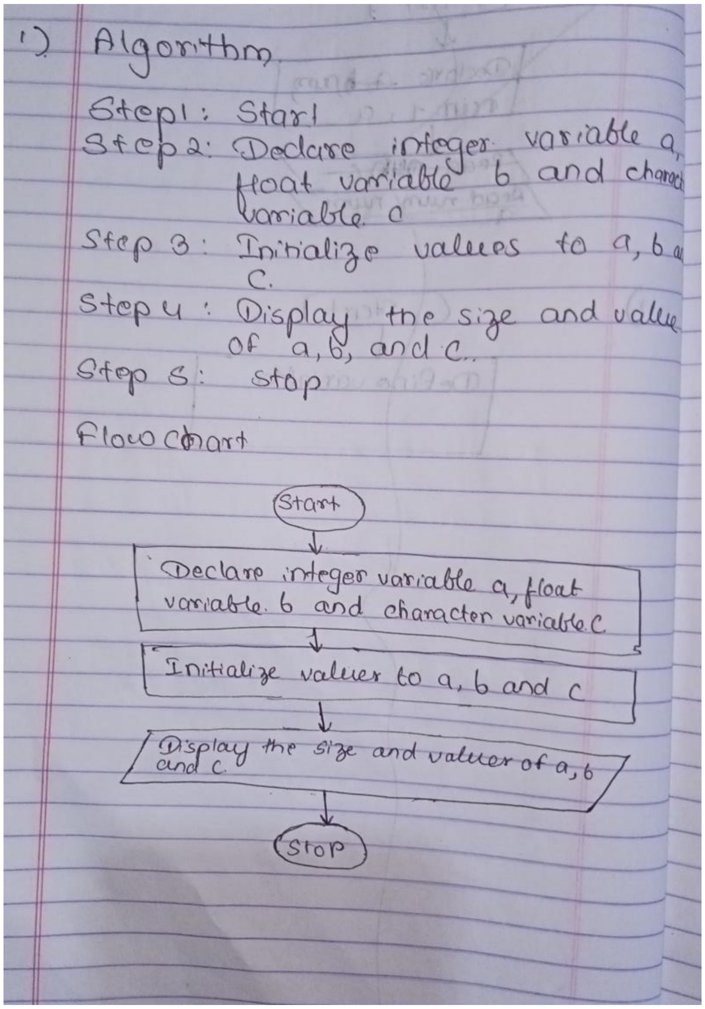
Lab-2

1. WAP to declare integer, float and character variable. Initialize them with certain value and print those values. Also display the size of variables.

   Algorithm and Flowchart:



1) Algorithm.

Step1: Start
Step2: Declare integer variable a, float variable b and charact
variable c
Step 3: Initialize values to a, b a
c.
Step 4: Display the size and value of a, b, and c.
Step 5: stop

Flowchart

Start

Declare integer variable a, float variable b and character variable c

Initialize values to a, b and c

Display the size and values of a, b and c.

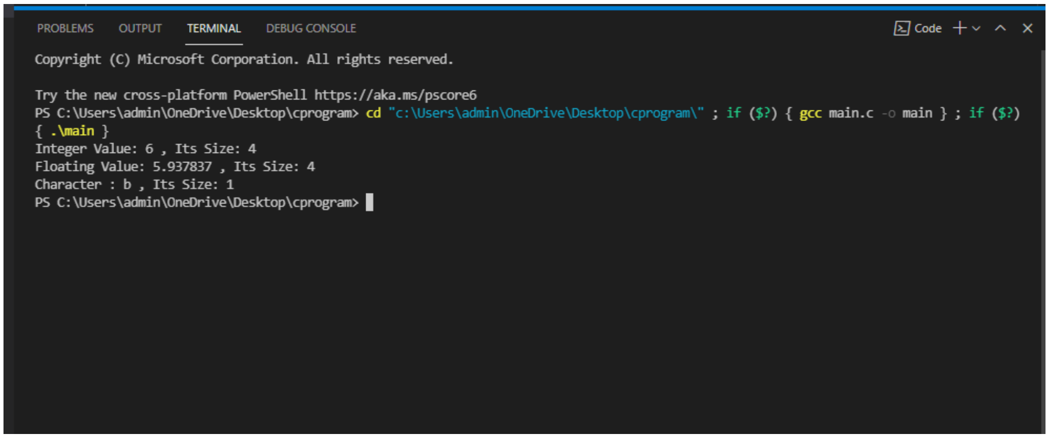
Stop

   Program:

```
#include<stdio.h>
int main(){
   int a= 6;
   float b = 5.937837;
   char c = 'b';
   printf("Integer Value: %d , Its Size: %d\n", num, sizeof(num));
   printf("Floating Value: %f , Its Size: %d\n", num1, sizeof(num1));
   printf("Character : %c , Its Size: %d\n", a, sizeof(a));
```

```
    return 0;
}
```

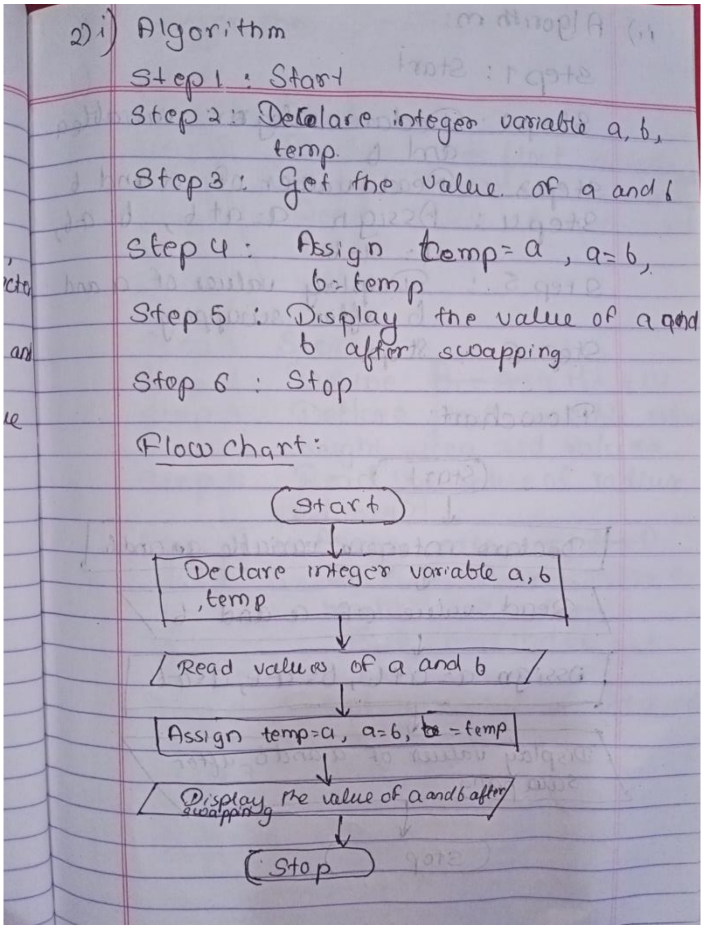
Output:



```
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o main } ; if ($?)
{ .\main }
Integer Value: 6 , Its Size: 4
Floating Value: 5.937837 , Its Size: 4
Character : b , Its Size: 1
PS C:\Users\admin\OneDrive\Desktop\cprogram>
```

2. WAP to swap the values of the variable with and without using third variable.

   a) With third variable
      Algorithm And Flowchart:



2) i) Algorithm

Step 1 : Start

Step 2 : Declare integer variable a, b, temp.

Step 3 : Get the value of a and b

Step 4 : Assign temp = a , a = b, b = temp

Step 5 : Display the value of a and b after swapping

Step 6 : Stop

Flow chart:

Start

Declare integer variable a, b ,temp

Read values of a and b

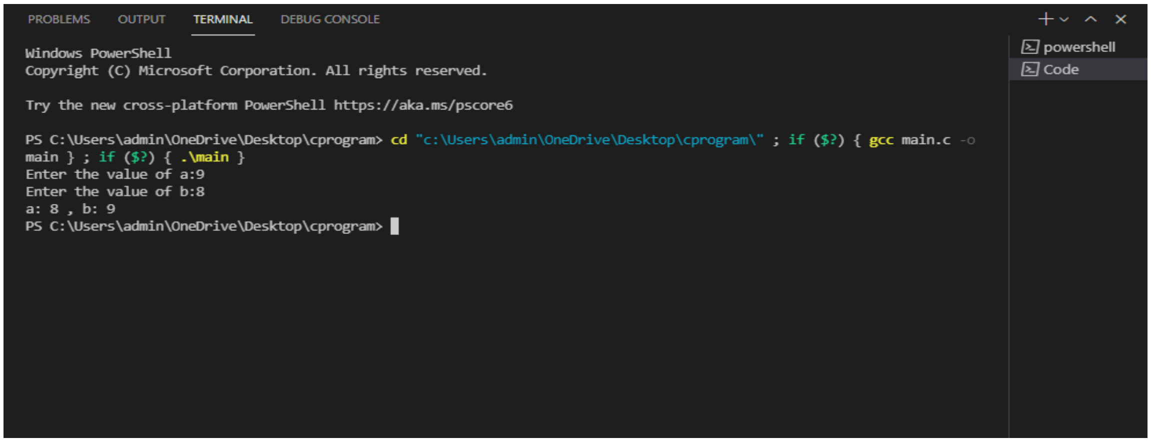
Assign temp=a, a=b, b=temp

Display the value of a and b after swapping

Stop

Code:

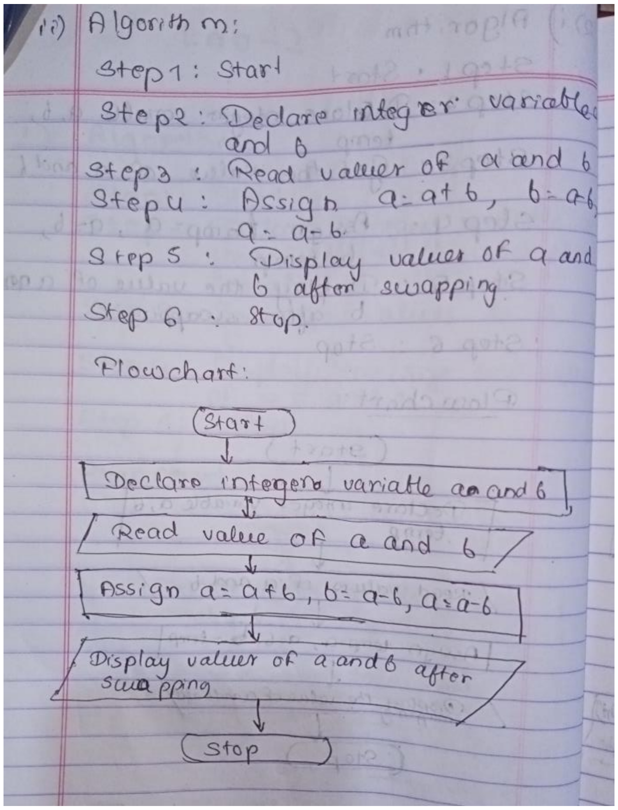
```
#include<stdio.h>
int main(){
   int a, b, temp;
   printf("Enter the value of a:");
   scanf("%d", &a);
   printf("Enter the value of b:");
   scanf("%d", &b);
   temp = a;
   a = b;
   b= temp;
   printf("a: %d , b: %d", a, b);
   return 0;
}
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter the value of a:9
Enter the value of b:8
a: 8 , b: 9
PS C:\Users\admin\OneDrive\Desktop\cprogram>
```

b) Without third variable

Algorithm and flowchart:



```
ii) Algorithm:
    Step1 : Start
    Step2 : Declare integer variables
            and b
    Step3 : Read values of a and b
    Step4 : Assign a = a+b, b = a-b,
            a = a-b
    Step 5 : Display values of a and
             b after swapping
    Step 6 : Stop.
```

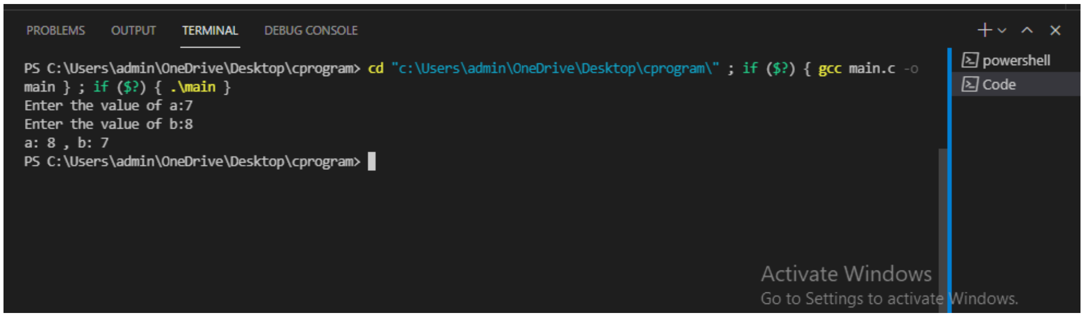
Flowchart:

Start
Declare integers variable a and b
Read value of a and b
Assign a = a+b, b = a-b, a = a-b
Display values of a and b after swapping
Stop

Code:

```
#include<stdio.h>
int main(){
   int a, b;
   printf("Enter the value of a:");
   scanf("%d", &a);
```

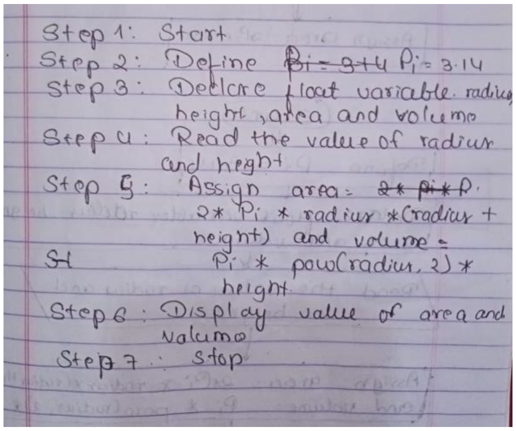
```
    printf("Enter the value of b:");
    scanf("%d", &b);
    a = a+b;
    b = a-b;
    a = a-b;
    printf("a: %d , b: %d", a, b);
    return 0;
}
```



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter the value of a:7
Enter the value of b:8
a: 8 , b: 7
PS C:\Users\admin\OneDrive\Desktop\cprogram>
```

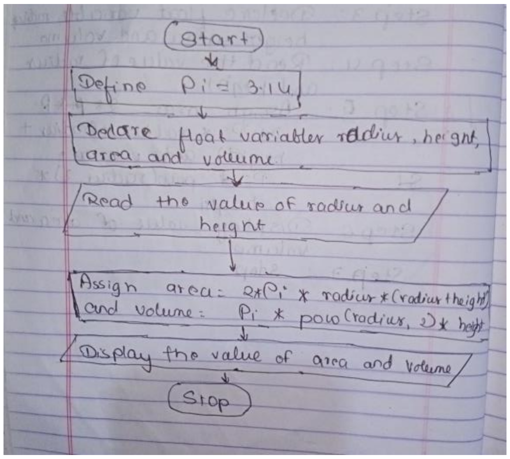
3. WAP to calculate the area and volume of a cylinder using pre-processor directive for value of PI.

Algorithm:



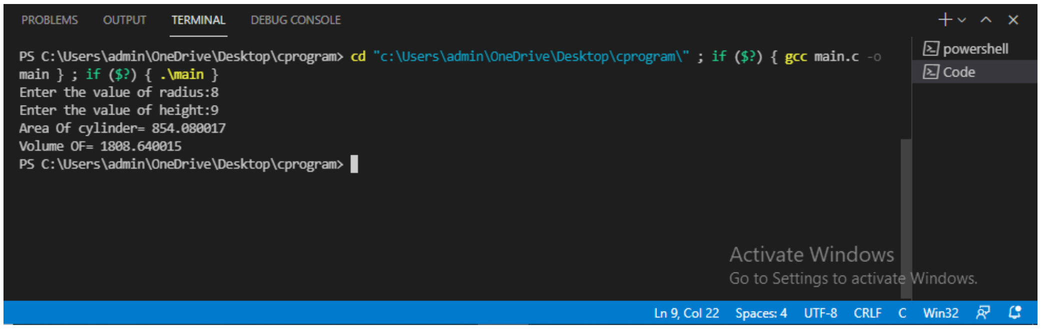
Step 1: Start
Step 2: Define ~~Pi = 3.14~~ Pi = 3.14
Step 3: Declare float variable radius, height, area and volume
Step 4: Read the value of radius and height
Step 5: Assign area = ~~2 * pi * R~~ 2 * Pi * radius *(radius + height) and volume =
~~St~~ Pi * pow(radius, 2) * height
Step 6: Display value of area and volume
Step 7: stop

Flowchart:


Start
Define Pi = 3.14
Declare float variabler radius, height, area and volume
Read the value of radius and height
Assign area = 2*Pi * radius *(radius+height) and volume = Pi * pow(radius, 2)* height
Display the value of area and volume
Stop

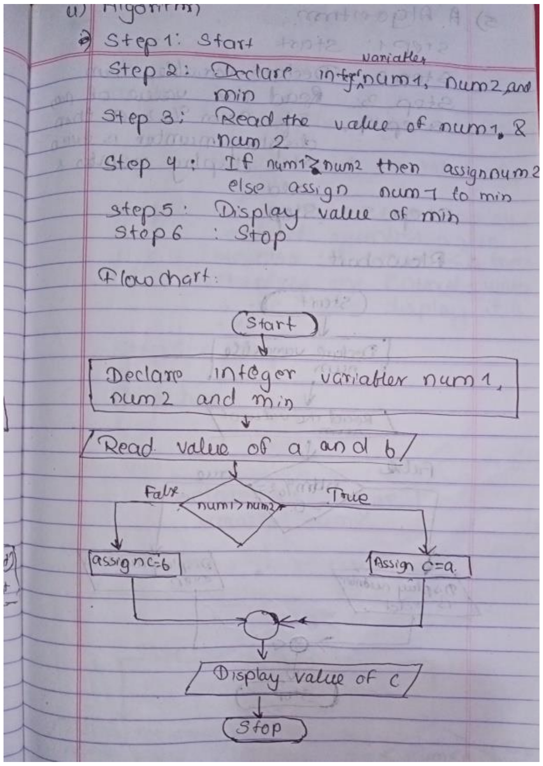
Code:

```
#include<stdio.h>
#include<math.h>
#define Pi 3.14
int main(){
   float radius, height, area, volume;
   printf("Enter the value of radius:");
   scanf("%f", &radius);
   printf("Enter the value of height:");
   scanf("%f", &height);
   area = 2*Pi*radius*(radius+height);
   volume= Pi * pow(radius,2) * height;
   printf("Area Of cylinder= %f", area);
   printf("Volume OF= %f", volume);

   return 0;
}
```



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter the value of radius:8
Enter the value of height:9
Area Of cylinder= 854.080017
Volume OF= 1808.640015
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

4. WAP to input two numbers from user and display the minimum using conditional operator.
Algorithm and flowchart:



u) Algorithm

Step 1: Start

Step 2: Declare integer variables num1, num2 and min

Step 3: Read the value of num1 & num 2

Step 4: If num1>num2 then assign num2 else assign num1 to min

step 5: Display value of min

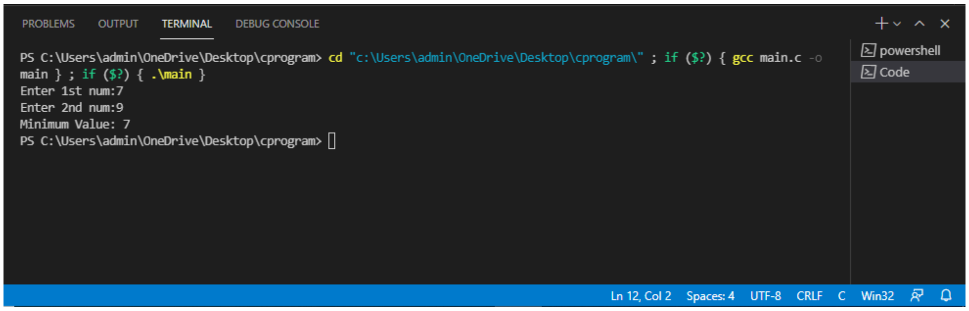
step 6 : Stop

Flowchart:

Start
Declare integer variables num1, num2 and min
Read value of a and b
False
num1>num2
True
assign c=b
Assign c=a
Display value of c
Stop

Code:

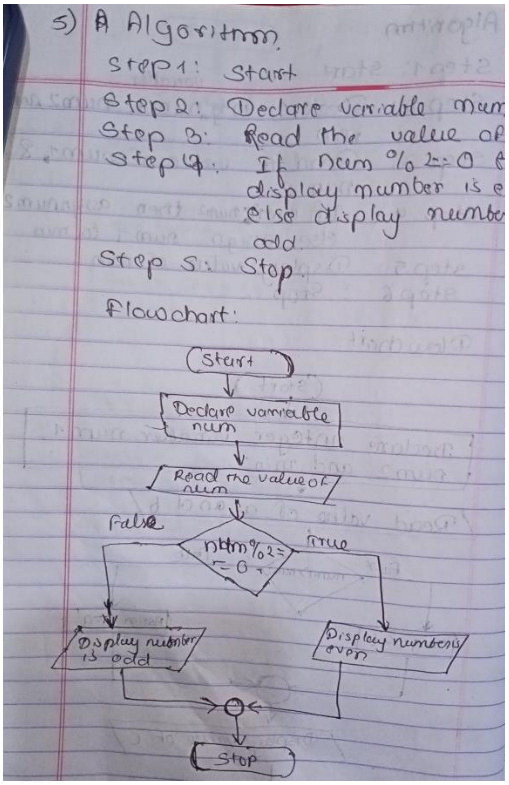
```
#include<stdio.h>
int main(){
    int num1,num2, min;
    printf("Enter 1st num:");
    scanf("%d", &num1);
    printf("Enter 2nd num:");
    scanf("%d", &num2);
    min = (num1>num2)?num2:num1;
    printf("Minimum Value: %d", min);

    return 0;
}
```



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter 1st num:7
Enter 2nd num:9
Minimum Value: 7
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

5) WAP to display whether a number is even or odd using conditional operator

Algorithm and Flowchart:



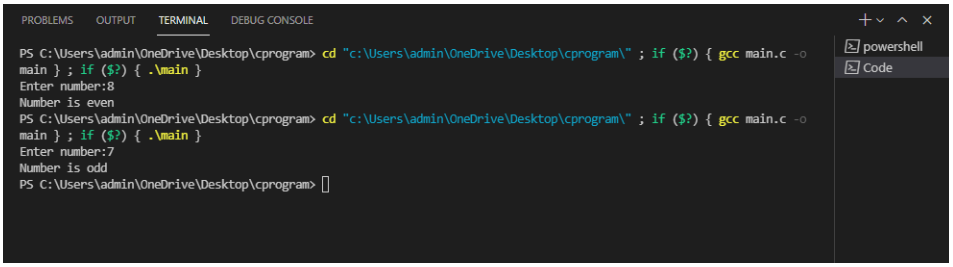
5) Algorithm.

```
Step1: Start
Step 2: Declare variable num
Step 3: Read the value of
Step 4: If num % 2==0 t
        display number is e
        else display numbe
        odd.
Step 5: Stop.
```

Flowchart:

Code:

```
#include<stdio.h>
int main(){
    int num;
    printf("Enter number:");
    scanf("%d", &num);
    (num%2==0)?printf("Number is even"):printf("Number is odd");

    return 0;
}
```



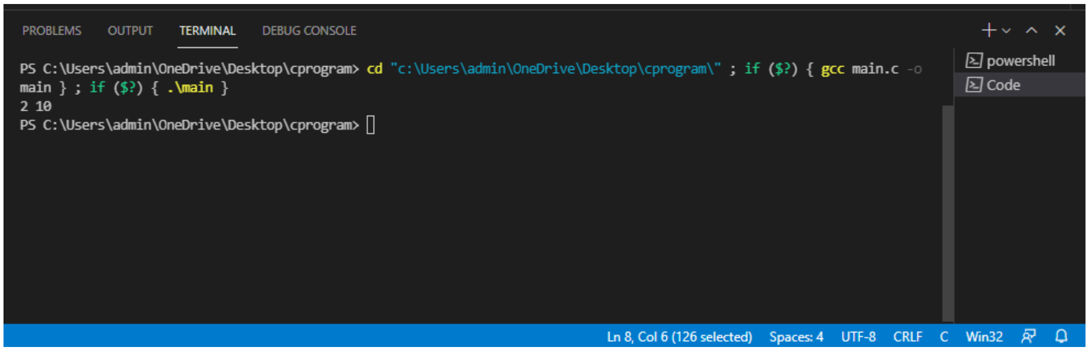
```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter number:8
Number is even
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter number:7
Number is odd
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

6.What are the output of the following programs.

a) Code:

```
#include<stdio.h>

int main()

{

   int a=2,b;

   b=a++ + a--+ ++a + --a;

   printf("%d %d",a,b);

   return 0;

   }
```

Output

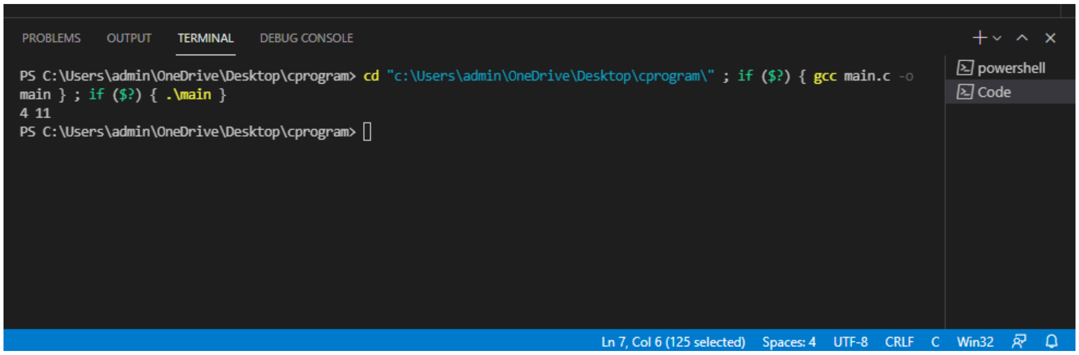


```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
2 10
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

.b) Code:

```
 #include<stdio.h>
 int main(){
   int a=2,b;
   b=++a + a--+ ++a + a++;
   printf("%d %d",a,b);
   return 0;
   }
```

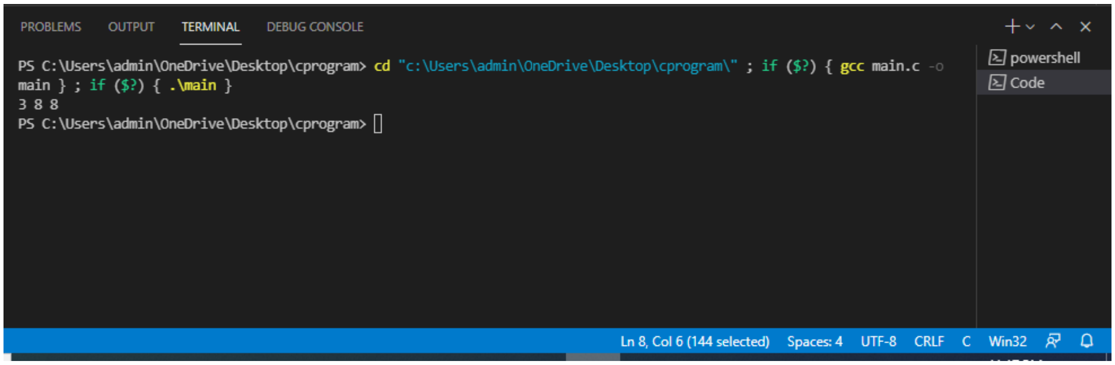
Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
4 11
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

c) Code:

```
 #include<stdio.h>
int main(){
   int a=2,b,c;
   b=++a + a--+ ++a;
   c=(a>b)?a:b;
   printf("%d %d %d",a,b,c);
   return 0;
   }
```

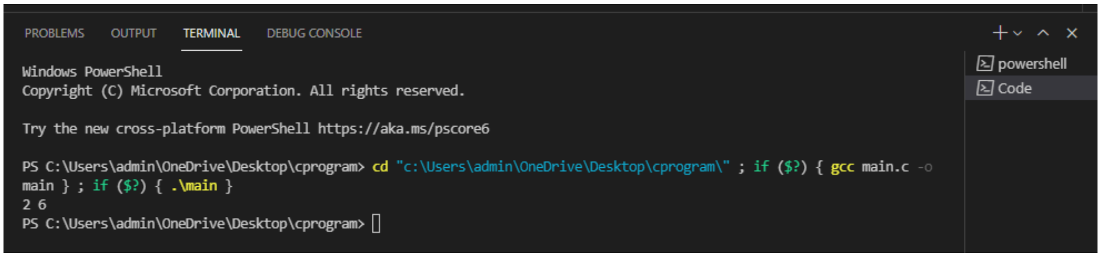


