

**INDEX**

## PRACTICAL - 01

AIM: To study the use of different types of datatypes.

SOURCE CODE:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    char name[40];
    char add[40];
    int rollno;
    float percent;
    char grade;
    long int mob;
    clrscr();
    printf("----- Demonstrate various datatypes --");
    printf("Name of the student");
    scanf("%s", &name);
    printf("Address of student");
    scanf("%s", &add);
    printf("Roll no of student");
    scanf("%d", &rollno);
    printf("Percentage of student");
    scanf("%f", &percent);
```

FSO

```
printf("Grade of Student");
scanf("%s", &grade);
printf("Mobile No");
scanf("%d", &mob);
printf("In studentname : %s", name);
printf("\n Student address : %s", add);
printf("\n Student rollno : %d", roll_no);
printf("In student percent : %f", percent);
printf("In student grade : %c", grade);
printf("In student mobilenr : %d", mob);
getch();
```

}

OUTPUT :-

028

----- Demonstrate various datatypes -----

Name of Student :

Dharan

Address of Student :

India

Roll no of Student :

1701

Percentage of Student :

80.00

Grade Of Student :

A

Mobile No of ~~Student~~ :

1234567890

Student Name : Dharan

Student Address : India

Student Roll No : 1701

Percentage of Student : 80.00

Student Grade : A

Student Mobile No : 1234567890

29/11/19

850

## OUTPUT:

Enter 1<sup>st</sup> number : 20

Enter 2<sup>nd</sup> number : 8

Addition of 2 numbers : 28

Subtraction of 2 numbers : 12

Multiplication of 2 numbers : 160

Division of 2 numbers :  $\frac{200}{2} \ 3 = 25$

## PRACTICAL - 02

i) AIM: Write a C program which will show the output of various different types of operations.

SOURCE CODE:-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int num1, num2, add, sub, mul, div;
    clrscr();
    printf("Enter 1st Number");
    scanf("%d", &num1);
    printf("Enter 2nd Number");
    scanf("%d", &num2);
    add = num1 + num2;
    printf("Addition of 2 numbers : %d, add");
    sub = num1 - num2;
    printf("Subtraction of 2 numbers : %d, sub");
    mul = num1 * num2;
    printf("Multiplication of 2 numbers : %d, mul");
    div = num1 / num2;
    printf("Division of 2 numbers : %d, div");
    getch();
}
```

b) #include <stdio.h>  
#include <conio.h>  
void main()  
{  
 int a, b, big;  
 clrscr();  
 printf("Enter two numbers");  
 scanf("%d %d", &a, &b);  
 big = a > b ? a : b;  
 printf("The biggest num is %d", big);  
 getch();  
}

c) #include <stdio.h>  
#include <conio.h>  
void main()  
{  
 int x, y, z, val1, val2, val3, val4, val5;  
 clrscr();  
 printf("Enter 1<sup>st</sup> value");  
 scanf("%d", &x);  
 printf("Enter 2<sup>nd</sup> value");  
 scanf("%d", &y);  
 printf("Enter 3<sup>rd</sup> value");  
 scanf("%d", &z);  
 val1 = (x < y) & & (z > y);  
 printf("val1 is %d", val1);

OUTPUT :-

Enter two numbers : 9 19

The biggest number is : 19

OUTPUT :-

Enter 1<sup>st</sup> value : 9

Enter 2<sup>nd</sup> value : 8

Enter 3<sup>rd</sup> value : 2

value 1 is : 0

value 2 is : 1

value 3 is : 1

value 4 is : 0

value 5 is : 1

val2 = ( $x = y$ ) && ( $z \leq y$ );  
printf ("Value 2 is %d ", val2);  
val3 = ( $x < y$ ) || ( $z = y$ );  
printf ("Value 3 is %d ", val3);  
val4 = ! ( $x == y$ );  
printf ("Value 4 is %d ", val4);  
val5 = ( $x == y$ );  
printf ("Value 5 is %d ", val5);  
getch();  
}

Jai  
17/01/2020

## PRACTICAL - 03

AIM : Programs on decision statements.

- 1) WAP to find whether entered year is leap year or not.

Algorithm:

Step 1: Take integer variable year

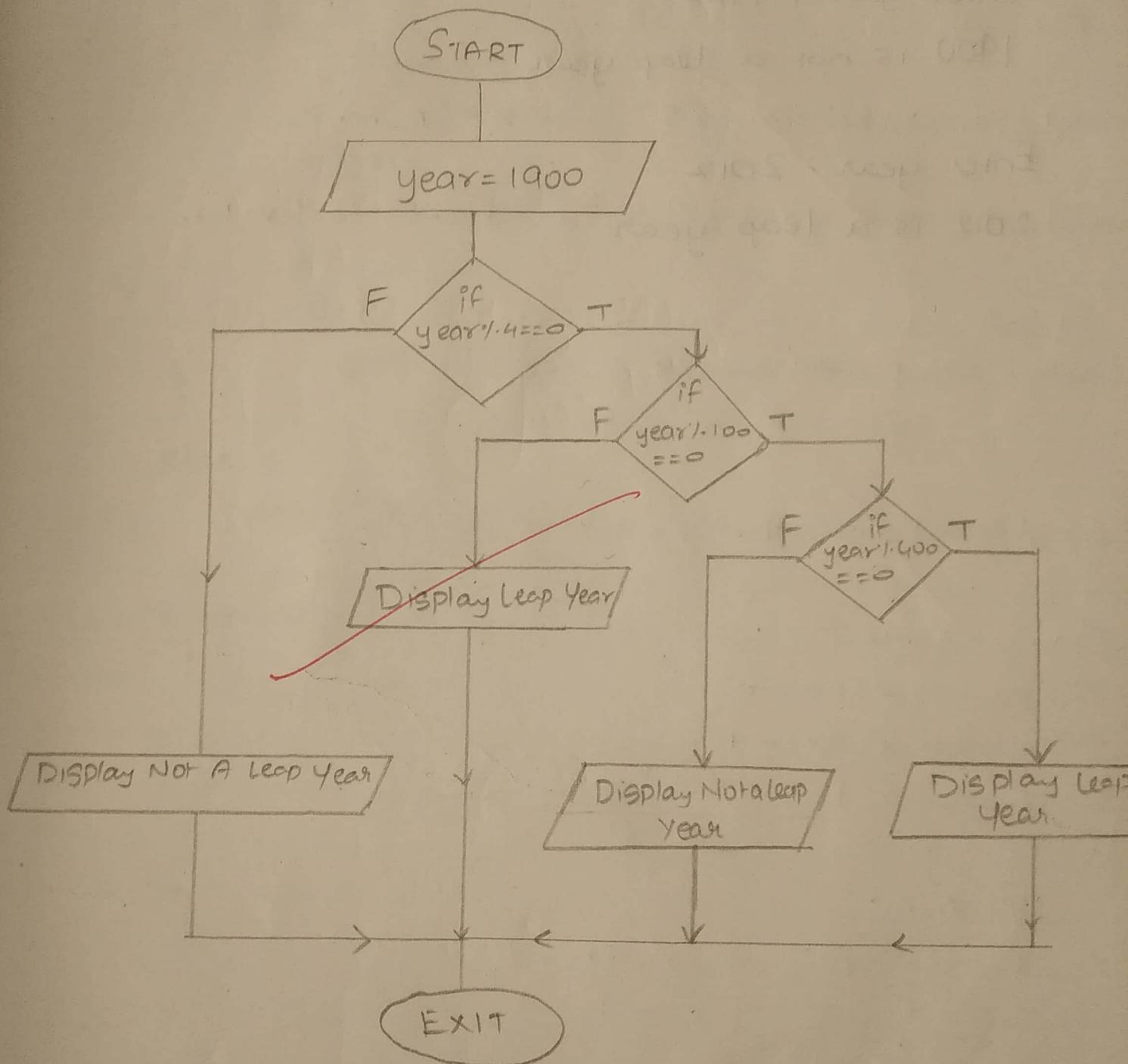
Step 2: Using user input take the value

Step 3: Using nested if else conditions, if the value is True, print it is a leap year or print it is not a leap year.

Source Code :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int year;
    clrscr();
    printf("Enter year : ");
    scanf("%d", &year);
    if (year % 4 == 0),
    {
```

Flowchart :-



580

OUTPUT:-

Enter year : 1900

1900 is not a leap year

Enter year : 2012

2012 is a leap year.

```
if (year % 100 == 0)
{
    if (year % 400 == 0)
        printf ("%d is a leap year", year);
    else
        printf ("%d is not a leap year", year);
}
else
    printf ("%d is not a leap year", year);

return 0;
```

2) WAP to find odd and even number.

Algorithm :

Step 1: Take integer variable number.

Step 2: Using user input, take the value.

Step 3: use conditional statement.

if (~~num~~ % 2 == 0) print even,  
else print odd.

Step 4: Print the result.

Source Code :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int num;
    printf("Enter the number : ");
    scanf("%d", &num);
    if (num % 2 == 0)
        printf("%d is an even number", num);
    else
        printf("%d is an odd number", num);
}
```

