**LAB SHEET NO.3 [To be familiar with select structure (branching)]**

Q1.WAP to check whether a number is negative, positive or zero.

**ALGORITHM:**

Step1: Start

Step2: Declare the number

Step3: Read the number num

Step4: if Num>0

Output (“Positive)

else if Num<0

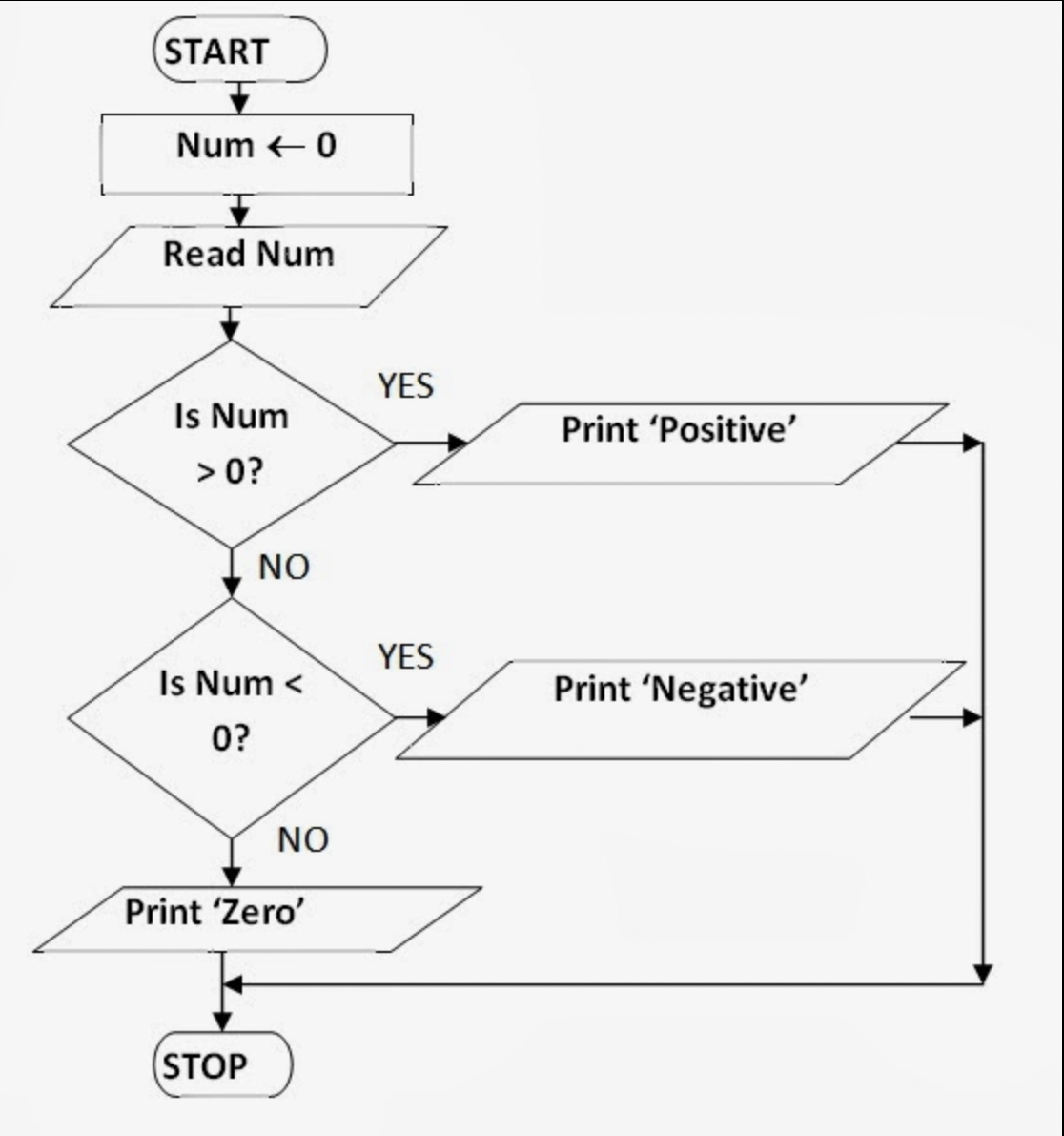
Output (“Negative”)

else Num=0

Output (“Zero”)

Step5: Stop

**FLOWCHART:**



**CODE:**

#include <stdio.h>

**int** main() {

**double** num;

printf("Enter a number: ");

scanf("%lf", &num);

**if** (num <= 0.0) {

**if** (num == 0.0)

printf("You entered 0.\n");

**else**

printf("You entered a negative number.\n");

}

**else**

printf("You entered a positive number.\n");

**return** 0;

}

**OUTPUT:**

Enter a number: 0

You entered 0.

Program ended with exit code: 0

Q2.WAP to find maximum between three numbers entered by user.

**ALGORITHM:**

Step1: Start

Step2: Ask the user to enter three integer values.

Step3: Read the three integer values in A, B, and C (integer variables).

Step4: Check if A is greater than B.

Step5: If true, then check if A is greater than C.