```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
3 8 8
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

d)Code:

```
   #include<stdio.h>

     int main(){

        int a=2,b;

        b=++a *a--;

        printf("%d %d",a,b);

        return 0;

        }
```

Output:

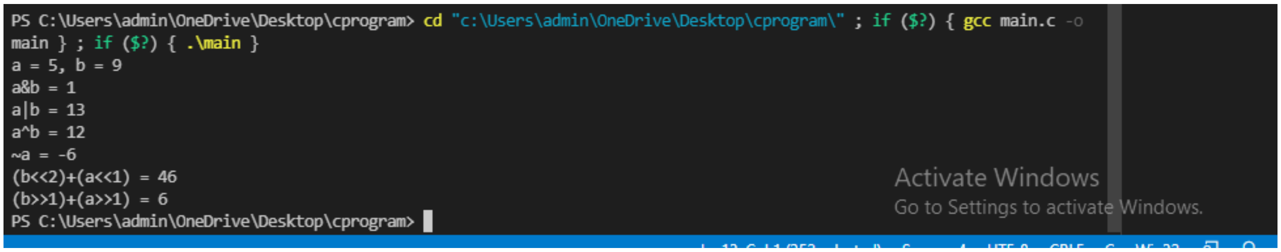


```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
2 6
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

e) Code:

```
#include <stdio.h>
int main(){
    int a = 5, b = 9;
    printf("a = %d, b = %d\n", a, b);
    printf("a&b = %d\n", a & b);
    printf("a|b = %d\n", a | b);
    printf("a^b = %d\n", a ^ b);
    printf("~a = %d\n", ~a);
    printf("(b<<2)+(a<<1) = %d\n",(b<<2)+(a<<1));
    printf("(b>>1)+(a>>1) = %d\n",(b>>1)+(a>>1));
    return 0;
}
```

Output:

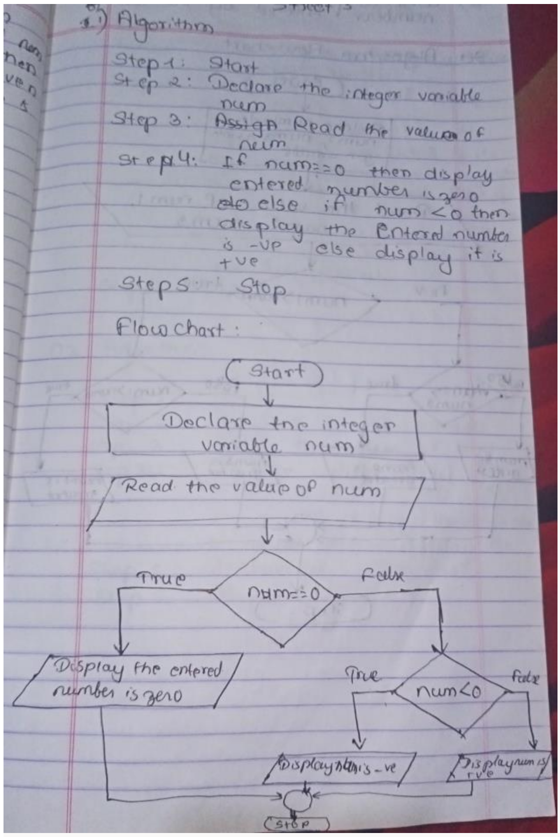


```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
a = 5, b = 9
a&b = 1
a|b = 13
a^b = 12
~a = -6
(b<<2)+(a<<1) = 46
(b>>1)+(a>>1) = 6
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

LAB-3

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

Algorithm and Flowchart:



1) Algorithm

Step 1: Start

Step 2: Declare the integer variable num

Step 3: ~~Assign~~ Read the value of num

Step 4: If num==0 then display entered number is zero ~~elo~~ else if num<0 then display the Enterd number is -ve else display it is +ve

Step 5: Stop

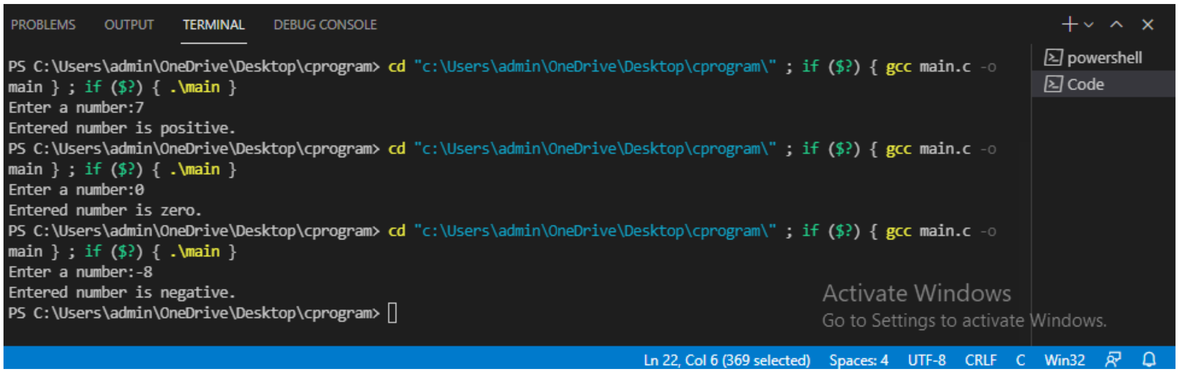
Flow chart:

Start
Declare the integer variable num
Read the value of num
num==0
True
False
Display the entered number is zero
num<0
True
False
Display num is -ve
Display num is +ve
Stop

Code:

```
#include<stdio.h>
int main()
{
   int num;
   printf("Enter a number:");
   scanf("%d", &num);
   if(num == 0)
   {
       printf("Entered number is zero.");
   }
   else
   {
       if(num<0)
       {
         printf("Entered number is negative.");
       }
       else
       {
         printf("Entered number is positive.");
       }
```

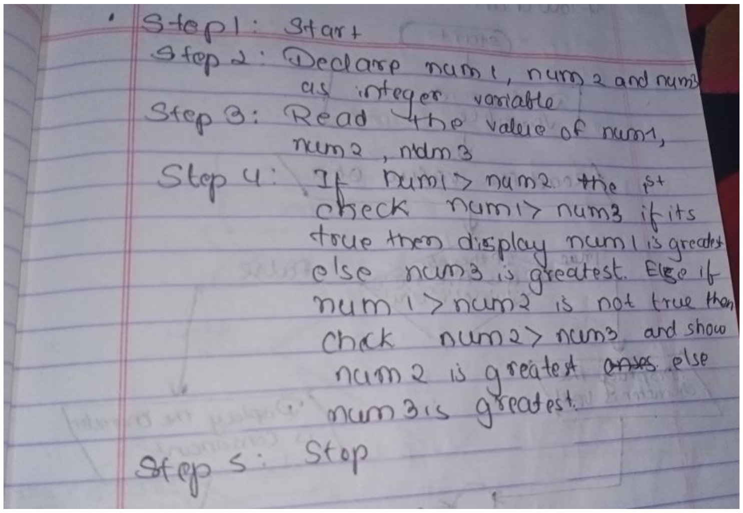
```
  }
 }
```



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:7
Entered number is positive.
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:0
Entered number is zero.
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:-8
Entered number is negative.
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

2. WAP to find maximum between three numbers entered by the user

Algorithm:



Step1: Start

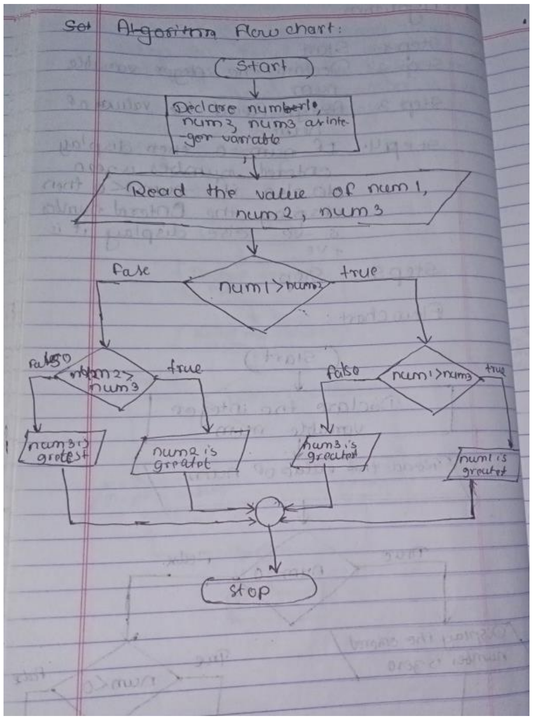
Step 2: Declare num1, num2 and num3 as integer variable

Step 3: Read the value of num1, num2, num3

Step 4: If num1 > num2 the 1st check num1 > num3 if its true then display num1 is greatest else num3 is greatest. Else if num1 > num2 is not true then check num2 > num3 and show num2 is greatest ~~ans~~. else num3 is greatest.