Output:-

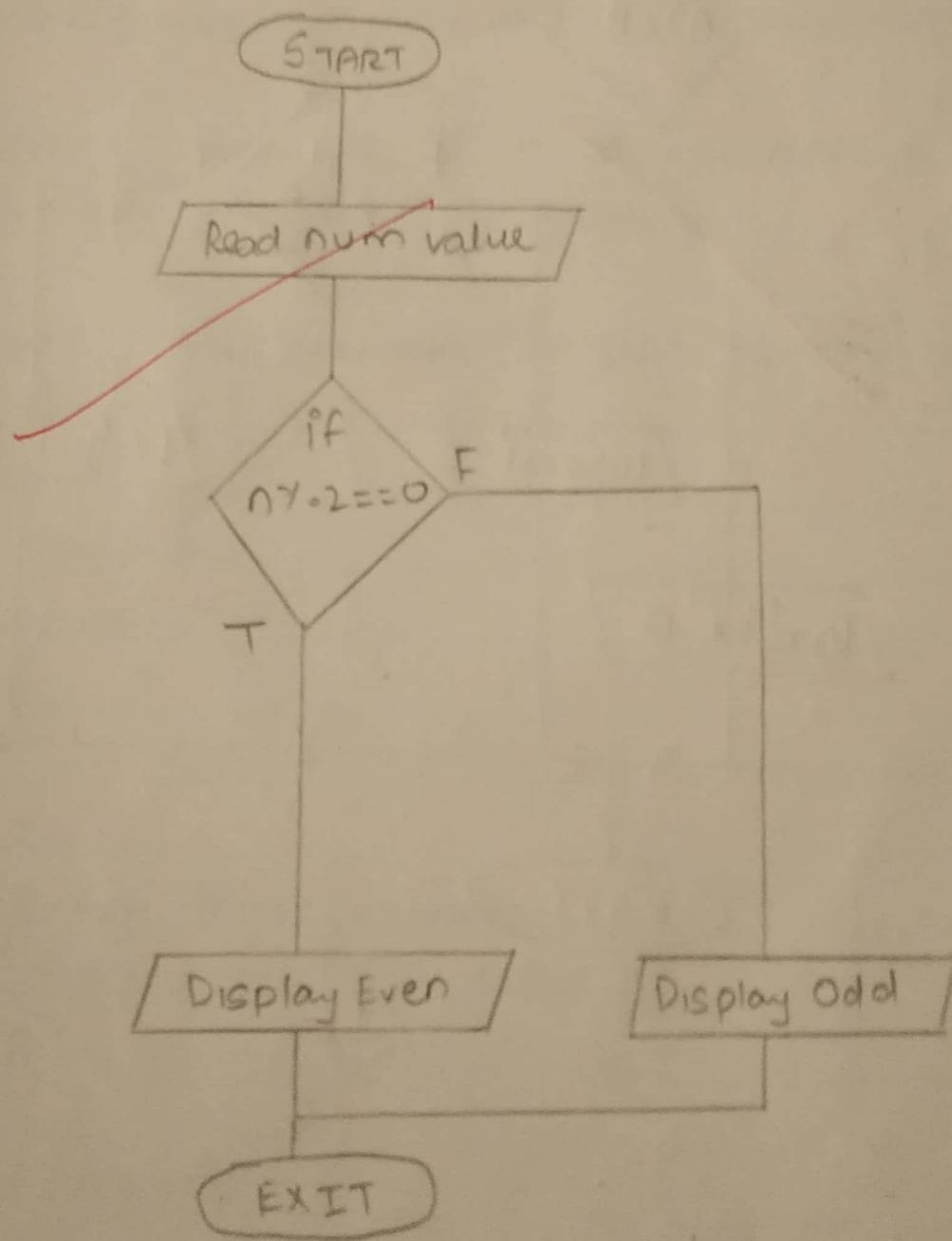
Enter the number :- 7

7 is an odd number

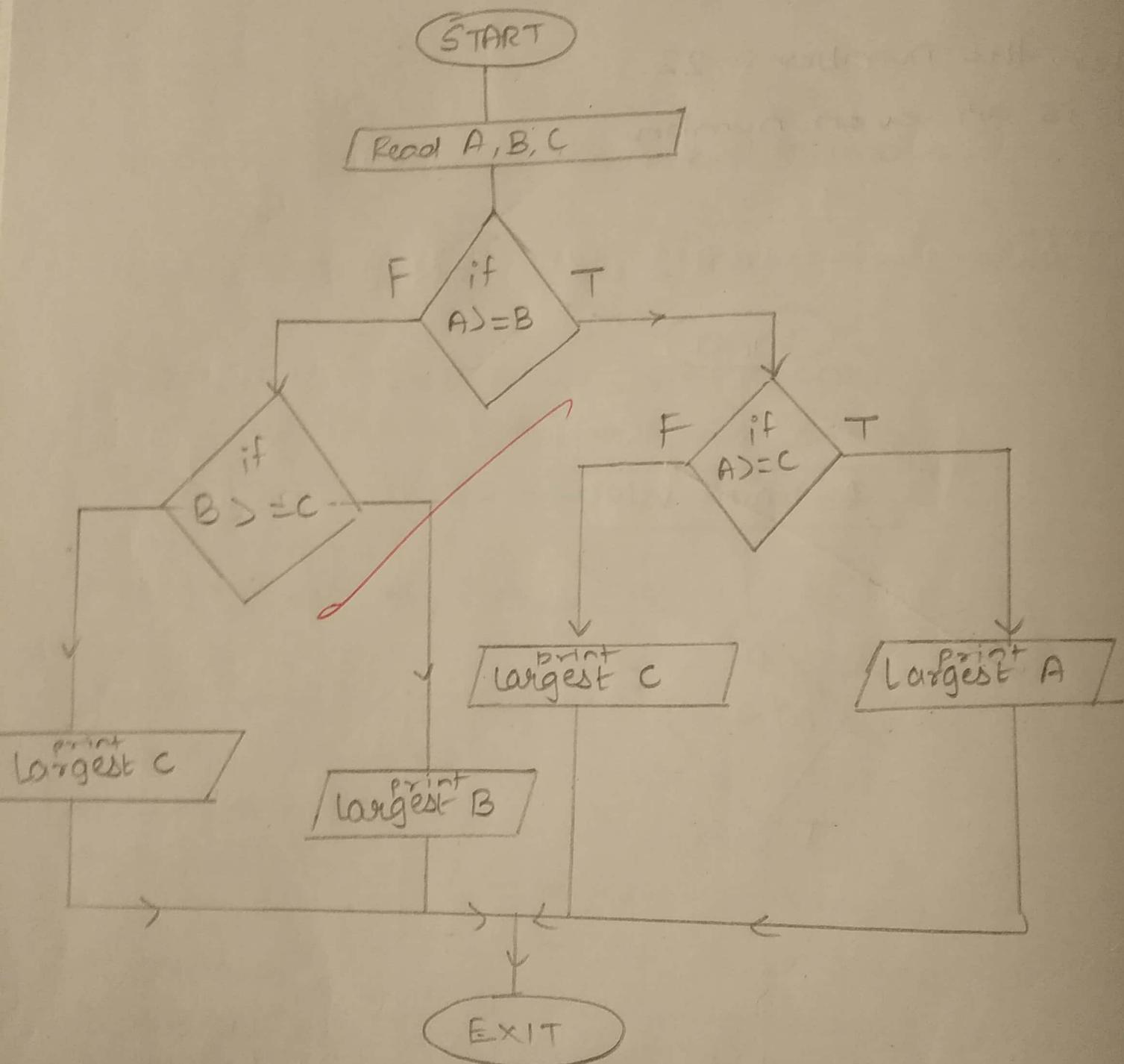
Enter the number :- 22

22 is an even number

Algorithm:-



## Algorithm:-



3. WAP to find largest of three numbers using nested if...else.

Algorithm:

Step 1: Take the three variables A, B, C.

Step 2: Using user input, take the values.

Step 3: Using nested if-else statement determine which number is greater.

Step 4: print the ~~great~~ largest number.