If true, then print A as the greatest number.

If false, then print C as the greatest number.

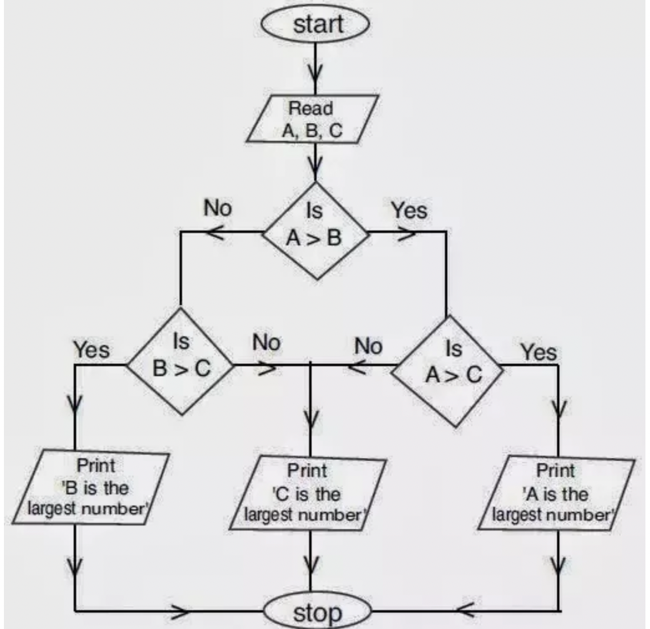
Step6: If false, then check if B is greater than C.

If true, then print B as the greatest number.

If false, then print C as the greatest number.

Step7: Stop

**FLOWCHART:**



**CODE:**

#include <stdio.h>

**int** main()

{

**int** A, B, C, max;

/\* Input three numbers from user \*/

printf("Enter three numbers: ");

scanf("%d%d%d", &A, &B, &C);

**if**((A > B) && (A > C))

{

/\* If num1 is greater than both \*/

max = A;

}

**else** **if**((B > A) && (B > C))

{

/\* If num2 is greater than both \*/

max = B;

}

**else** **if**((C > A) && (C > B))

{

/\* If num3 is greater than both \*/

max = C;

}

/\* Print maximum number \*/

printf("Maximum among all three numbers = %d\n", max);

**return** 0;

}

**OUTPUT:**

Enter three numbers: 3 6 9 //User Defined

Maximum among all three numbers = 9

Program ended with exit code: 0

**Q3.WAP to input a character from user and check whether the character is vowel or consonant.**

**ALGORITHM:**

Step1: Start

Step2: Declare character type variable ch

Step3: Read ch from user

Step4: if (ch=a|| ch=A|| ch=e|| ch=E|| ch=i|| ch=I|| ch=o||

ch=O|| ch=u|| ch=U)

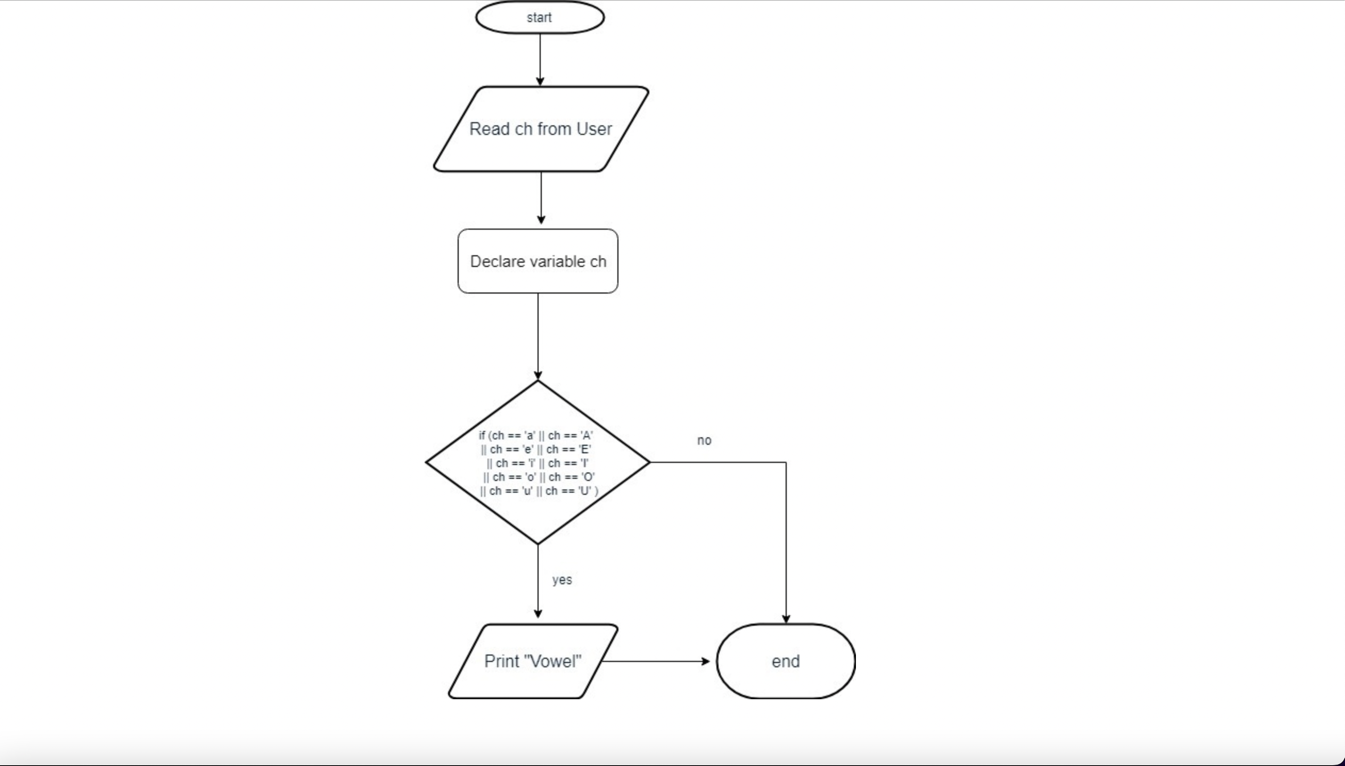
Print (“Vowel”)