Step 5: Stop

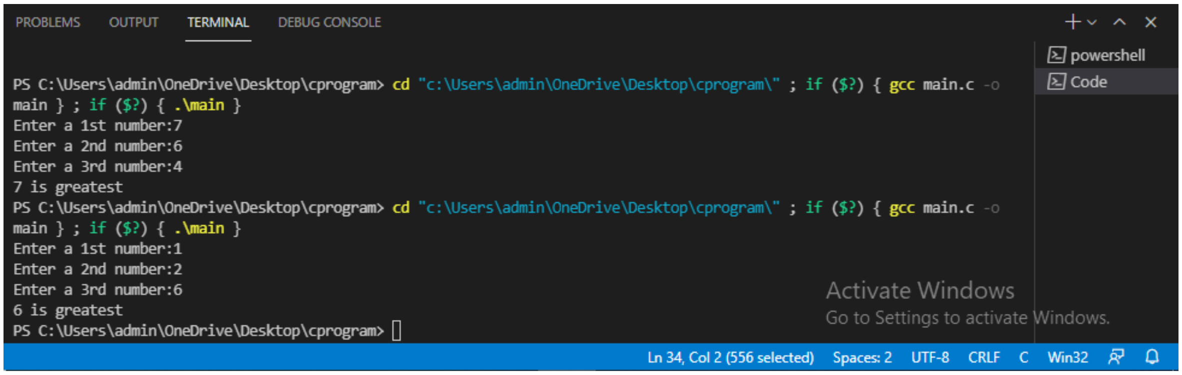
Flow chart:



Code:

```
#include<stdio.h>
int main()
{
 int num1, num2, num3;
 printf("Enter a 1st number:");
 scanf("%d",&num1);
 printf("Enter a 2nd number:");
 scanf("%d",&num2);
 printf("Enter a 3rd number:");
 scanf("%d",&num3);
 if(num1>num2)
 {
  if(num1>num3)
  {
   printf("%d is greatest", num1);
  }
  else
  {
   printf("%d is greatest", num3);
  }
 }
 else
 {
  if(num2>num3)
  {
   printf("%d is greatest", num2);
  }
  else
  {
   printf("%d is greatest", num3);
  }
 }
 return 0;
}
```

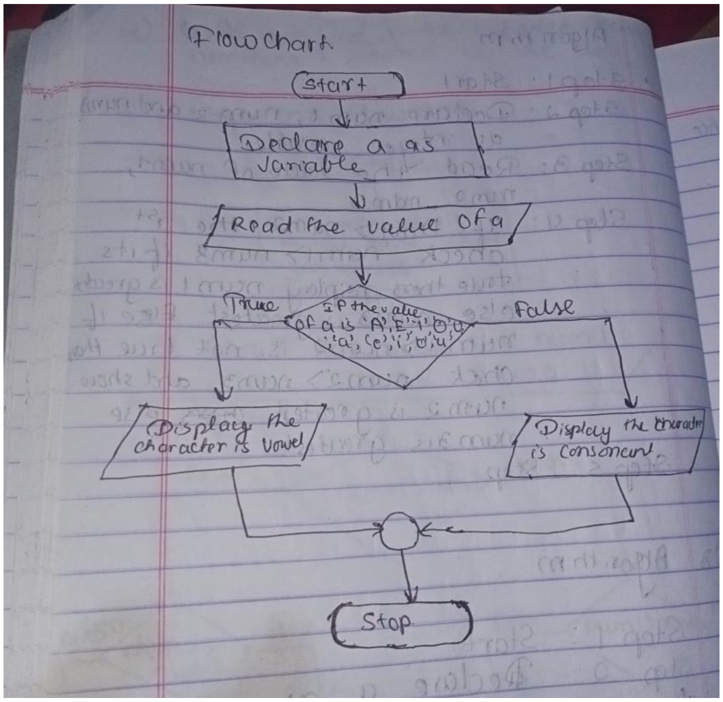
.Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a 1st number:7
Enter a 2nd number:6
Enter a 3rd number:4
7 is greatest
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a 1st number:1
Enter a 2nd number:2
Enter a 3rd number:6
6 is greatest
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

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

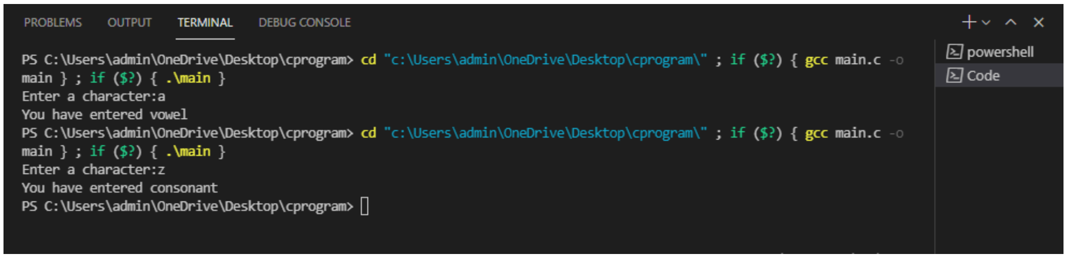
Flowchart:


Flow chart.
Start
Declare a as Variable
Read the value of a
If the value of a is 'A','E','I','O','U' ,'a','e','i','o','u'
True
False
Display the character is vowel
Display the character is Consonant.
Stop.

Code:

```
#include<stdio.h>

int main()

{

 char a;

 printf("Enter a character:");

 scanf("%c",&a);

 if (a=='a'||a=='e'||a=='i'||a=='o'||a=='u'||a=='A'||a=='E'||a=='I'||a=='O'||a=='U')

 {

  printf("You have entered vowel");

 }

 else

 {
   printf("You have entered consonant");

 }

 return 0;

}
```

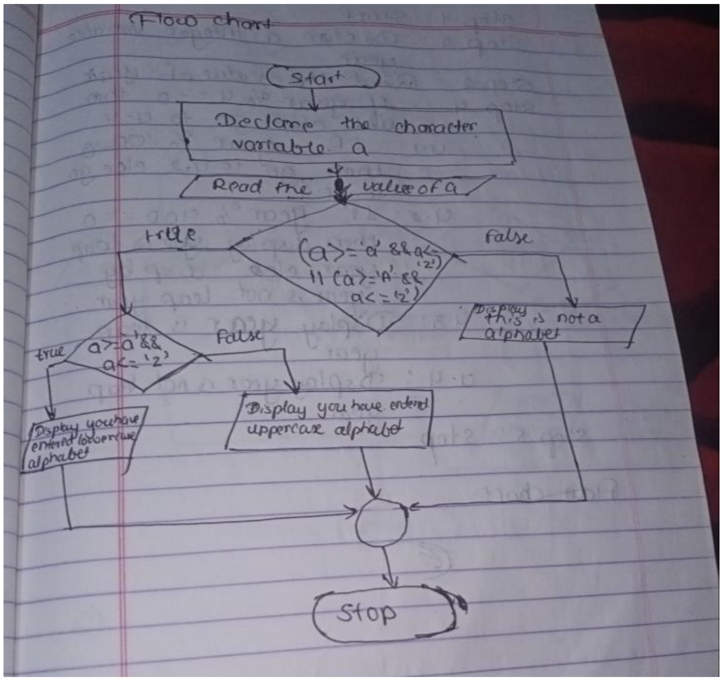
Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a character:a
You have entered vowel
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a character:z
You have entered consonant
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

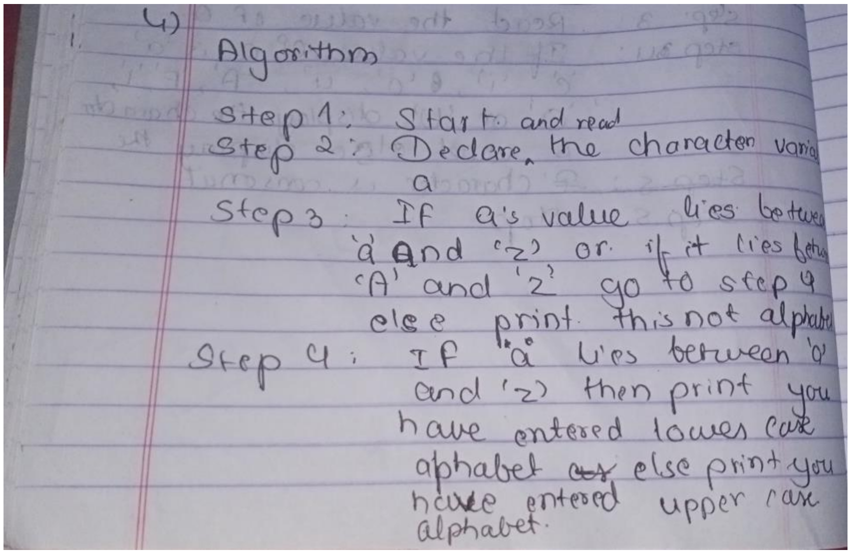
4.WAP to input a character from the user and check whether the character is Alphabet or not. If the character is Alphabet then show whether it is uppercase or lowercase.

Flowchart:



Flow chart
Start
Declare the character variable a
Read the value of a
(a>='a' && a<='z') || (a>='A' && a<='z')
true
False
Display this is not a alphabet
a>='a' && a<='z'
true
False
Display you have entered lowercase alphabet
Display you have enterd uppercase alphabet
Stop

Algorithm:,



4)

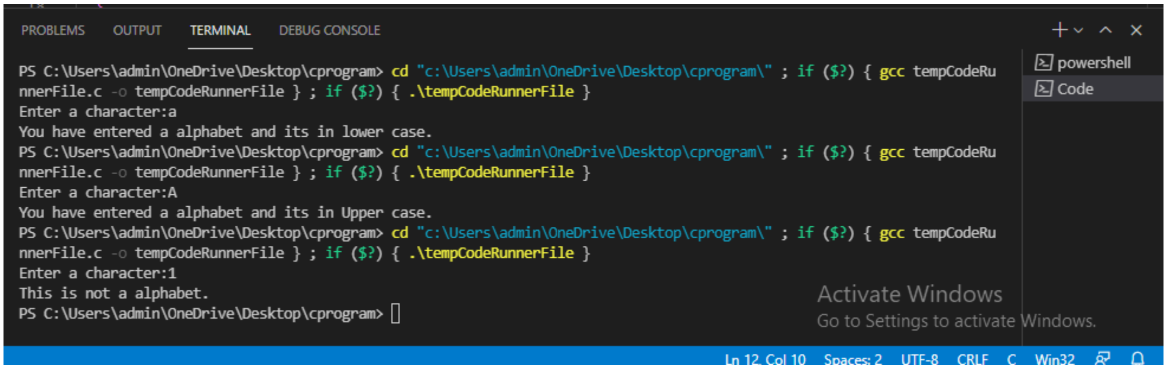
Algorithm

Step 1: Start and read
Step 2: Declare the character varia
a
Step 3: If a's value lies betwee
'a' and 'z' or if it lies betw
'A' and 'Z' go to step 4
else print this not alphabe
Step 4: If 'a' lies between 'a'
and 'z' then print you
have entered lower case
aphabet ~~or~~ else print you
have entered upper case
alphabet.