Source code:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int A,B,C;
    printf("Enter three numbers");
    scanf("%d%d%d", &A, &B, &C);
    if (A>=B)
    {
        if (A>=C)
            printf("%.d is the largest number", A);
        else
            printf("%.d is the largest number", C);
    }
}
```

```
else  
{
```

```
    if (B >= C)
```

~~printf(".\n.d is the largest number", B);~~

```
    else
```

~~printf(".\n.d is the largest number", C);~~

```
return 0;
```

```
}
```

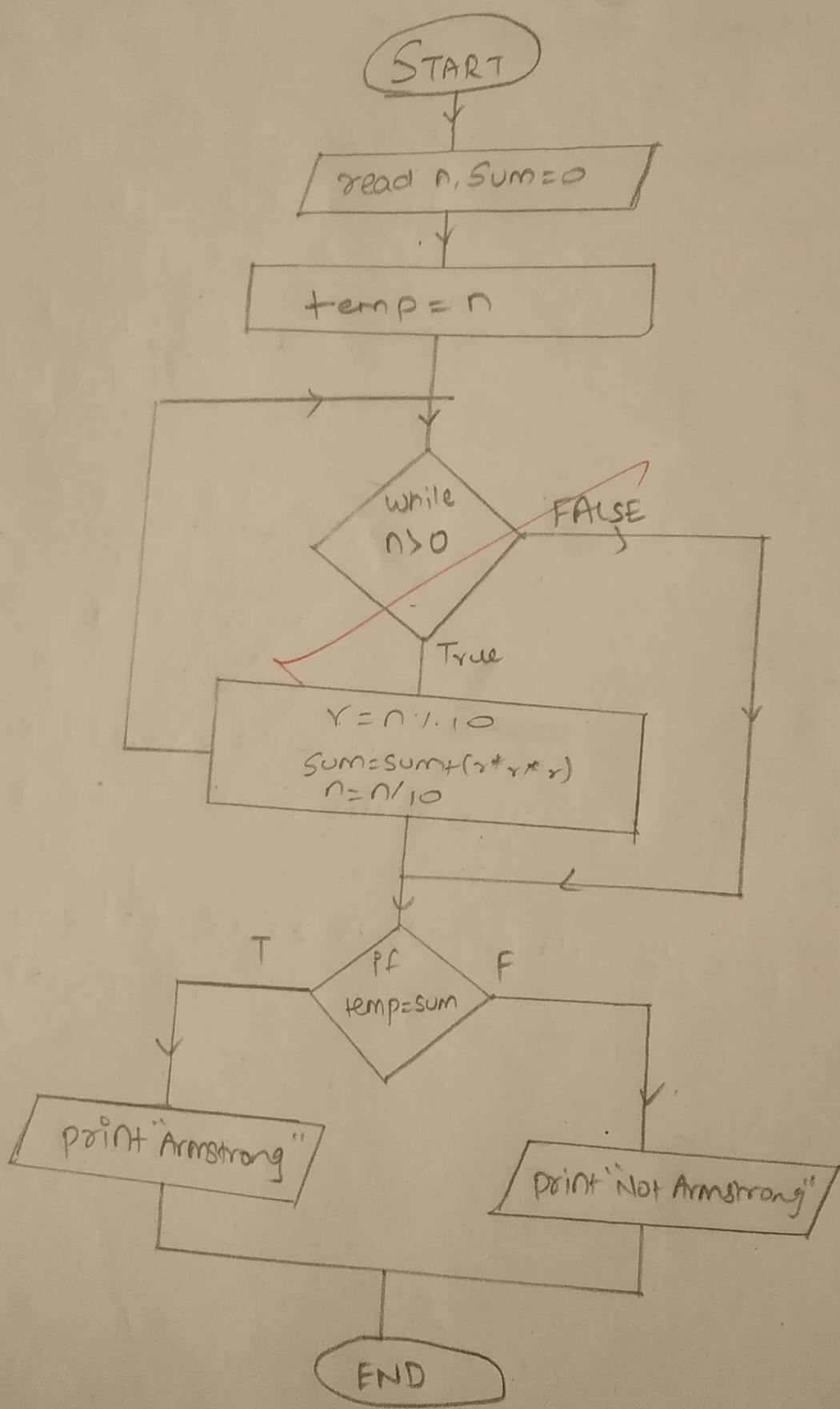
Output :-

Enter the numbers: 2 8 1

8 is the largest number

Jm  
29/01/2020

# Flowchart:-



## PRACTICAL-4

TOPIC: Programs on Looping

Aim: Check whether the given number is Armstrong or not?

Algorithm:

Step 1: Take the integer variable 'n' and using user input take the value.

Step 2: Store the number in temp variable too.

Using while loop check,  $r=n \% 10$  and get the sum of individual numbers as,  
 $Sum = Sum + (r * r * r)$  and  $n = n / 10$ .

Step 3: If the temp number and sum are equal, print result is Armstrong number. Or print It is not an armstrong number.

Step 4: Or return 0.

Source Code :-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n, r, sum=0, temp;
    printf ("Enter Number");
    scanf ("%d", &n);
    temp=n;
    while (n>0)
    {
        r=n % 10;
        Sum=Sum+(r*r*r);
        n=n/10;
    }
    if (temp==sum)
        printf ("It is an Armstrong Number");
    else
        printf ("It is not an Armstrong Number");
    return 0;
}
```