else

Print (“Consonant”)

Step5: Stop

**FLOWCHART:**



**CODE:**

#include<stdio.h>

**int** main()

{

**char** ch;

printf("Please Enter an alphabet: ");

scanf(" %c", &ch);

**if**(ch == 'a' || ch == 'e' || ch == 'i' || ch == 'o' || ch == 'u' ||

ch == 'A' || ch == 'E' || ch == 'I' || ch == 'O' || ch == 'U') {

printf("%c is a VOWEL.\n", ch);

}

**else** {

printf("%c is a CONSONANT.\n", ch);

}

**return** 0;

}

**OUTPUT:**

Please Enter an alphabet: r //User Defined

r is a CONSONANT.

Program ended with exit code: 0

Q4.WAP to input a character from user and check whether the character is Alphabet or not. If the character is Alphabet then show whether it is uppercase or lowercase.

**ALGORITHM:**

Step1: Start

Step2: Declare character type variable ch

Step3: Read ch from user

Step4: if (ch>=’A’ && ch<=’Z’)

Print (“uppercase alphabet”)

elseif (ch>=’a’ && ch<=’z’)

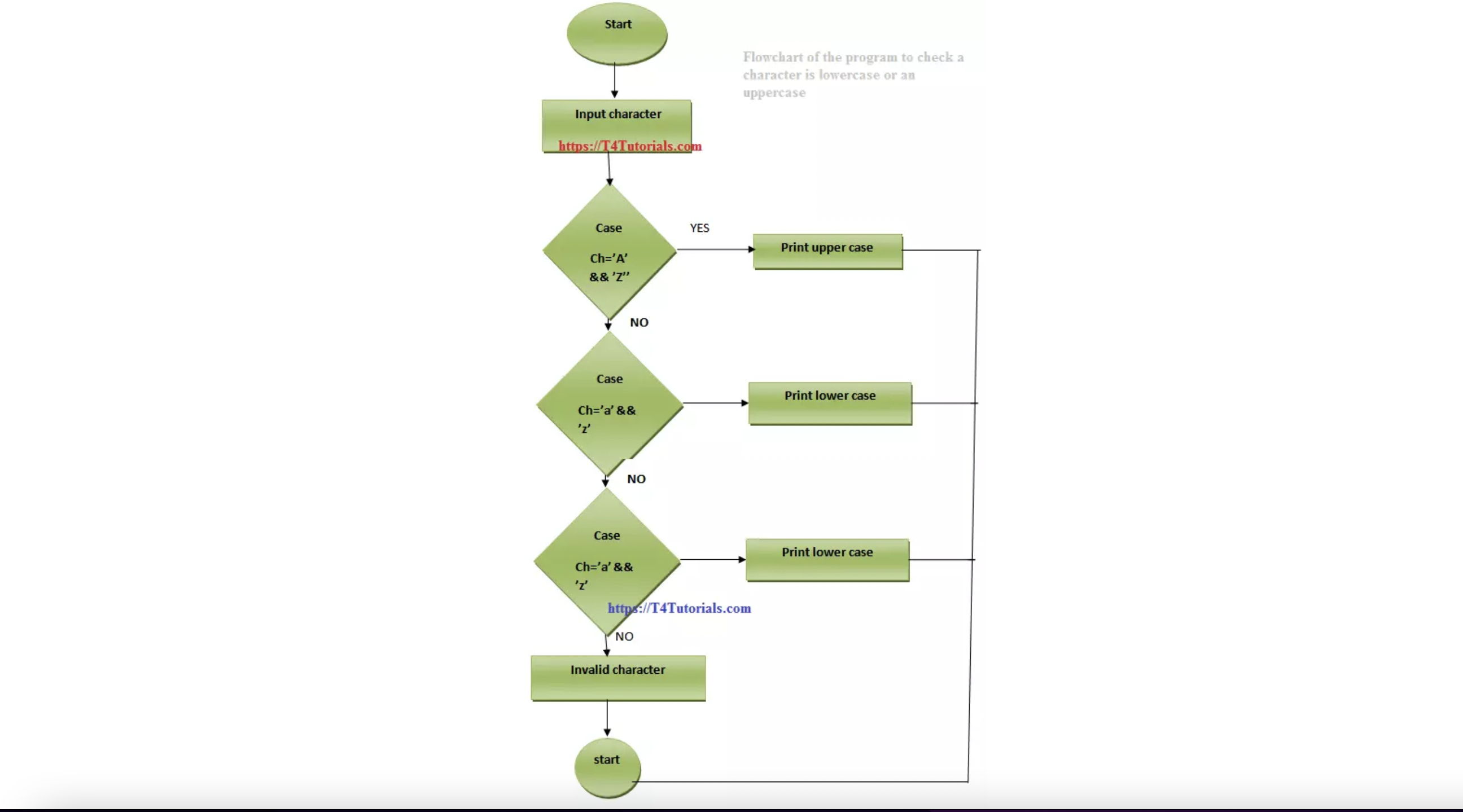
Print (“Lowercase alphabet”)

