of %s is : %lu" , c3 , l3);
```

```
return 0;
```

```
}
```

```
Anushka surdas
```

```
MITU22BTBI0016
```

```
Enter your name : anushka
```

```
Enter your name : mukul
```

```
Enter your name : surdas
```

```
the length of anushka is : 7
```

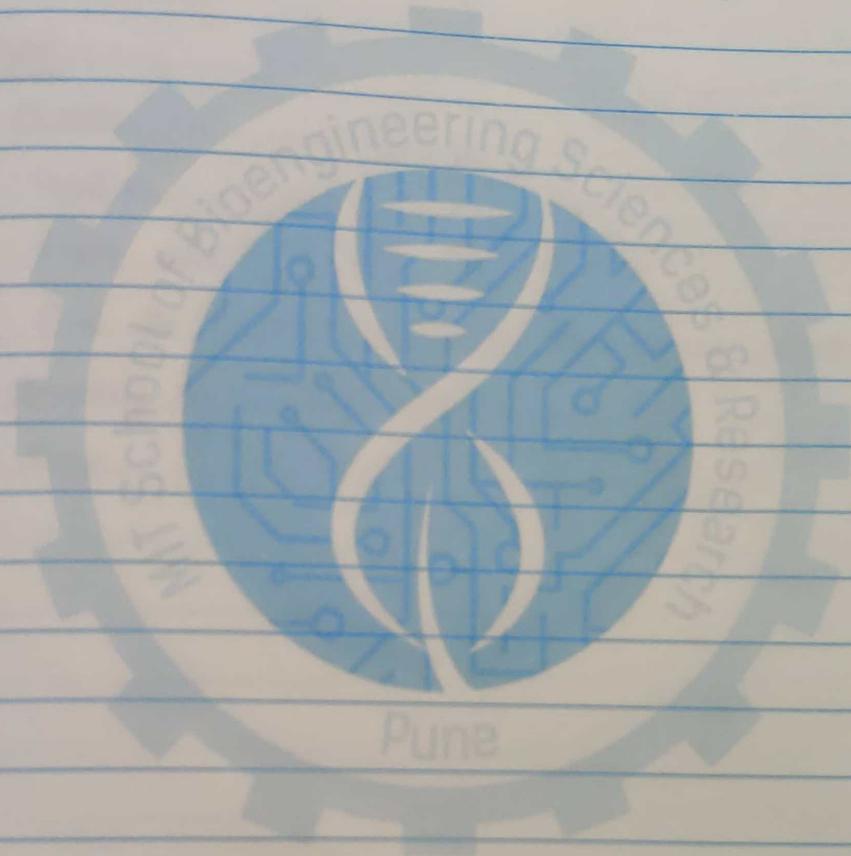
```
the length of mukul is : 5
```

```
the length of surdas is : 6
```



⇒ Conclusion:

- A program to compi accept an array of characters and find its length was successfully compiled and executed.
- For was used to accept and display the entered array.
- <String.h> is included to include its functions
- strlen() was used to find the lengths of strings entered.





### Exercise - 12

#### ⇒ Problem statement:

write a code to accept 2 arrays of characters of 'n' elements and compare them.

#### ⇒ Algorithm:

Step 1: Start

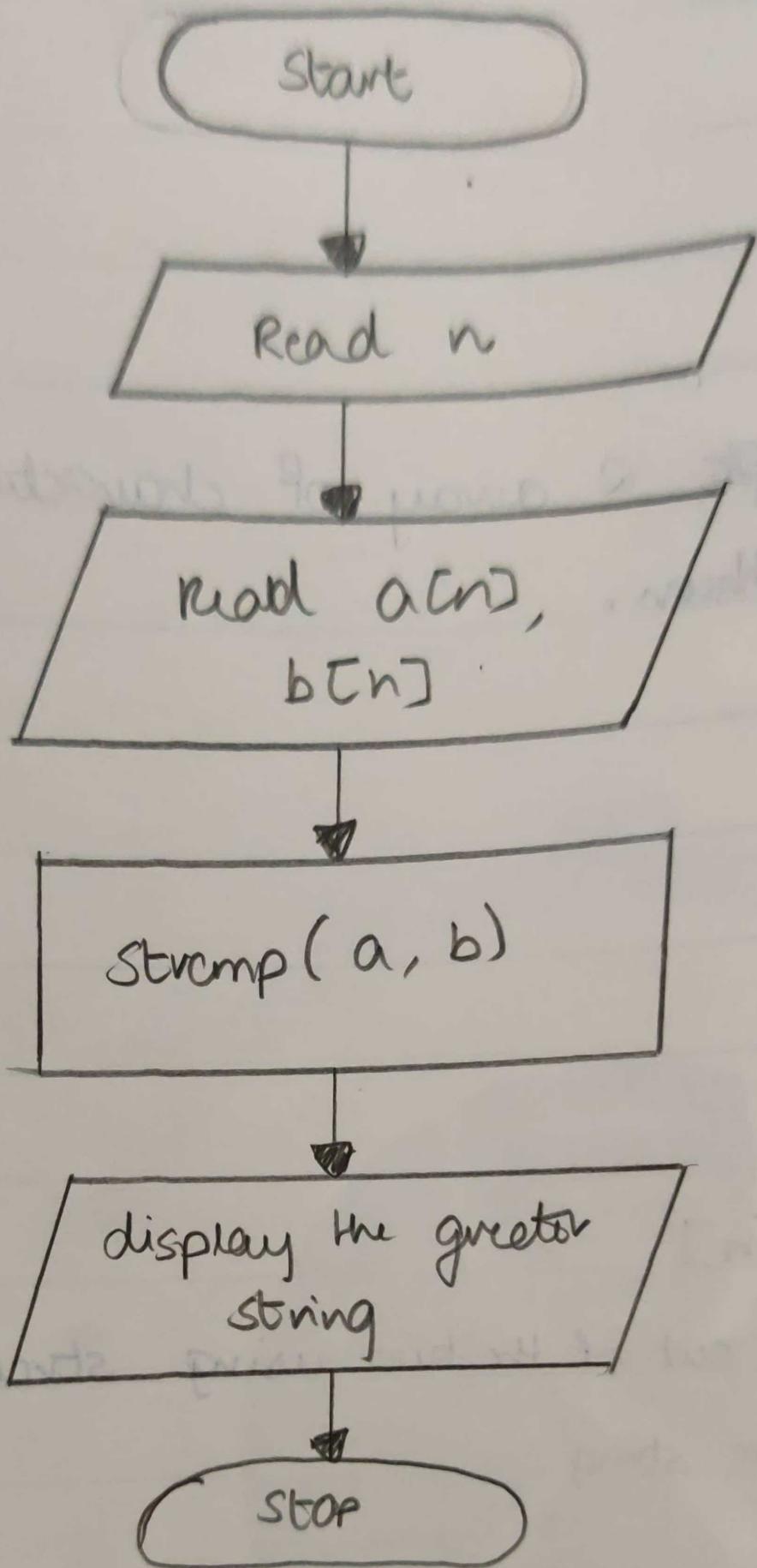
Step 2: Read n

Step 3: Read a[n], b[n]

Step 4: Find the greater out of the two using strcmp()

Step 5: display the greater string

Step 6: Stop.



a) problem statement : write a code to accept 2 strings and compare them.

Code:

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

main()
{
    printf("\nAnushka surdas");
    printf("\nMITU22BTB10016");
    char c1[20];
    char c2[20];
    char c3[20];
    printf("\nEnter your name : ");
    fgets(c1 , sizeof(c1) , stdin);
    c1[strcspn(c1, "\n")]=0;
    printf("\nEnter your name : ");
    fgets(c2 , sizeof(c2) , stdin);
    c2[strcspn(c2, "\n")]=0;
    int sc=strcmp(c1,c2);
    printf("%d" , sc);
    if(sc==0)
        printf("\nstring 1 is equal to string 2");
    else if(sc<0)
        printf("\nString 1 is less than string 2");
    else
        printf("\nString 2 is less than string 1");
}
```