037

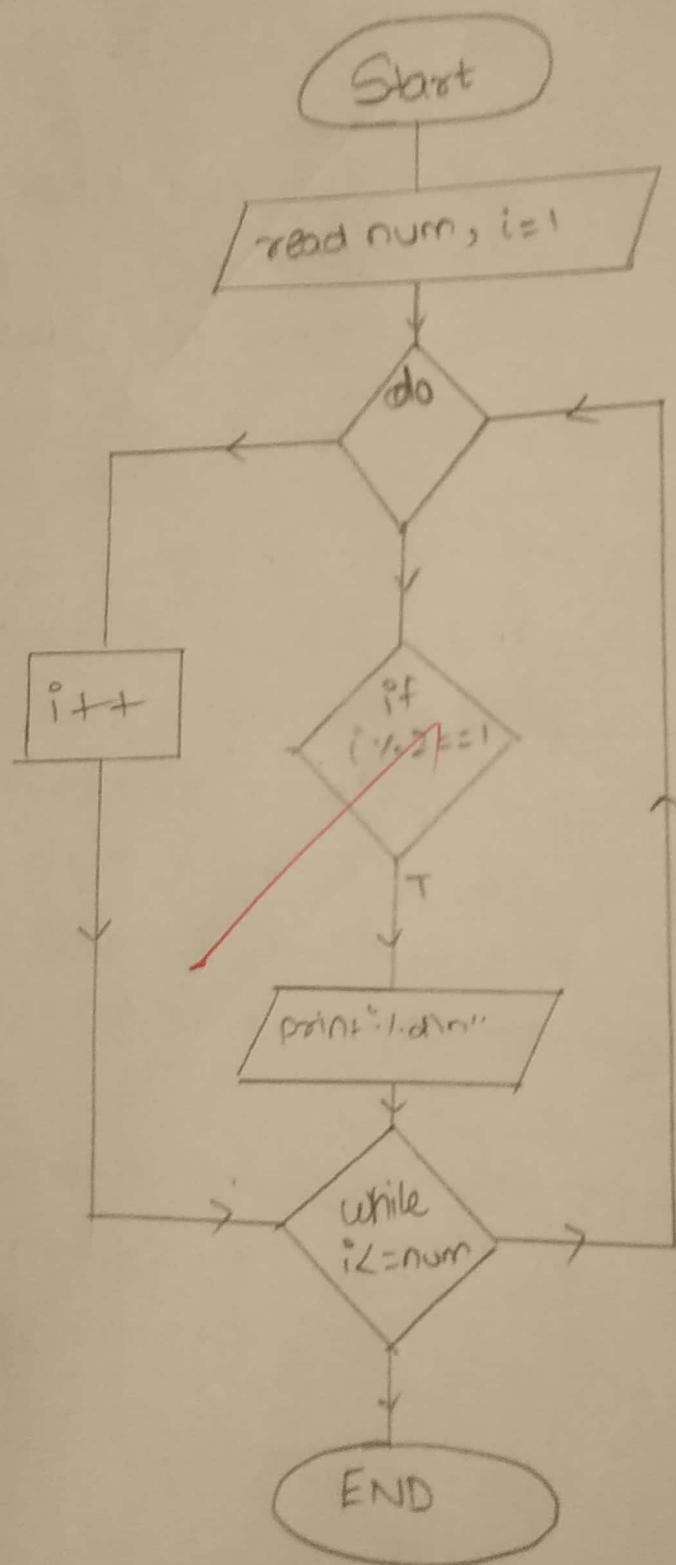
Output :-

Enter Number: 153

It is an Armstrong Number

780

Flowchart:-



2] WAP to print odd numbers bet" 1 to 50 using do-while loop.

Algorithm :-

Step 1 : Take the variable int i and int num and using user input take the value until which you want to print odd numbers.

Step 2 : initialize i=1 and using do<sup>while</sup> loop, check condition for odd number and print all the odd numbers which satisfy the condition

Step 3 : Print the result.

Source Code :-

```
#include <stdio.h>
#include <conio.h>
void main()
{
    int i, num;
    clrscr();
    printf("Enter number until which you want to print odd
           numbers");
    scanf("%d", &num);
    i = 1;
    do
    {
```

```
if (i % 2 == 1)
{
    printf("%d\n", i);
}
```

```
i++;
}
```

```
while (i <= num);
```

```
getch();
```

```
}
```

Output :-

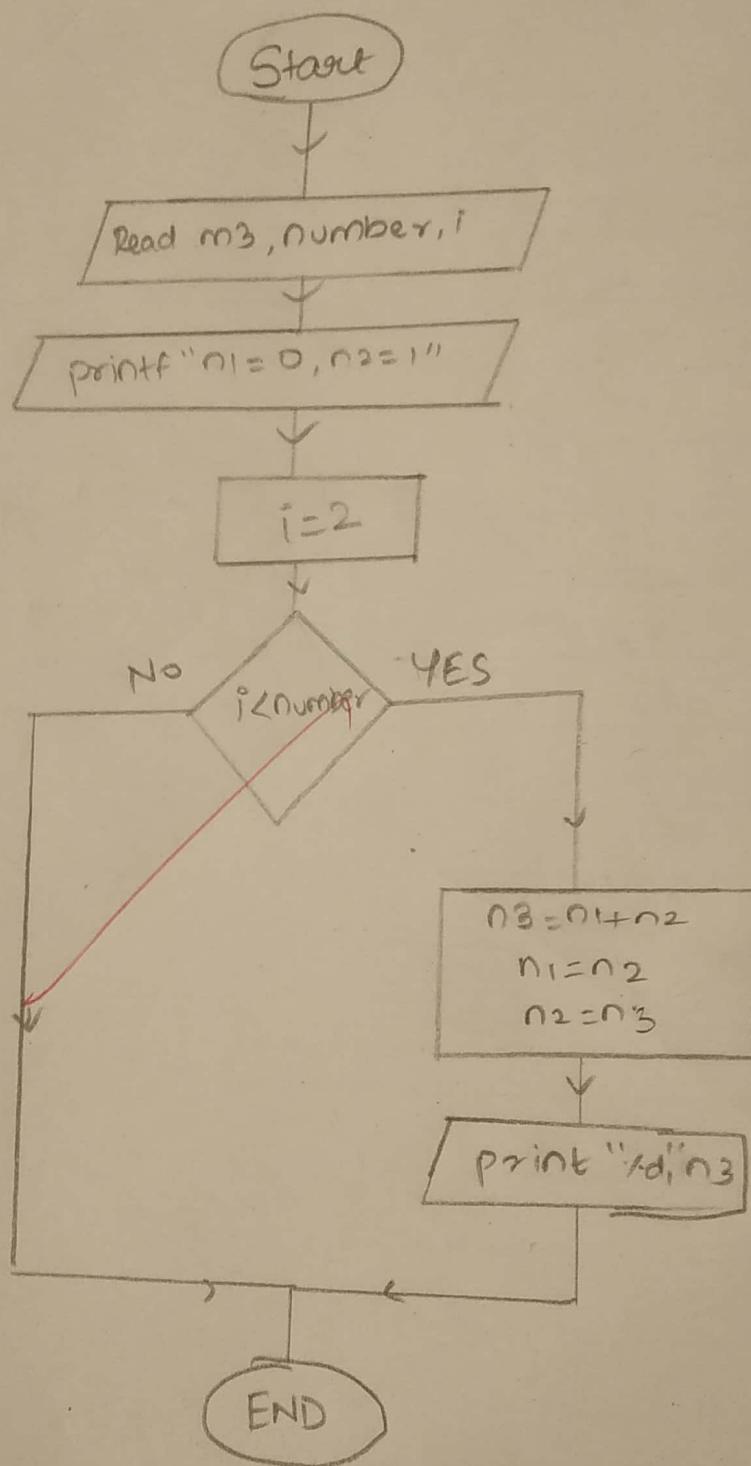
039

Print all odd numbers until 50

1  
3  
5  
7  
9  
11  
13  
15  
17  
19  
21  
23  
25  
27  
29  
31  
33  
35  
37  
39  
41  
43  
45  
47  
49