else

Print (“Not an alphabet”)

Step5: Stop

**FLOWCHART:**



**CODE:**

#include <stdio.h>

**int** main()

{

**char** ch;

printf("Enter any character: ");

scanf("%c", &ch);

**if**((ch >= 'A' && ch <= 'Z') || (ch >= 'a' && ch <= 'z'))

{

**if**(ch >= 'A' && ch <= 'Z')

{

printf("'%c' is uppercase alphabet.\n", ch);

}

**else** **if**(ch >= 'a' && ch <= 'z')

{

printf("'%c' is lowercase alphabet.\n", ch);

}

**else**

{

printf("'%c' is not an alphabet.", ch);

}

}

**else**{

printf("'%c is not an alphabet\n",ch);

}

**return** 0;

}

**OUTPUT:**

Enter any character: u //User Defined

'u' is lowercase alphabet.

Program ended with exit code: 0

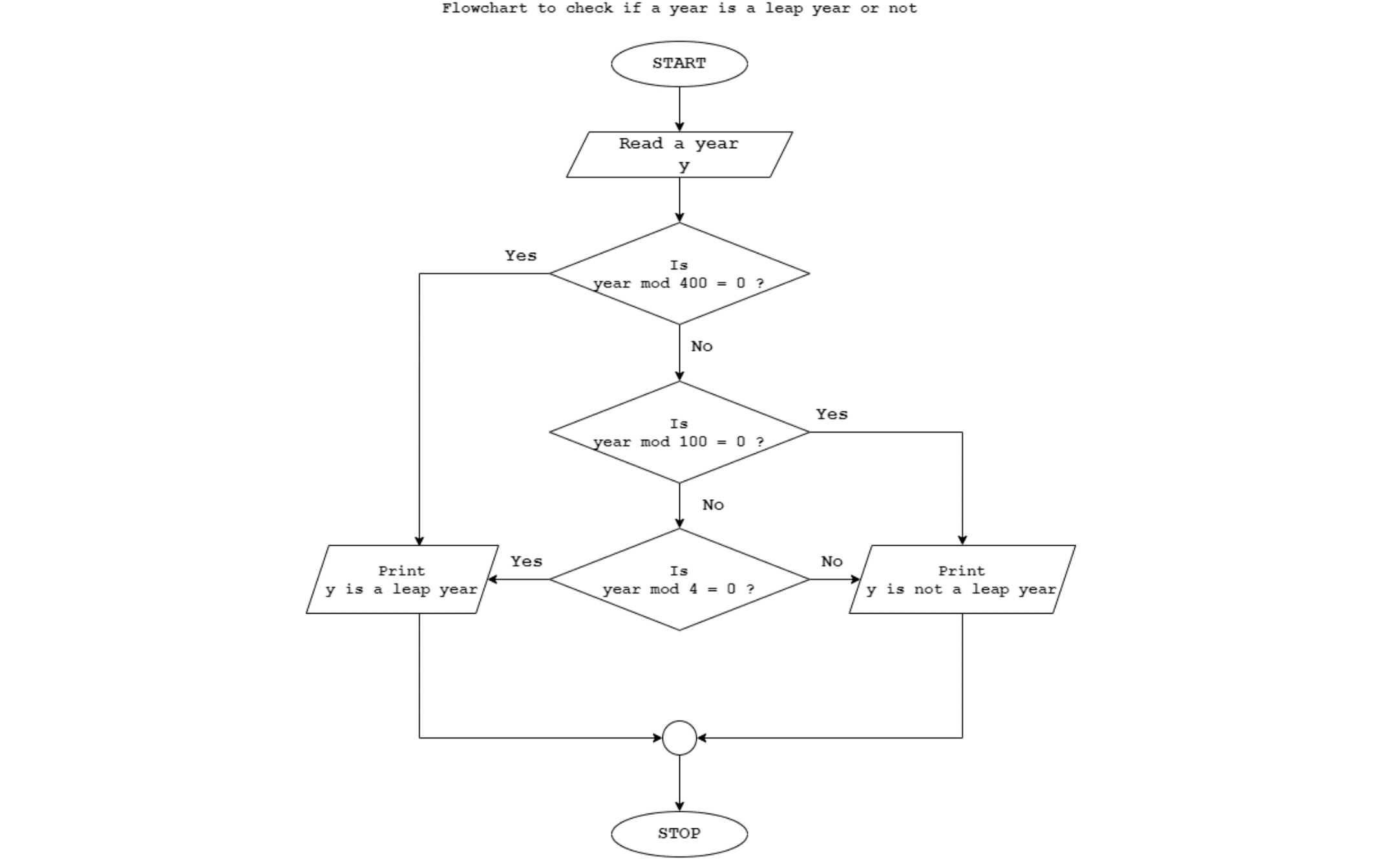
Q5.WAP to check whether the year entered by user is leap year or not.