Anushka surdas

MITU22BTBI0016

Enter your name : anushka

Enter your name : mukull

-1  
String 1 is less than string 2

→ Conclusion:

- A program was successfully compiled and executed to compare two arrays of characters and display the greater string.
- for loop was used to accept and display the entered array.
- strcmp() was used to compare the two strings.

### Exercise - 13

→ Problem statement:

Write a code to accept two arrays of characters and combine the strings.

→ Algorithm:

Step 1: Start

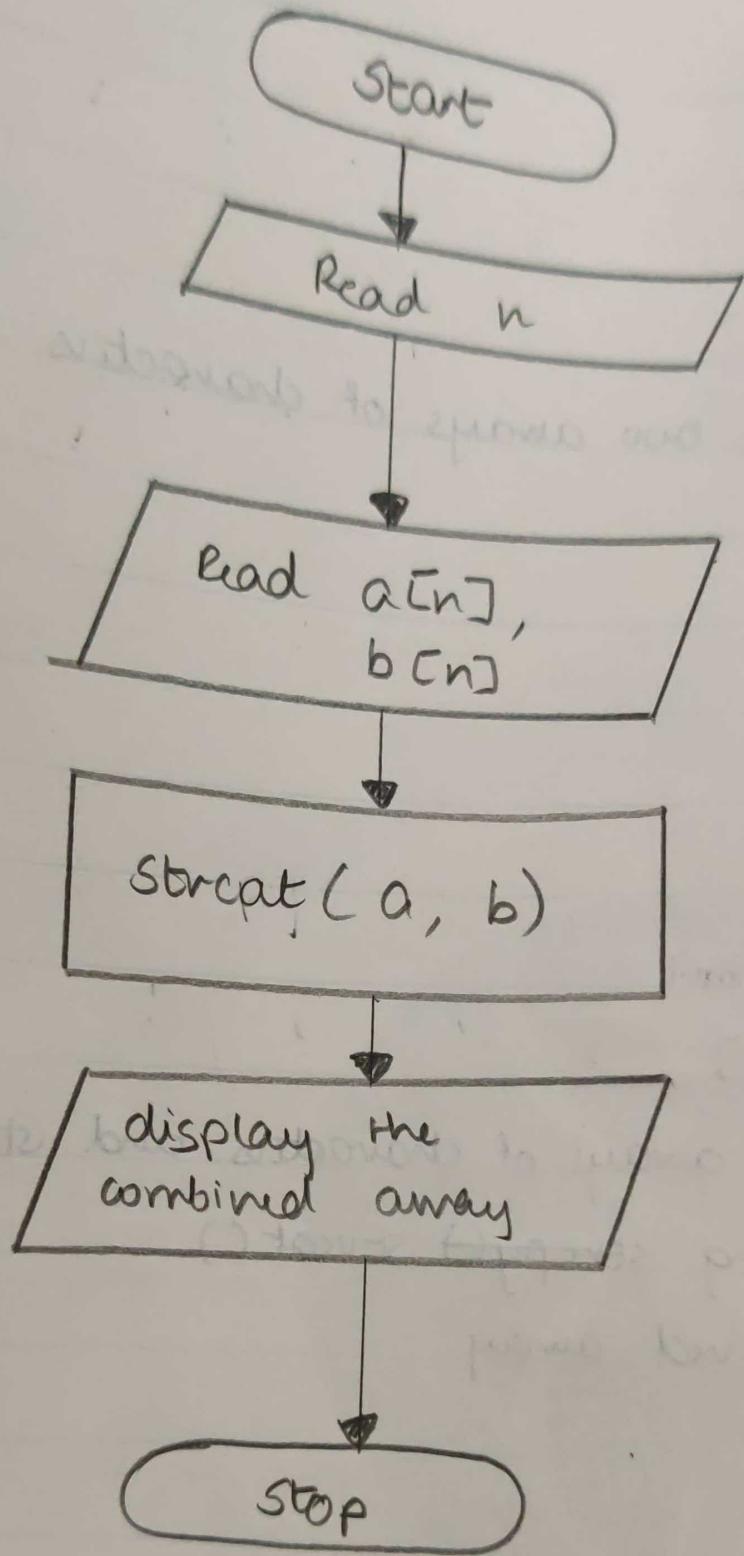
Step 2: Read n, upper limit

Step 3: Read a[n], b[n]

Step 4: Combine both the arrays of characters and store it in a third variable using strcpy() strcat()

Step 5: display the combined array

Step 6: Stop



Q) problem statement : Write a code to accept 2 strings and combine them .

Code:

```
#include<stdio.h>
#include<string.h>
main()
{
    printf("\nAnushka surdas");
    printf("\nMITU22BTB10016");
    char c1[20];
    char c2[20];
    char c3[20];
    printf("\nEnter your name : ");
    fgets(c1 , sizeof(c1) , stdin);
    c1[strcspn(c1, "\n")]=0;
    printf("\nEnter your name : ");
    fgets(c2 , sizeof(c2) , stdin);
    c2[strcspn(c2, "\n")]=0;
    printf("%s" , strcat(c1,c2));
}
```

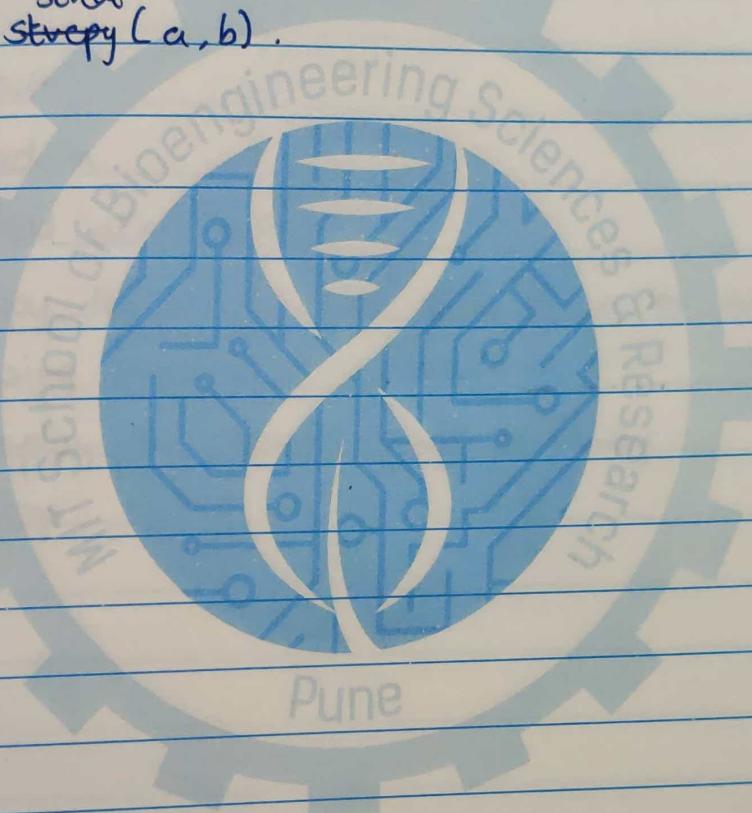
```
Anushka surdas
MITU22BTB10016
Enter your name : anushka
Enter your name : surdas
anushka surdas
```

### Conclusion:

A program was successfully compiled and executed to accept two arrays of characters and join or combine the two and display them.

for loop was used to accept and display the array of characters.

`<string.h>` library was used included to use the string function, `strcat`, `strcpy(a, b)`.



### Exercise - 14

⇒ Problem statement:

Write a code to accept 2 strings or arrays of characters and copy the strings.

⇒ Algorithm:

Step 1: Start

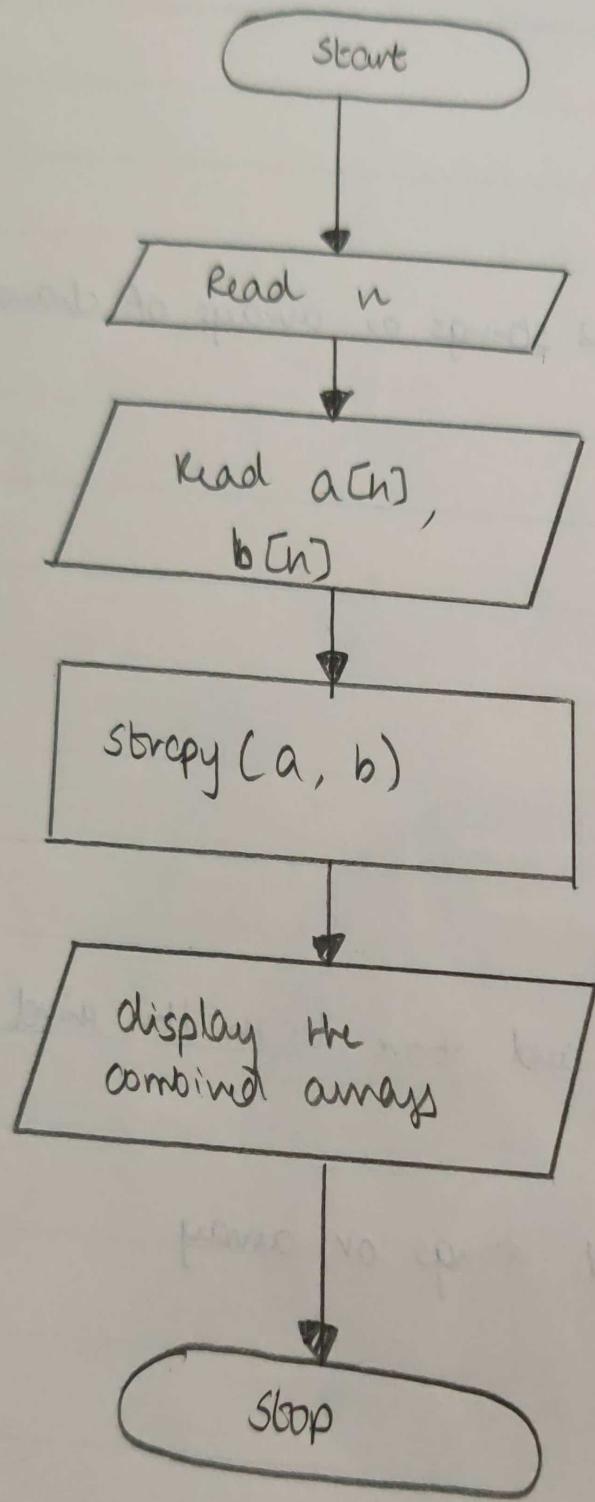
Step 2: Read n

Step 3: Read a[n], b[n]

Step 4: Copy the strings and store it in the third variable using strcpy()

Step 5: display the copied strings or array

Step 6: Stop



a) problem statement : Write a code to accept 2 strings and copy them.

Code:

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

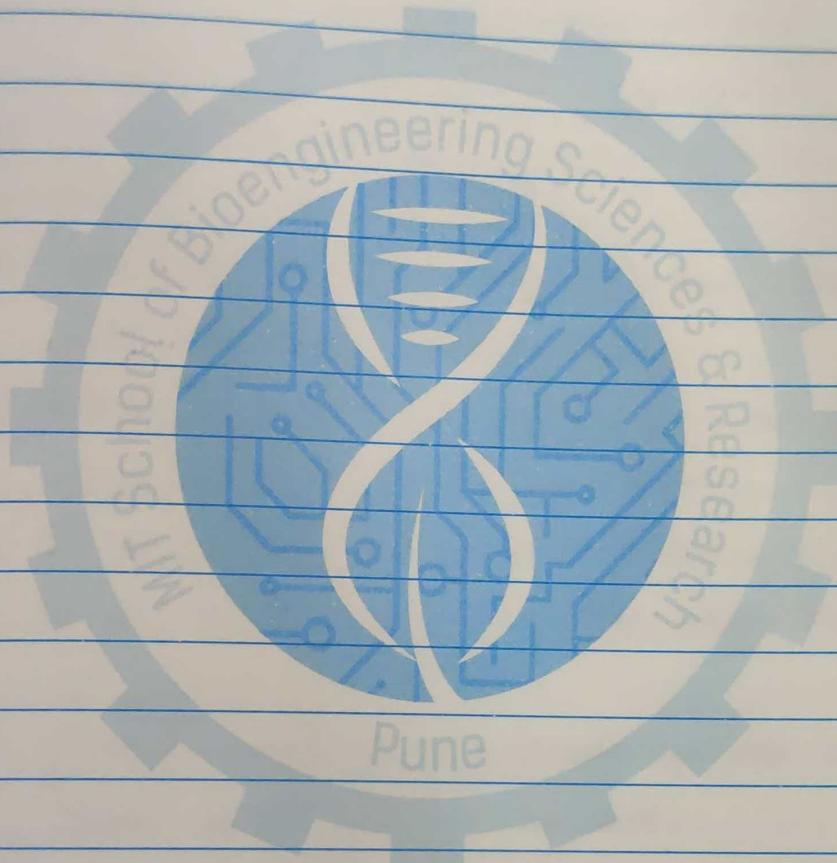
main()
{
    printf("\nAnushka surdas");
    printf("\nMITU22BTBI0016");
    char c1[20];
    char c2[20];
    char c3[20];
    printf("\nEnter your name : ");
    fgets(c1 , sizeof(c1) , stdin);
    c1[strcspn(c1, "\n")]=0;
    printf("\nEnter your name : ");
    fgets(c2 , sizeof(c2) , stdin);
    c2[strcspn(c2, "\n")]=0;
    printf("%s" , strcpy(c1,c2));
}
```