Flowchart :-



3] WAP to find Fibonacci Series.

Algorithm :-

Step1: Take variable int number and take the user input.

Step2: Using for loop initialize and update the expression.

Step3: print the loop.

Source Code:-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int n1=0, n2=1, n3, i, number;
    clrscr();
    printf("Enter Number of elements");
    scanf("%d", &number);
    printf("\n%d %d", n1, n2);
    for (i=2; i<number; i++)
    {
        n3 = n1 + n2;
        printf(" %d", n3);
        n1 = n2;
```

```
n2=n3;  
}  
getch();  
return 0;  
}
```

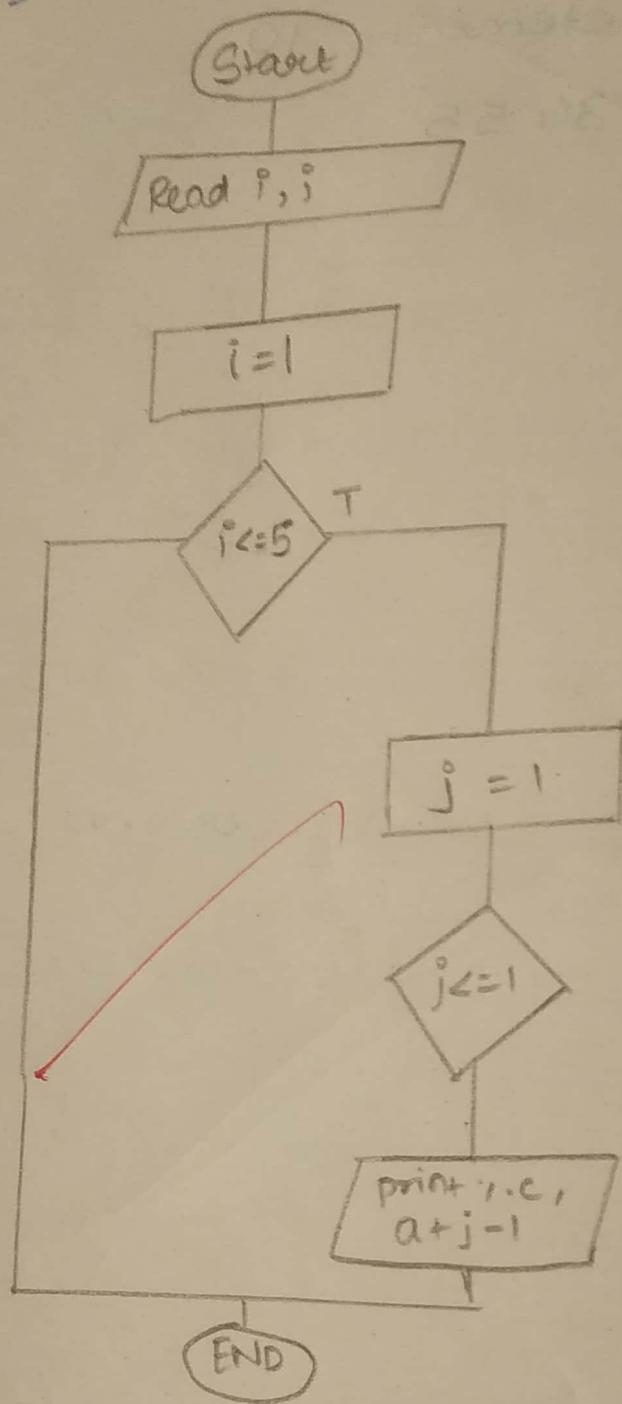
Output:

040

Enter number of elements: 10

0 1 1 2 3 5 8 13 21 34 55

Flowchart :-



Output:

a  
a b  
a b c  
a b c d  
a b c d e

4] Pattern:-

a  
a b  
a b c  
a b c d  
a b c d e

Algorithm:-

Step1: Take variable i and j as int.

Step2: Using ~~for~~ for loop, initialize and update the expressions.

~~Step3:~~ print the output.

Source code:-

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int i,j;
    for (i=1; i<=5; i++)
    {
        for (j=1; j<=i; j++)
        {
            printf("%c", 'a'+j-1);
        }
        printf("\n");
    }
}
```

*Jrvi*  
07/02/2020

## PRACTICAL-05

### TOPIC: PROGRAMS ON ARRAY

i) WAP to find the largest of three number in array

Algorithm:

Step 1: Take an array A and define its value

Step 2: Declare largest as integer

Step 3: Set largest to 0.