**ALGORITHM:**

step 1: Start  
Step 2: Read year  
Step 3: If the year is divisible by 4 then go to Step 4 else go to S  
Step 4: If the year is divisible by 100 then go to Step 5 else go to

Step 5: If the year is divisible by 400 then go to Step 6 else go to  
Step 6: Print "Leap year"  
Step 7: Print "Not a leap year"  
Step 8: Stop

**FLOWCHART:**



**CODE:**

#include<stdio.h>

**int** main ()

{

**int** year;

printf("Enter the year: ");

scanf("%d",&year);

**if**(year%4==0)

{

**if**(year%100==0)

{

**if**(year%400==0)

printf("%d is a leap year\n",year);

**else**

printf("%d is not a leap year\n",year);

}

**else**{

printf("%d is a leap year\n",year);

}

}

**else**

{

printf("%d is not a leap year\n",year);

}

**return** 0;

}

**OUTPUT:**

Enter the year: 1996 //User Defined

1996 is a leap year

Process returned 0

Q6: WAP to check whether the number entered by the user is divisible by 5 and 11.

**Algorithm:**

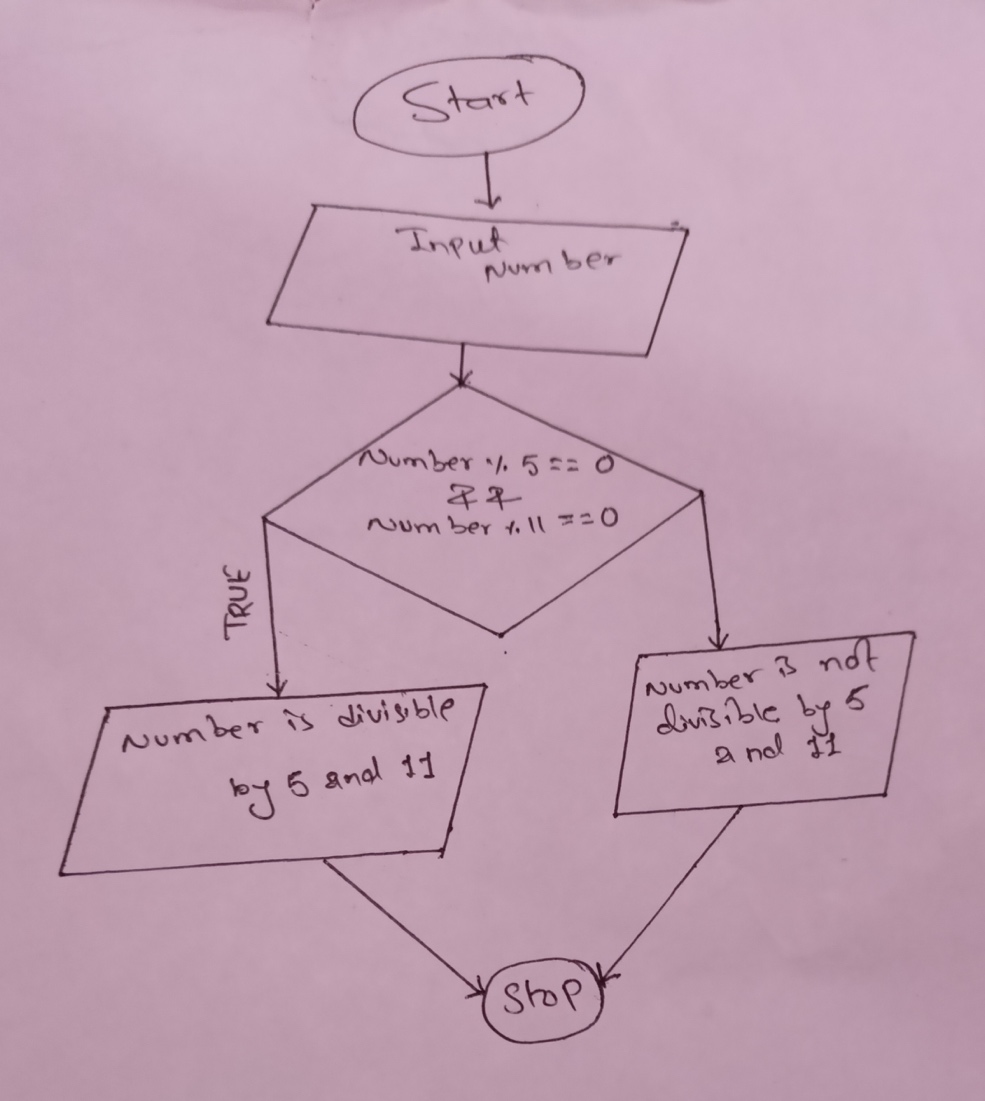
Step 1: Start.