```
Anushka surdas
MITU22BTBI0016
Enter your name : cup

Enter your name : tea
tea
```

→ Conclusion:

- A program was successfully compiled and executed to accept 2 strings and combine them.
- For loop was used to accept and display the array of characters.
- <string.h> library was included to use string functions like strcpy(a, b)





## Exercise - 15

### → Problem Statement:

Write a program to create a user defined function to accept two numbers and display their sum, difference, product and quotient with return with argument

### → Algorithm:

Step 1: start

Step 2: Declare four functions to perform four arithmetic operation i.e., add, subtract, product, quotient

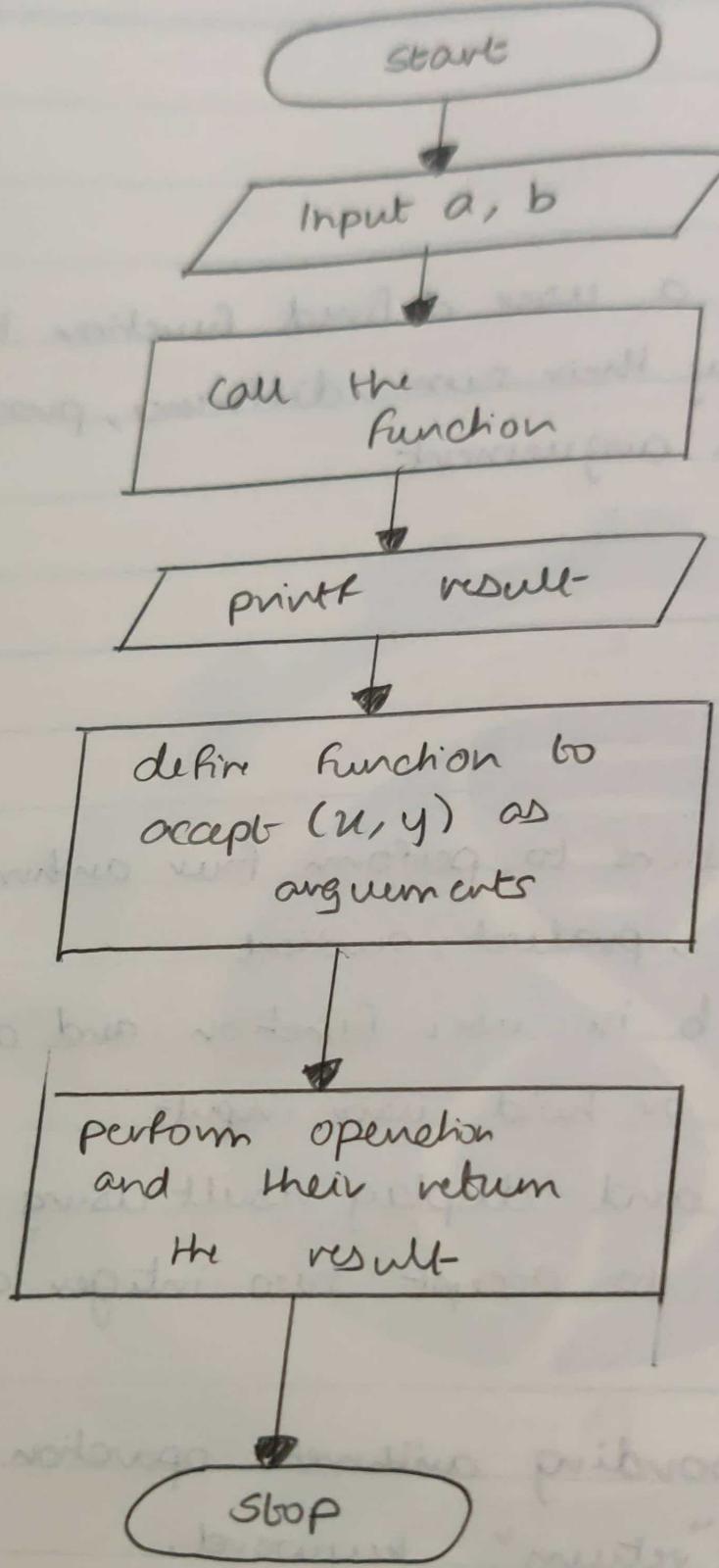
Step 3: Declare a and b in user function and accept their value from user or hold user input

Step 4: Call the function and display result using printf.

Step 5: declare function to accept two integer argument 'n' any 'y'

Step 6: perform corresponding arithmetic operation and return result using "return" keyword.

Step 7: stop



a) problem Statement: Write a user defined function code with return with arguments

Code:

```
#include<stdio.h>
//with return with arguments
int a,b;
int add(int,int);
int subtract(int,int);
int prod(int, int);
int quot(int,int); //function declaration
main()
{
    printf("Anushka surdas MITU22BTB10016");
    printf("\nEnter the first number");
    scanf("%d",&a);
    printf("\nEnter the second number");
    scanf("%d",&b);
    printf("\nAddition = %d", add(a,b));
    printf("\nSubtraction = %d", subtract(a,b));
    printf("\nProduct = %d", prod(a,b));
    printf("\nQuotient = %d", quot(a,b));
}