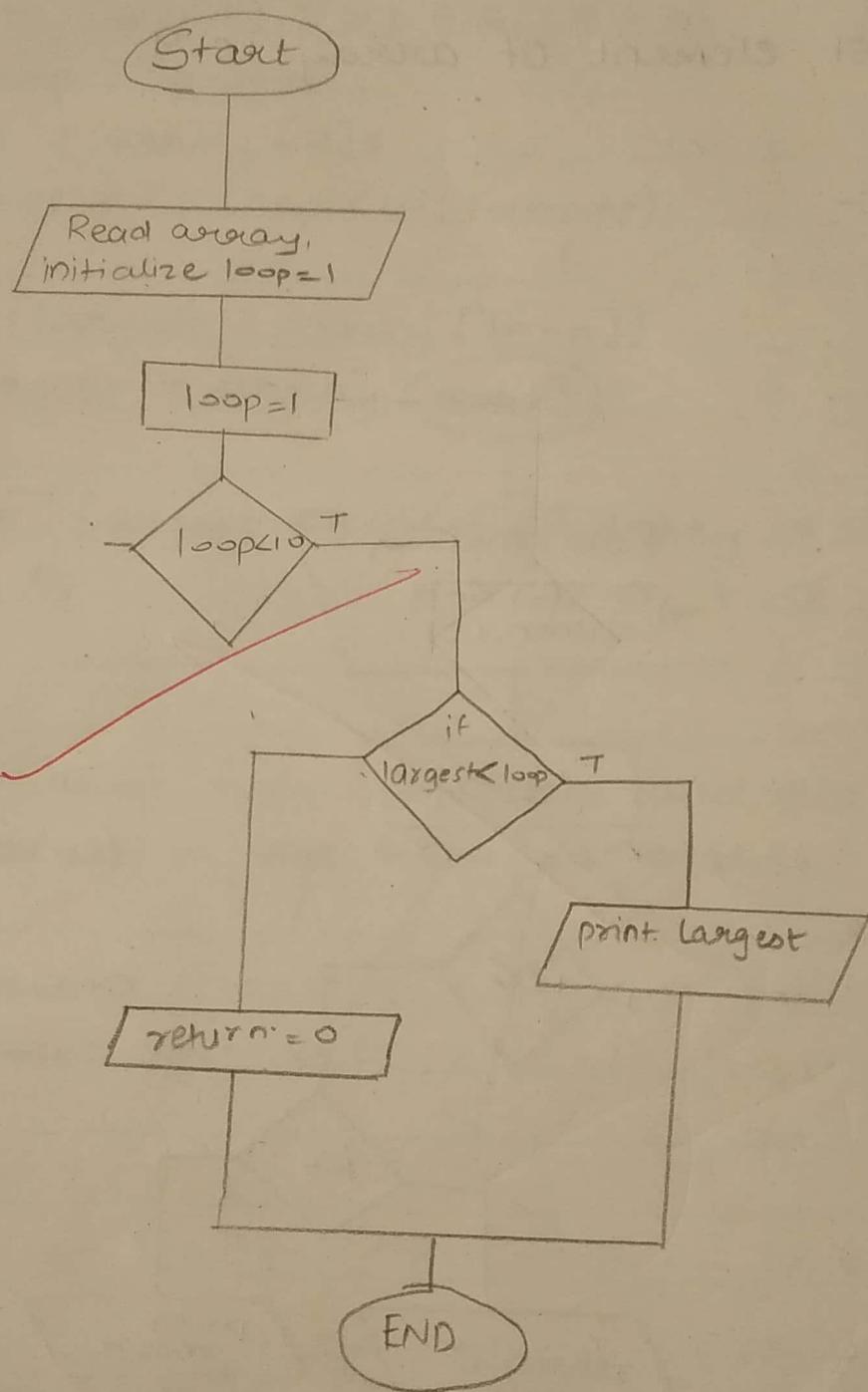
Step 4: Loop for each value of A

Step 5: If  $A[n] > \text{largest}$ , Assign  $A[n]$  to largest

Step 6: After loop finishes, Display largest as largest element of array

Source code:-

```
#include <stdio.h>
#include<conio.h>
int main()
{
```

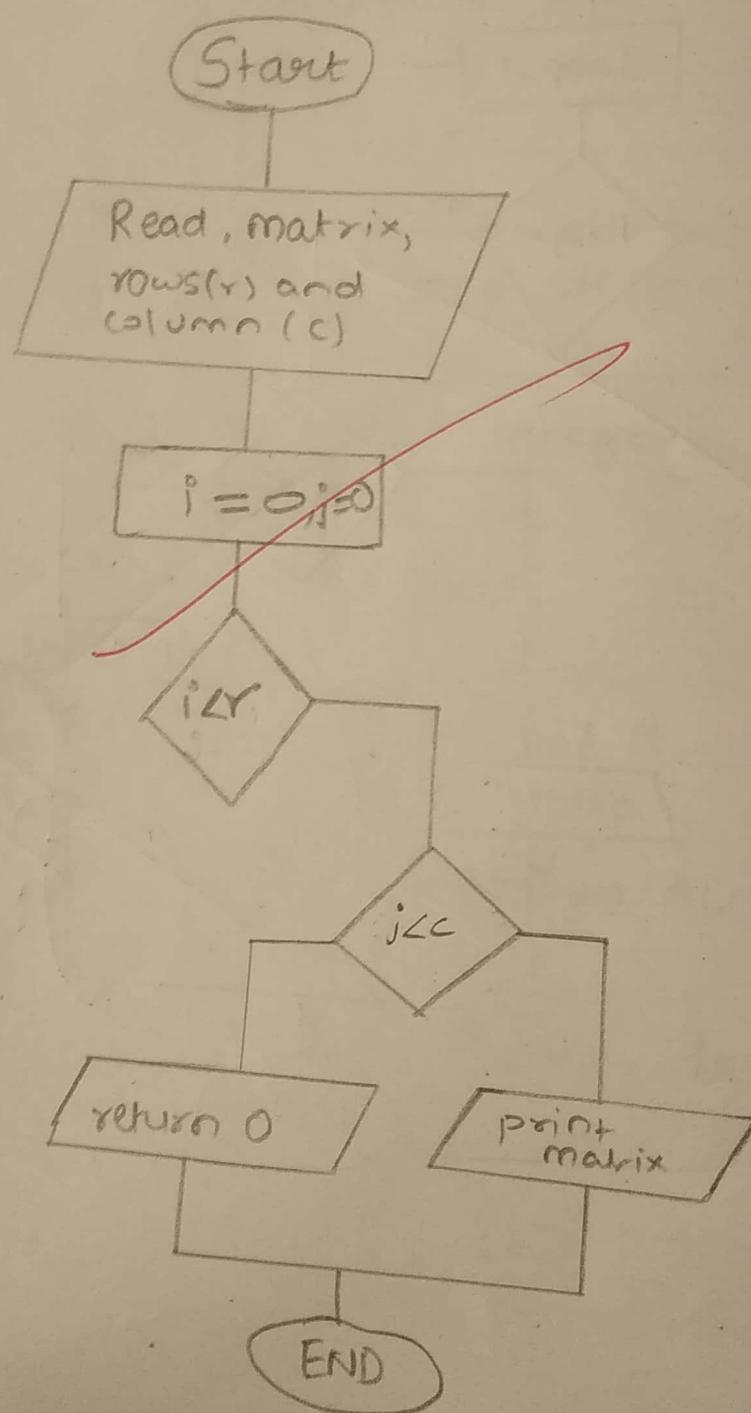


Q40

Output :-

Largest element of array is 9.

Flowchart :-



```

int array[10] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 0};
int loop, largest;
largest = array[0];
for (loop = 1; loop < 10; loop++)
{
    if (largest < array[loop])
        largest = array[loop];
}
printf("Largest element of array is %.d", largest);
return 0;
}

```

WAP to take values from the user and display the taken values in the form of Matrix.

```

#include <stdio.h>
#include <conio.h>
int main()
{
    int matrix;
    int i, j, r, c;
    printf("Enter Rows");
    scanf("%d", &r);
    printf("Enter Columns");
    scanf("%d", &c);
    printf("Enter matrix elements");
    for (i = 0; i < r; i++)
    {

```

```
for(j=0;j<c;j++)
{
    printf ("\n%d\n",matrix[i][j]);
}
printf ("\n");
}
return 0;
}
```

- 3] WAP to perform matrix addition.

Algorithm:

Step 1: Define integers and arrays for storing.

Step 2: User input no. of rows and columns  
input the two matrices user using  
loop.

Step 3: Add the two matrices and store it in  
array.

Step 4: Display the added matrix.