Code:

```
#include<stdio.h>

int main()

{

  char a;

  printf("Enter a character:");

  scanf(" %c", &a);

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

    if(a>='a' && a<='z')

    {

      printf("You have entered a alphabet and its in lower case.");

    }

    else{

       printf("You have entered a alphabet and its in Upper case.");

    }

  }

  else{
        printf("This is not a alphabet.");
    }
  }
```

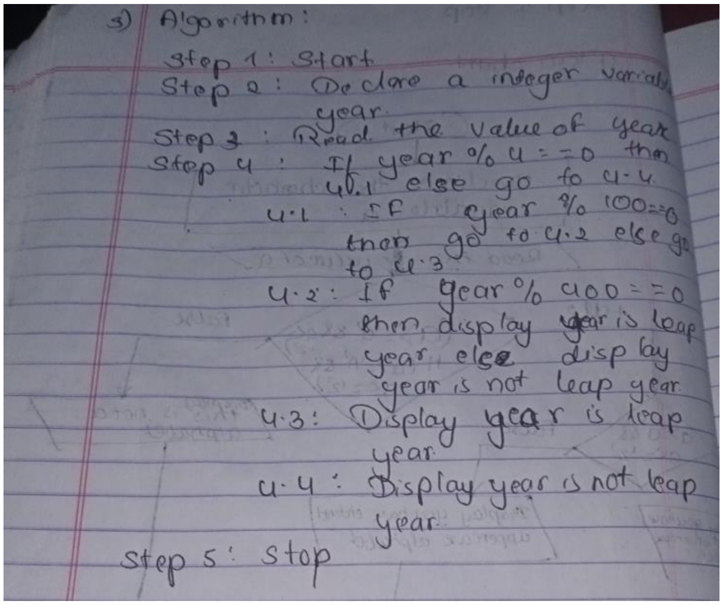
Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc tempCodeRu
nnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter a character:a
You have entered a alphabet and its in lower case.
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc tempCodeRu
nnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter a character:A
You have entered a alphabet and its in Upper case.
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc tempCodeRu
nnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter a character:1
This is not a alphabet.
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

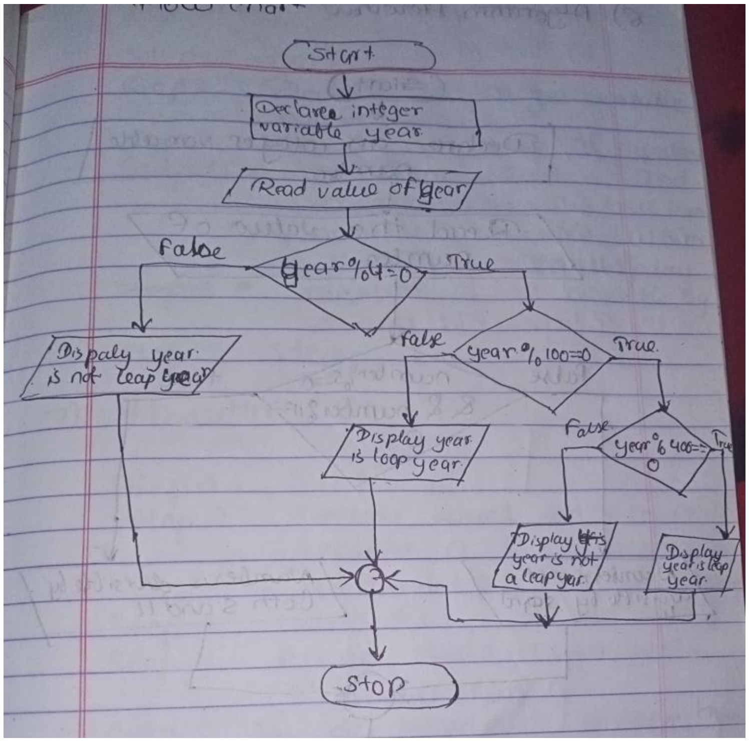
Algorithm:

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



```
5) Algorithm:
    Step 1: Start
    Step 2: Declare a integer variab
            year.
    Step 3: Read the value of year
    Step 4: If year % 4 == 0 then
            4.1 else go to 4.4
        4.1: If year % 100 == 0
             then go to 4.2 else go
             to 4.3
        4.2: If year % 400 == 0
             then, display year is leap
             year else display
             year is not leap year.
        4.3: Display year is leap
             year.
        4.4: Display year is not leap
             year.
    Step 5: Stop
```

Flowchart:

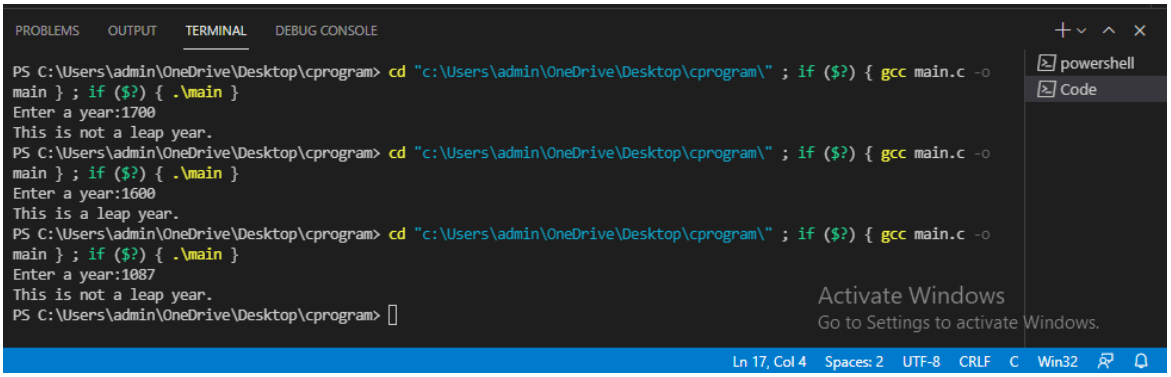

Start
Declare integer variable year
Read value of year
False
year%4=0
True
Dispaly year is not leap year
false
year%100==0
True
Display year is leap year.
False
year%400==0
True
Display this year is not a leap year.
Display year is leap year.
Stop

Code:

```
#include <stdio.h>
int main()
{
  int year;
  printf("Enter a year: ");
  scanf("%d", &year);
  if (year % 4 == 0)
  {
    if (year % 100 == 0)
    {
      if (year % 400 == 0)
      {
```

```
    printf("%d is leap year.", year);
   }
   else
   {
    printf("%d is not leap year.", year);
   }
  }
  else
  {
   printf("%d is leap year.", year);
  }
 }
 else
  printf("%d is not leap year.", year);
 return 0;
}
```

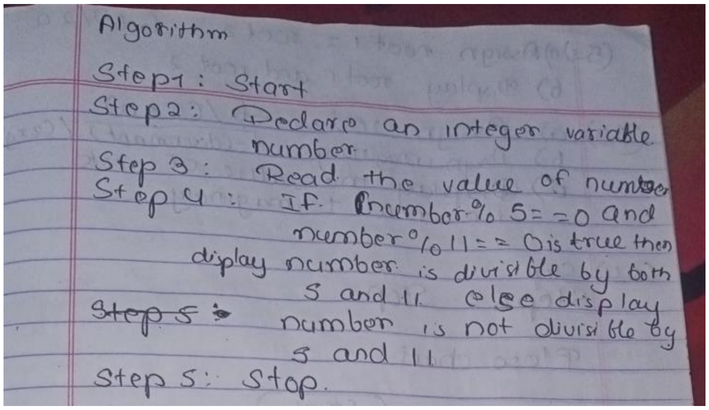
Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a year:1700
This is not a leap year.
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a year:1600
This is a leap year.
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a year:1087
This is not a leap year.
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

6) WAP to check whether the number entered by the user is divisible by 5 and 11 or not.

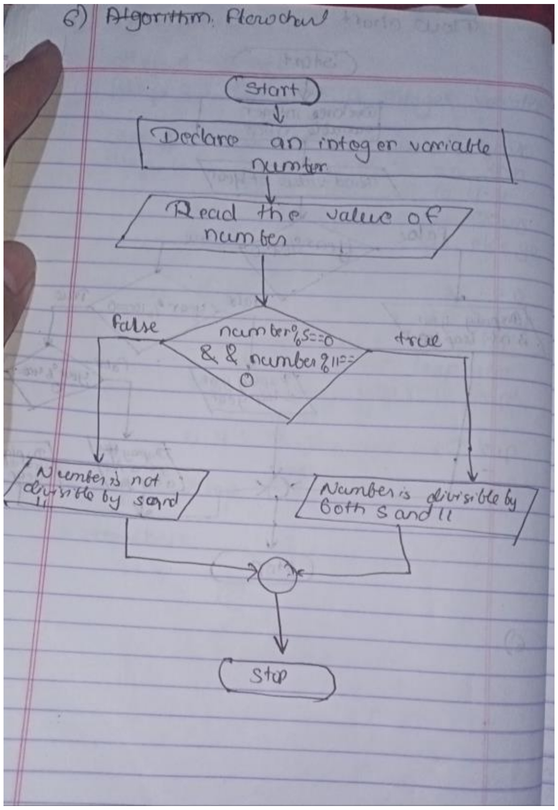
Algorithm:



Algorithm

```
Step1: Start
Step2: Declare an integer variable
       number
Step 3: Read the value of number
Step 4: If number % 5 = = 0 and
           number % 11 = = 0 is true then
        display number is divisible by both
           5 and 11 else display
~~Step 5~~: number is not divisible by
           5 and 11
Step 5: Stop.
```

Flowchart:

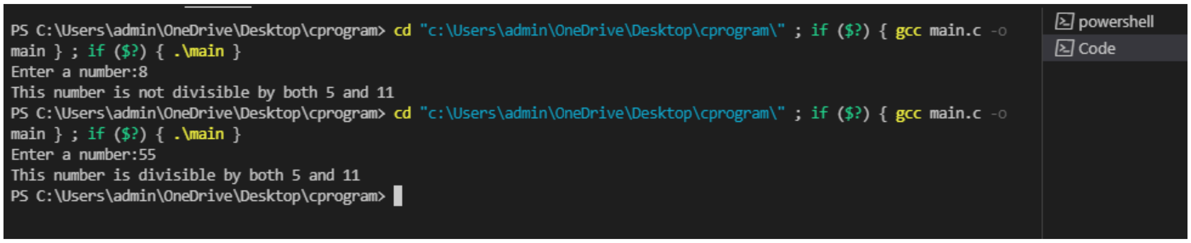

6) Algorithm Flowchart
Start
Declare an integer variable number
Read the value of number
number%5==0 && number%11==0
False
true
Number is not divisible by 5 and 11
Number is divisible by both 5 and 11
Stop

Code:

```
#include<stdio.h>

int main()

{

  int number;
```

```
 printf("Enter a number:");

 scanf("%d", &number);

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

 {

   printf("This number is divisible by both 5 and 11");

 }

 else

 {

   printf("This number is not divisible by both 5 and 11");

 }



 return 0;

}.
```

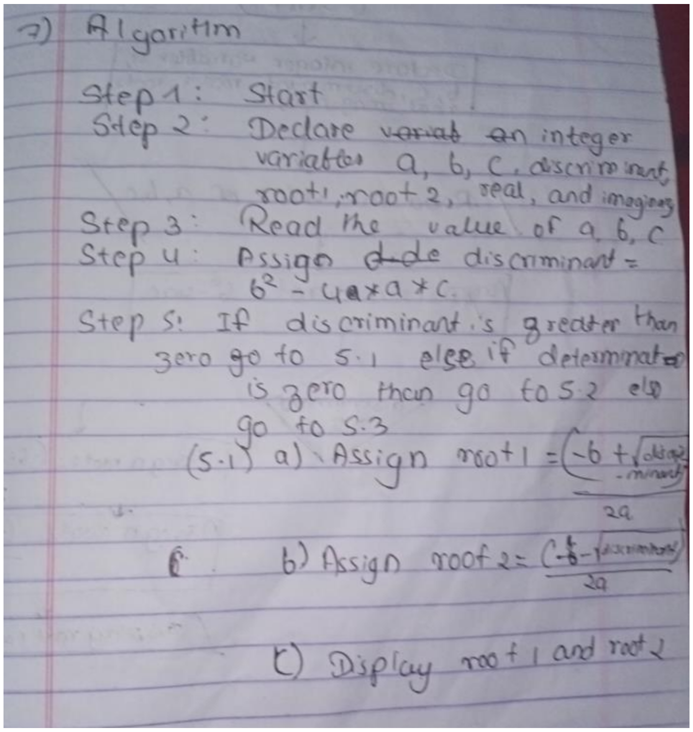
Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:8
This number is not divisible by both 5 and 11
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:55
This number is divisible by both 5 and 11
PS C:\Users\admin\OneDrive\Desktop\cprogram>
```

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

Algorithm:



7) Algaritim

Step 1: Start

Step 2: Declare ~~variab~~ ~~an~~ integer variables a, b, c, discriminant, root1, root 2, real, and imaginary

Step 3: Read the value of a, b, c

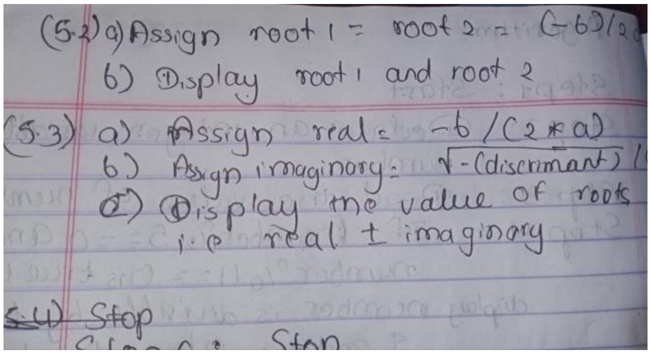
Step 4: Assign ~~d de~~ discriminant = $b^2 - 4 * a * c$.

Step 5: If discriminant is greater than zero go to 5.1 else if determinate is zero than go to 5.2 else go to 5.3

(5.1) a) Assign root1 = $\frac{(-b + \sqrt{discriminant})}{2a}$

b) Assign roof 2 = $\frac{(-b - \sqrt{discriminant})}{2a}$

c) Display root 1 and root 2



(5.2) a) Assign root 1 = root 2 = (-b)/2a

b) Display root 1 and root 2

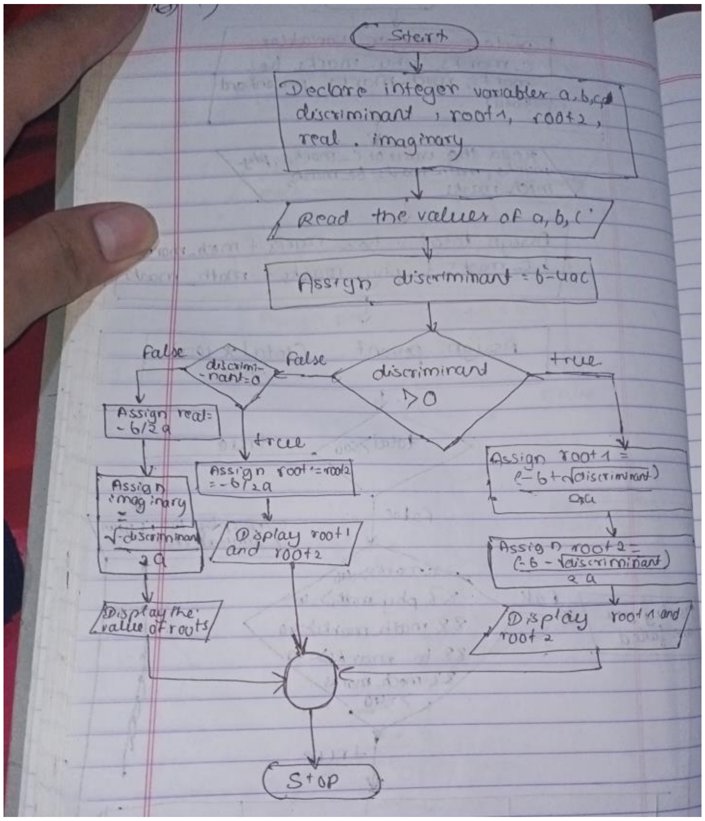
(5.3) a) Assign real = $-b / (2 * a)$

b) Assign imaginary = $\sqrt{-(\text{discrimant})}$ /

c) Display the value of roots i.e real ± imaginary

~~5.4)~~ Stop

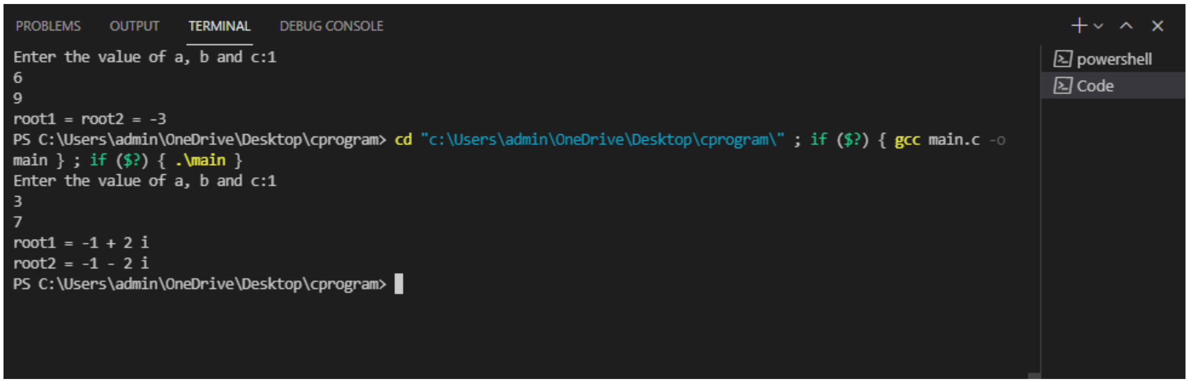
Flowchart:


Start
Declare integer variables a,b,c, discriminant, root1, root2, real, imaginary
Read the values of a, b, c
Assign discriminant = b²-4ac
discriminant >0
False
true
discriminant=0
False
true
Assign real= -b/2a
Assign imaginary = √-discriminant / 2a
Display the value of roots
Assign root1=root2 = -b/2a
Display root1 and root2
Assign root1 = (-b+√discriminant) / 2a
Assign root2 = (-b - √discriminant) / 2a
Display root1 and root2
Stop

Code:

```
#include<stdio.h>

#include<math.h>

int main()

{

  int a,b,c, discriminant, root1, root2, real, imaginary;

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

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

  discriminant = pow(b,2)-(4*a*c);

  if(discriminant >0)

  {

    root1 = (-b + sqrt(discriminant))/(2*a);

    root2 = (-b - sqrt(discriminant))/(2*a);
    printf("root1 = %d", root1);
    printf("root2 = %d", root2);