int add(int x, int y)      //function definition
{
    return x+y;
}

int subtract(int x, int y)
```

```
{  
    return x+y;  
}  
int prod(int x, int y)  
{  
    return x*y;  
}  
int quot(int x, int y)  
{  
    return x/y;  
}
```

Anushka surdas MITU22BTBI0016  
Enter the first number 5

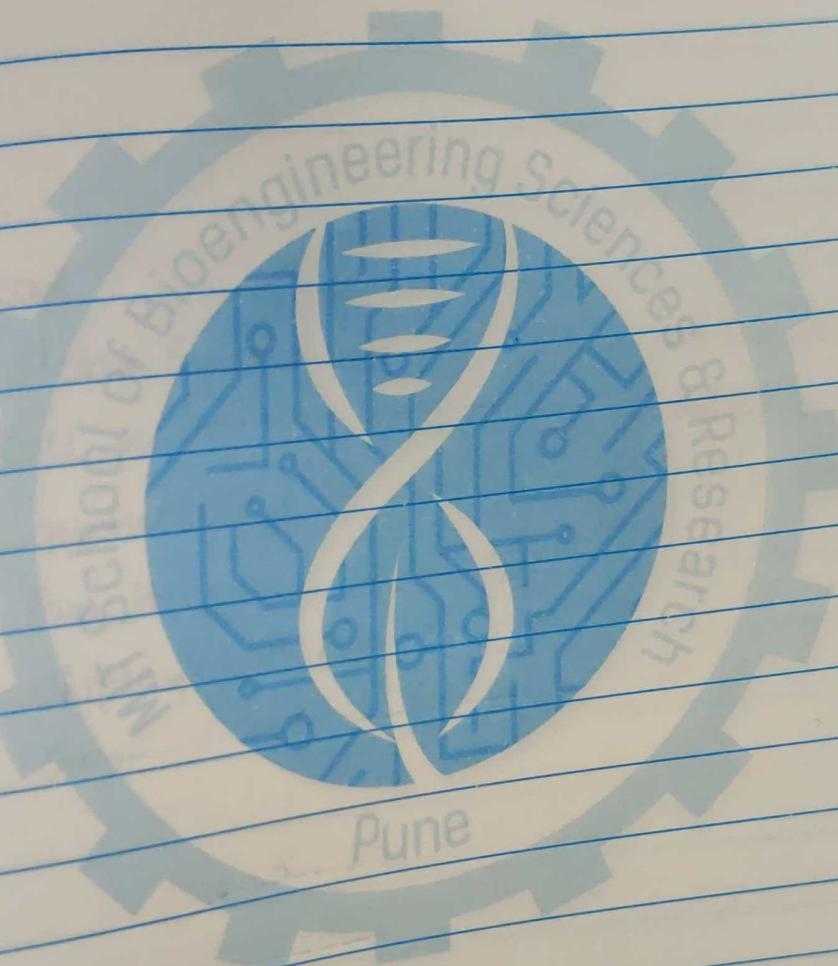
Enter the second number 4

Addition = 9  
Subtraction = 1  
Product = 20  
Quotient = 1



Conclusion:

A program was successfully created and compiled to create a user defined function to perform 4 arithmetic functions with return with argument.



### Exercise - 16

#### ⇒ Problem statement:

Write a program to create a user defined function to accept two numbers and display their sum, difference, product and quotient without return without arguments.

#### ⇒ Algorithm:

Step 1: Start

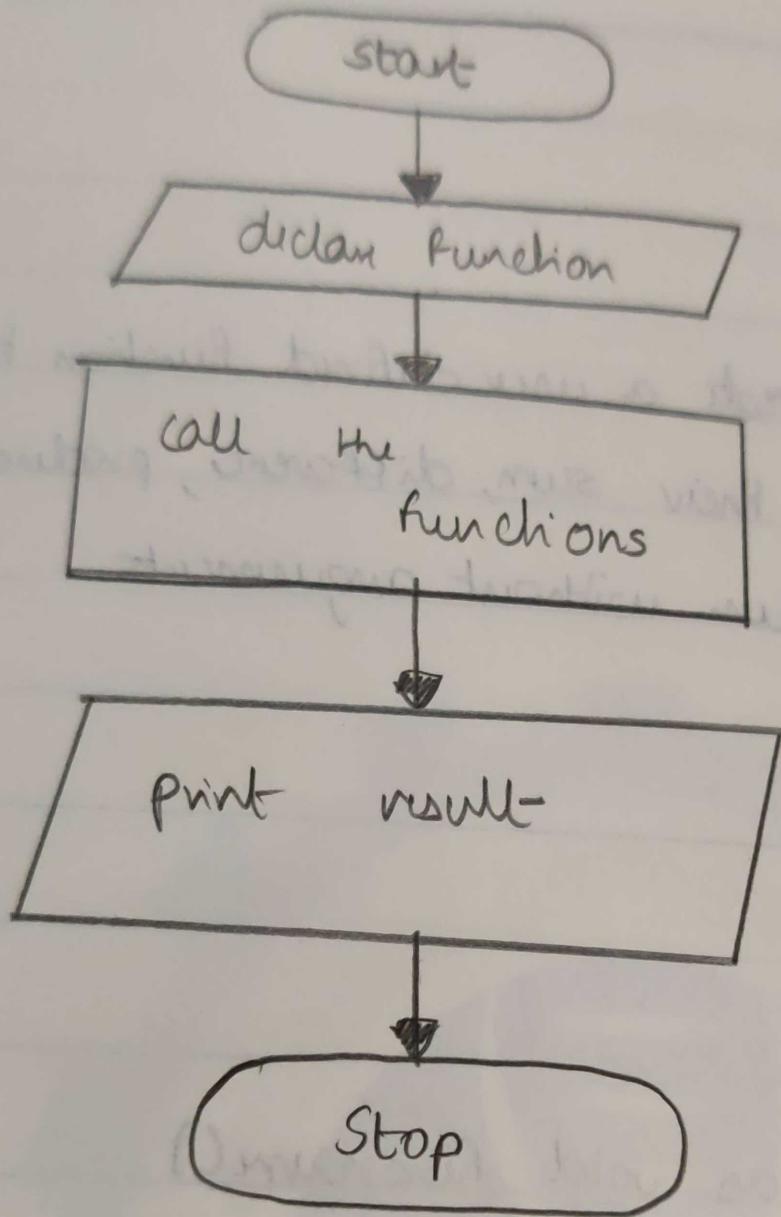
Step 2: declare function as void funcname()

Step 3: Call the function add(), subtract(), product(),  
quotient()

Step 4: Pass the values of variables

Step 5: display the result

Step 6: Stop



a) problem Statement: Write a user defined function code without return without arguments.

Code:

```
#include<stdio.h>
//without return without arguments
int a,b;      //global variables
void add();
void subtract();
void prod();
void quot(); //function declaration
main()
{
    printf("Anushka surdas MITU22BTBI0016");
    printf("\nEnter the first number");
    scanf("%d",&a);
    printf("\nEnter the second number");
    scanf("%d",&b);
    add();
    subtract();
    prod();
    quot();
}
void add() //function definition
{
    printf("\nAddition = %d", a+b);
}
void subtract()
```