Step 2: Read a number from the user.

Step 3: Check the number is divisible by 5 and 11 using if else or conditional operators

Step 4: Print result.

**FLOWCHART:**



**CODE:**

#include<stdio.h>

**int** main()

{

**int** number;

printf("Enter the number: ");

scanf("%d",&number);

**if**((number%5==0)&&(number%11==0))

printf("Given number %d is divisible by 5 and 11\n",number);

**else**

printf("Given number %d is not divisible by 5 and 11\n",number);

**return** 0;

}

**OUTPUT:**

Enter the number: 55 //User Defined

Given number 55 is divisible by 5 and 11

Program ended with exit code: 0

Q7. WAP to find the all the roots of a quadratic equation.

**ALGORITHM:**

Step1: Input the value of a, b, c.

Step2: Calculate k = b\*b - 4\*a\*c

Step3: If (d < 0)

Display "Roots are Imaginary, calculater1 = (-b +i ?k)/ 2a and r2 =(b + i?k)/ 2a.

else if (d = 0)

Display "Roots are Equal" and calculate r1 = r2 = (-b / 2\*a)

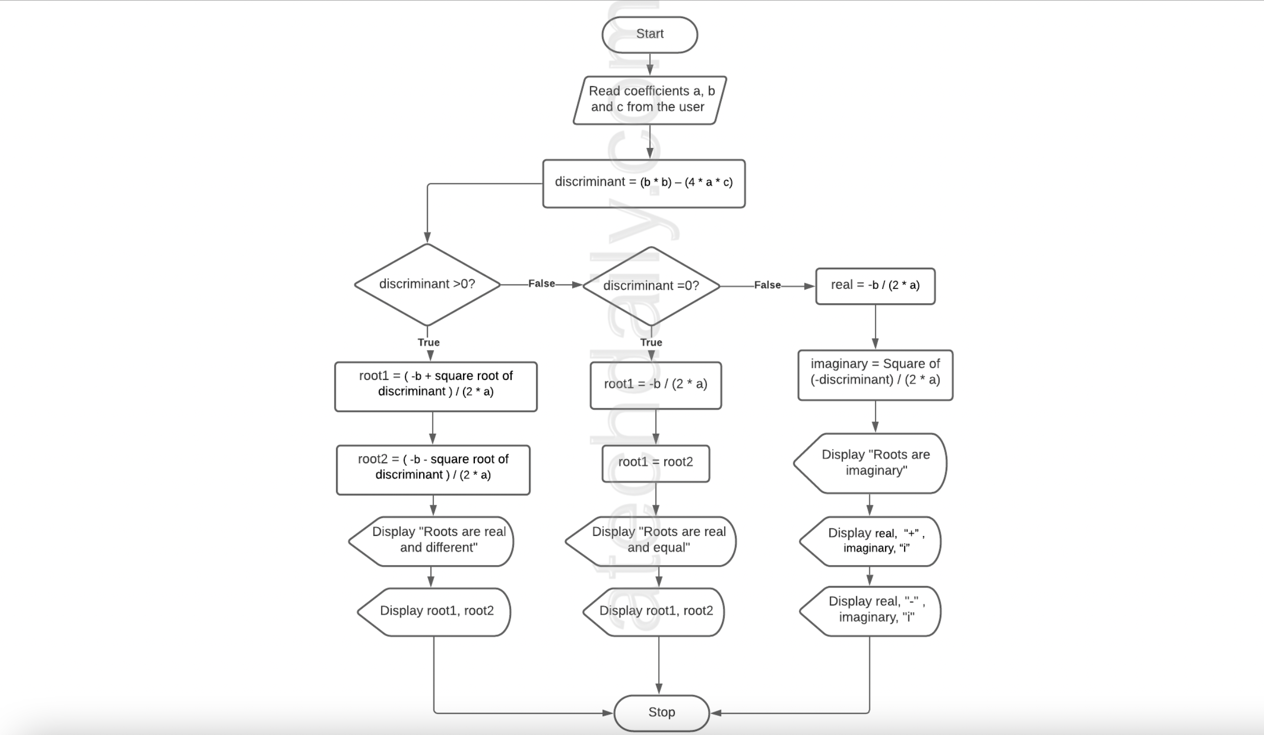
else

Display "Roots are real and calculate r1 = -b +?d / 2\*a and r2 = -b - ?d / 2\*a

Step4: Print r1 and r2.