Output:

Enter number of rows: 3

Enter number of columns: 3

Enter matrix elements:

Enter element [1,1] : 1

Enter element [1,2] : 1

Enter element [1,3] : 1

Enter element [2,1] : 2

Enter element [2,2] : 2

Enter element [2,3] : 2

Enter element [3,1] : 3

Enter element [3,2] : 3

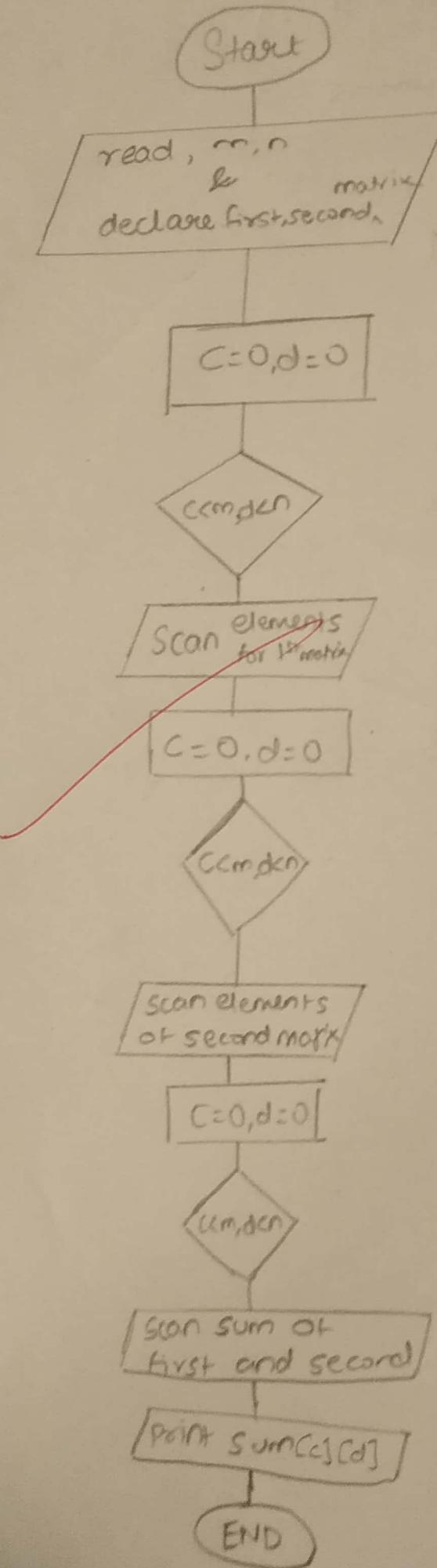
Enter element [3,3] : 3

Matrix is :

1 1 1

2 2 2

3 3 3



Source Code:

```
#include <stdio.h>
#include <conio.h>
int main()
{
    int m,n,c,d, first [10][10], second [10][10], sum[10][10];
    printf ("Enter the number of rows and columns");
    scanf ("%d %d", &m, &n);
    printf ("Enter the elements of first matrix\n");
    for (c=0; c<m; c++)
    {
        for (d=0; d<n; d++)
        {
            scanf ("%d", &first [c][d]);
        }
    }
    printf ("Enter the elements of second matrix\n");
    for (c=0; c<m; c++)
    {
        for (d=0; d<n; d++)
        {
            scanf ("%d", &second [c][d]);
        }
    }
    printf ("Sum of entered matrices:\n");
    for (c=0; c<m; c++)
    {
        for (d=0; d<n; d++)
        {

```

```
scanf[c][d] = first[c][d] + second[c][d];  
printf("%d\n", sum[c][d]);  
}  
printf("\n");  
return 0;
```

Output:-

Enter the number of rows and columns

2  
2

Enter the elements of first matrix

1 2  
3 4

Enter the elements of second matrix

5 6  
2 1

Sum of matrices:

6 8  
5 5

~~Sum:  
11 13  
7 6~~