```
{  
    printf("\nSubtraction = %d", a-b);
```

```
}
```

```
void prod()
```

```
{
```

```
    printf("\nProduct = %d", a*b);
```

```
}
```

```
void quot()
```

```
{
```

```
    printf("\nQuotient = %d", a/b);
```

```
}
```

```
Anushka surdas MITU22BTBI0016  
Enter the first number5
```

```
Enter the second number8
```

```
Addition = 13
```

```
Subtraction = -3
```

```
Product = 40
```

```
Quotient = 0
```

⇒ Conclusion:

→ A program was successfully compiled and executed to create a user defined function to accept two numbers and display their sum, difference, product and quotient without return without argument.



### Exercise - 17

#### ⇒ Problem statement:

write a program to accept create a user defined function to accept two numbers and display their sum, difference, product and quotient with return without argument.

#### ⇒ Algorithm:

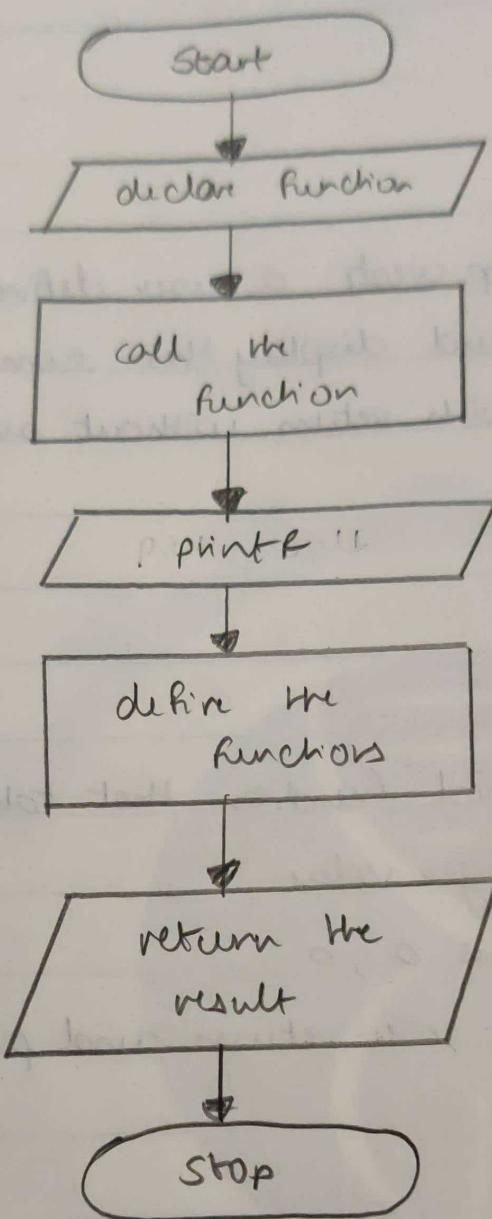
Step 1: start

Step 2: declare user defined function that take no arguments and return an integer value.

Step 3: declare 2 variables a, b

Step 4: call the function with return and print result-

Step 5: Stop



Q] problem Statement: Write a user defined function code without return without arguments.

Code:

```
#include<stdio.h>
//with return without arguments
int a,b;
int add();
int subtract();
int prod();
int quot(); //function declaration
main()
{
    printf("Anushka Surdas MITU22BTB10016");
    printf("\nEnter the first number");
    scanf("%d",&a);
    printf("\nEnter the second number");
    scanf("%d",&b);
    printf("\nAddition = %d", add());
    printf("\nSubtraction = %d", subtract());
    printf("\nProduct = %d", prod());
    printf("\nQuotient = %d", quot());
}