Step5: End the algorithm

**FLOWCHART:**



**CODE:**

#include<stdio.h>

#include<math.h>

**int** main ()

{

**float** a,b,c,discriminant,root1, root2, real,img;

printf("Enter the value of a, b and c: ");

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

discriminant=b\*b-4.0\*a\*c;

**if** (discriminant==0)

{

root1=root2= (-b/(2.0\*a));

printf("Roots are real and equal \n");

printf("Root1=%lf,Root2=%lf\n",root1, root2);

}

**else** **if**(discriminant>0)

{

root1=(-b+sqrt(discriminant))/(2.0\*a);

root2=(-b-sqrt(discriminant))/(2.0\*a);

printf("Roots are real and unequal \n");

printf("Root1=%lf, Root2=%lf\n", root1, root2);

}

**else**

{

real=-b/(2.0\*a);

img=sqrt(-discriminant/(2.0\*a));

printf("Roots are imaginary \n");

printf("Root1=%lf +%lfi and Root2=%lf-%lfi\n",real,img,real,img);

}

**return** 0;

}

**OUTPUT:**

Enter the value of a, b and c: 2 4 3 //User Defined

Roots are imaginary

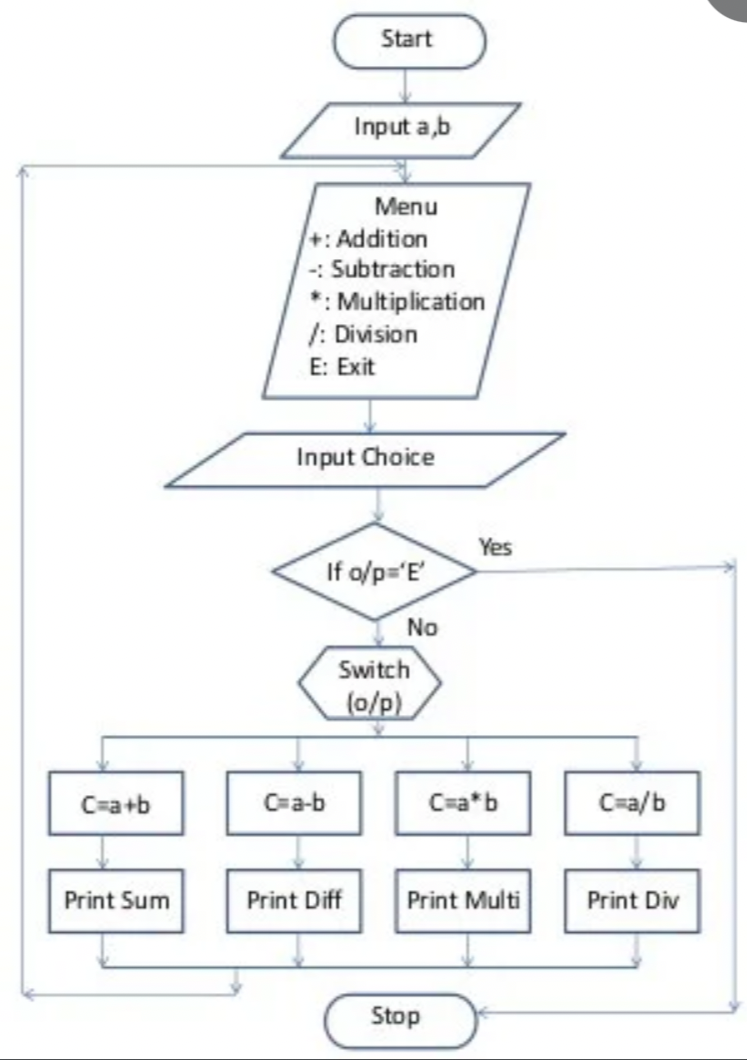
Root1=-1.000000 +1.414214i and Root2=-1.000000-1.414214i

Program ended with exit code: 0

8: WAP to input two numbers and operators among [ +, -, \*, /]. If user enters + then the program should perform the addition of the number and display the sum. I f user enters – then the program should perform the subtraction of the number and display the difference and so on for \* and /.

**ALGORITHM:**  
Step 1: Start  
Step 2: Read a and b values  
Step 3: Read option + or – or \* or /   
Step 4: If option is ‘+’ res = a + b  
Step 5: If option is ‘-’ res = a - b  
Step 6: If option is ‘\*’ res = a \* b  
Step 7: If option is ‘/’ res = a / b  
Step 8: If option does not match with + or – or \* or /   
 Print select option +, -, \*, / only  
Step 9: Print a, option, b, res values  
Step 10: Stop

**FLOWCHART:**



**CODE:**

#include <stdio.h>

**int** main()

{

**float** a, b, c;

**char** ch;

printf("Enter your operator(+, -, /, \*, %): ");

scanf("%c", &ch);

printf("Enter the values of a and b: ");

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

**switch**(ch)

{

**case** '+': c = a + b;

printf("addition of two numbers is %.2f\n", c);