int add()      //function definition
{
    return a+b;
}
```

```
}  
int subtract()  
{  
    return a-b;  
}  
int prod()  
{  
    return a*b;  
}  
int quot()  
{  
    return a/b;  
}
```

Anushka Surdas MITU22BTBI0016  
Enter the first number8

Enter the second number2

Addition = 10

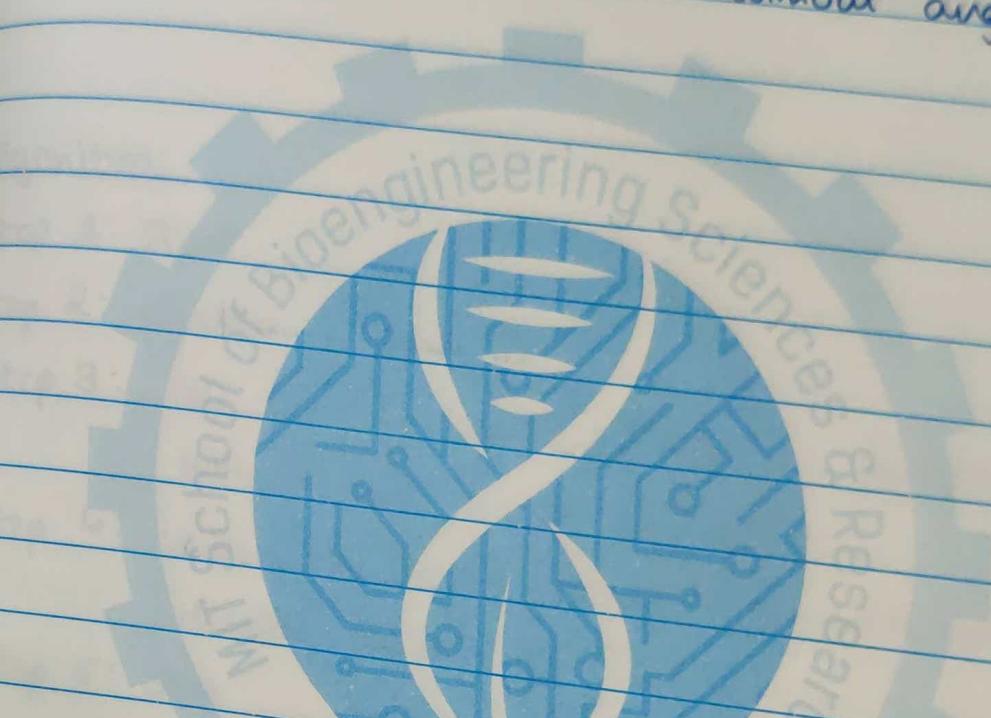
Subtraction = 6

Product = 16

Quotient = 4

Conclusion:

A program was successfully compiled and executed to create a user defined function to accept two numbers and display their sum, difference, product and quotient with with return without argument.





## Exercise - 18

### Problem Statement:

Write a program to create a user defined function to accept two numbers and display their sum, product, difference and quotient without return & with arguments

### Algorithm:

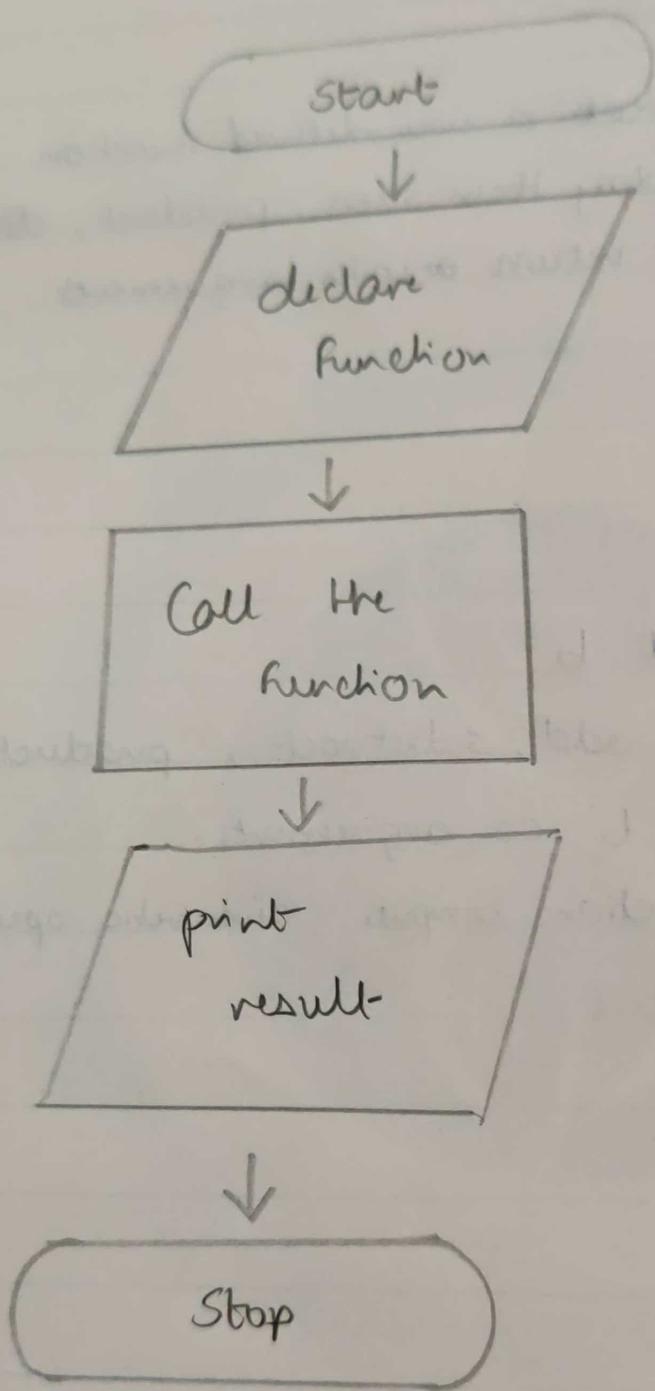
Step 1: Start

Step 2: declare a and b

Step 3: call function add, subtract, product, quotient with 'a' and 'b' as arguments.

Step 4: Within the function compute arithmetic operation and display result.

Step 5: Stop



Q) problem Statement: Write a user defined function code without return without arguments.

Code:

```
#include<stdio.h>
//without return with arguments
int a,b; //global variables
void add(int,int);
void subtract(int,int);
void prod(int,int);
void quot(int,int); //function declaration

main()
{
    printf("Anushka Surdas MITU22BTB10016");
    printf("\nEnter the first number");
    scanf("%d",&a);
    printf("\nEnter the second number");
    scanf("%d",&b);

    add(a,b);
    subtract(a,b);
    prod(a,b) ;
    quot(a,b);

    void add(int a, int b) //function definition
```

```
printf("\nAddition = %d", a+b);
```

```
} void subtract(int a, int b)
```

```
{ printf("\nSubtraction = %d", a-b);
```

```
} void prod(int a, int b)
```

```
{ printf("\nProduct = %d", a*b);
```

```
} void quot(int a, int b)
```

```
{ printf("\nQuotient = %d", a/b);
```

Anushka Surdas MITU22BTBI0016  
Enter the first number 6

Enter the second number 1

Addition = 7

Subtraction = 5

Product = 6

Quotient = 6

Conclusion:

A program was successfully compiled and executed to accept two numbers and display their sum, difference, product and quotient without return with arguments