**break**;

**case** '-': c = a - b;

printf("substraction of two numbers is %.2f\n", c);

**break**;

**case** '\*': c = a \* b;

printf("multiplication of two numbers is %.2f\n", c);

**break**;

**case** '/': c = (**float**)a / b;

printf("remainder of two numbers is %.2f\n", c);

**break**;

**default**: printf("Invalid operator\n");

**break**;

}

**return** 0;

}

**OUTPUT:**

Enter your operator(+, -, /, \*, ): +

Enter the values of a and b: 10 30

addition of two numbers is 40

Program ended with exit code: 0

Q9: WAP in C to input marks of 5 subjects C-programming, Physics, Math's, Applied Mechanics and Basic electrical. Display whether the student passed or failed. Take F.M=100 and P.M=40. For passed students calculate percentage and grade according to following:

Percentage>=90%: A

Percentage>=80%: B

Percentage>=70%: C

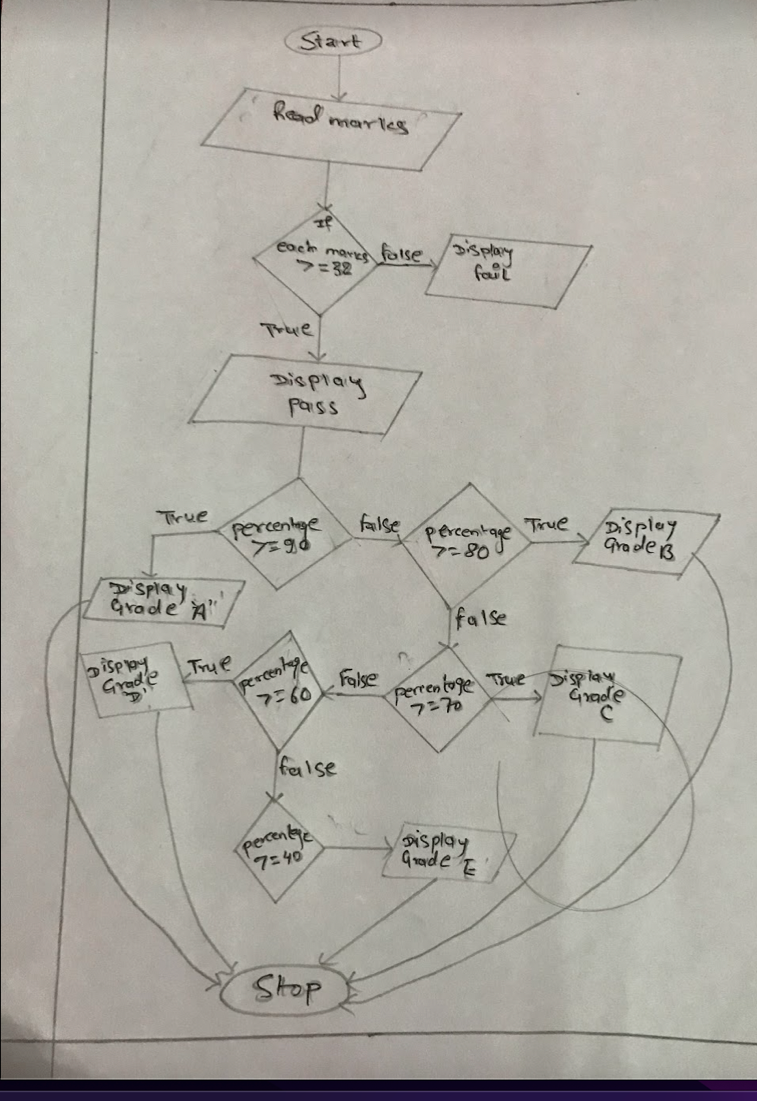
Percentage>=60%: D

Percentage>=40%: E

**ALGORITHM:**

Step1:

**FLOWCHART:**



**CODE:**

#include <stdio.h>

**int** main()

{

**int** C, phy, applied, math, electrical;

**float** per;

/\* Input marks of five subjects from user \*/

printf("Enter five subjects marks: ");

scanf("%d%d%d%d%d", &C, &phy, &applied, &math, &electrical);

**if**(C>=32 && phy>=32 && applied>=32 && math>=32 && electrical>=32)

{

printf("You have passed the Exams.\n");

/\* Calculate percentage \*/

per = (C + phy + applied + math + electrical) / 5.0;

printf("Percentage = %.2f\n", per);

/\* Find grade according to the percentage \*/

**if**(per >= 90)

{

printf("Grade A\n");

}

**else** **if**(per >= 80)

{

printf("Grade B\n");

}

**else** **if**(per >= 70)

{

printf("Grade C\n");

}

**else** **if**(per >= 60)

{

printf("Grade D\n");

}

**else** **if**(per >= 40)

{

printf("Grade E\n");

}

}

**else**{

printf("you have failed the Exams.\n");

}

**return** 0;

}

**OUTPUT:**

Enter five subjects marks: 40 50 65 72 80 //User Defined

You have passed the Exams.

Percentage = 61.40

Grade D

Program ended with exit code: 0