  }
  else if(discriminant ==0)
  {
    root1 = root2 =  (-b )/(2*a);
    printf("root1 = root2 = %d", root1);
  }
  else
  {
    real = -b/(2*a);
    imaginary = sqrt(-discriminant)/(2*a);
    printf("root1 = %d + %d i \n", real, imaginary);
    printf("root2 = %d - %d i", real, imaginary);

  }


  return 0;
}
```

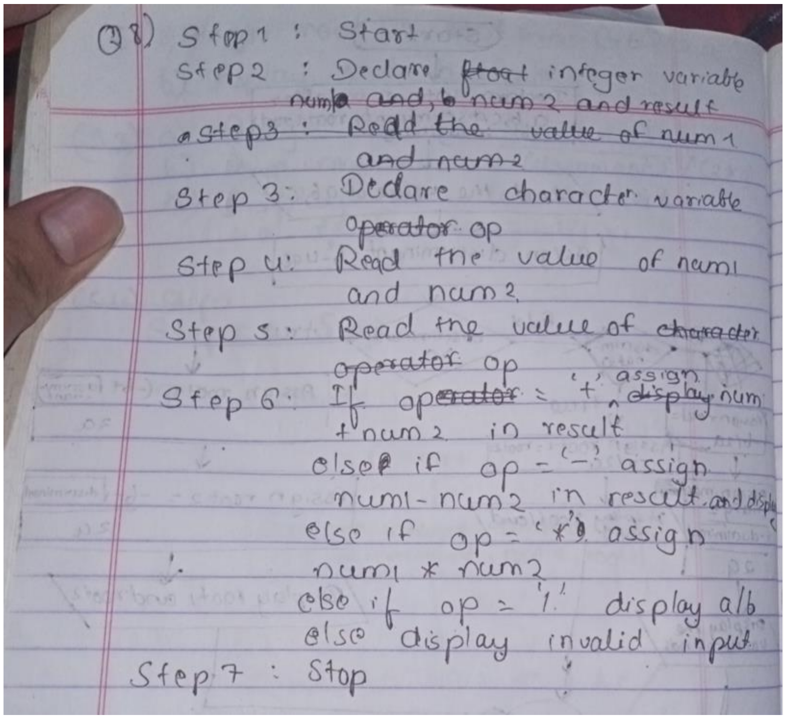
Output:



```
Enter the value of a, b and c:1
6
9
root1 = root2 = -3
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter the value of a, b and c:1
3
7
root1 = -1 + 2 i
root2 = -1 - 2 i
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

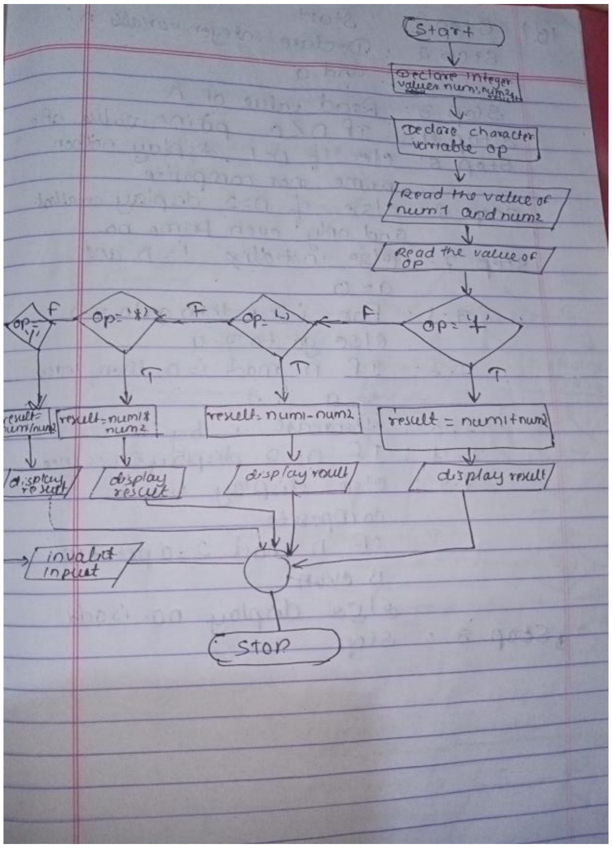
8.WAP to input two numbers and operator among [+ , -,* , /]. If user enters + then the program should perform the addition of the number and display the sum. If user enters–then the program should perform subtraction of number and display the difference and so on for * and /.

Algorithm:



```
Q8) Step1 : Start
    Step2 : Declare ~~float~~ integer variable
            num1 and, num 2 and result
    ~~Step3 : Read the value of num1
            and num2~~
    Step 3: Declare character variable
            ~~operator~~ op
    Step 4: Read the value of num1
            and num2.
    Step 5 : Read the value of ~~character~~
            ~~operator~~ op
    Step 6: If op~~erator~~ = '+' assign ~~display~~ num
            + num 2 in result
            else if op = '-' assign
            num1 - num2 in result and display
            else if op = '*' assign
            num1 * num2
            else if op = '/' display a/b
            else display invalid input
    Step 7 : Stop
```

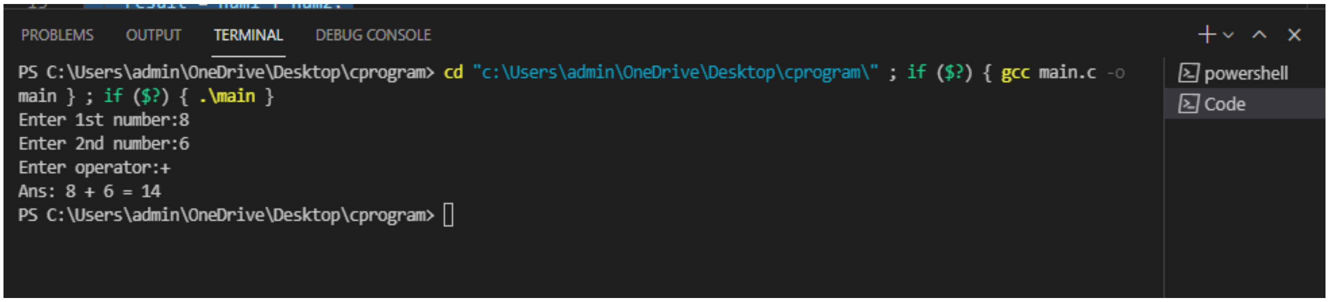
Flowchart:



Code:

```
#include<stdio.h>
int main()
{
  int num1, num2, result;
  char op;
  printf("Enter 1st number:");
  scanf("%d", &num1);
  printf("Enter 2nd number:");
  scanf("%d", &num2);
  printf("Enter operator:");
  scanf(" %c", &op);
  switch (op)
  {
  case '+':
    result = num1 + num2;
    printf("Ans: %d %c %d = %d", num1, op, num2,result);
    break;
  case '-':
    result = num1 - num2;
    printf("Ans: %d %c %d = %d", num1, op, num2,result);
    break;
  case '*':
    result = num1 * num2;
    printf("Ans: %d %c %d = %d", num1, op, num2,result);
    break;
  case '/':
    result = num1 / num2;
    printf("Ans: %d %c %d = %d", num1, op, num2,result);
    break;

  default:
    printf("Please Enter proper sign.");
    break;
  }

}
```

Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter 1st number:8
Enter 2nd number:6
Enter operator:+
Ans: 8 + 6 = 14
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

9) WAP in C to input marks of five subjects C-programming, Physics, Maths, 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

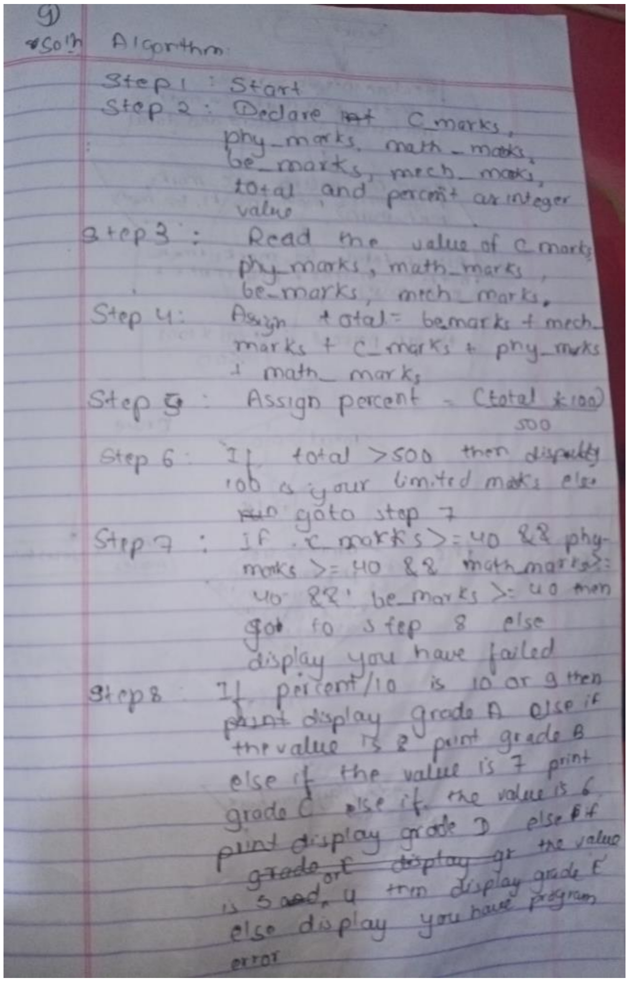
Percentage >=80% : B

Percentage >= 70% : C

Percentage >= 60% : D

Percentage >= 40% : E

Algorithm:



9)

Sol'n Algorithm:

Step 1 : Start

Step 2 : Declare ~~int~~ C_marks, phy_marks, math_marks, be_marks, mech_marks, total and percent as integer value

Step 3 : Read the value of C_marks, phy_marks, math_marks, be_marks, mech_marks.

Step 4: Assign total = be_marks + mech_marks + C_marks + phy_marks + math_marks

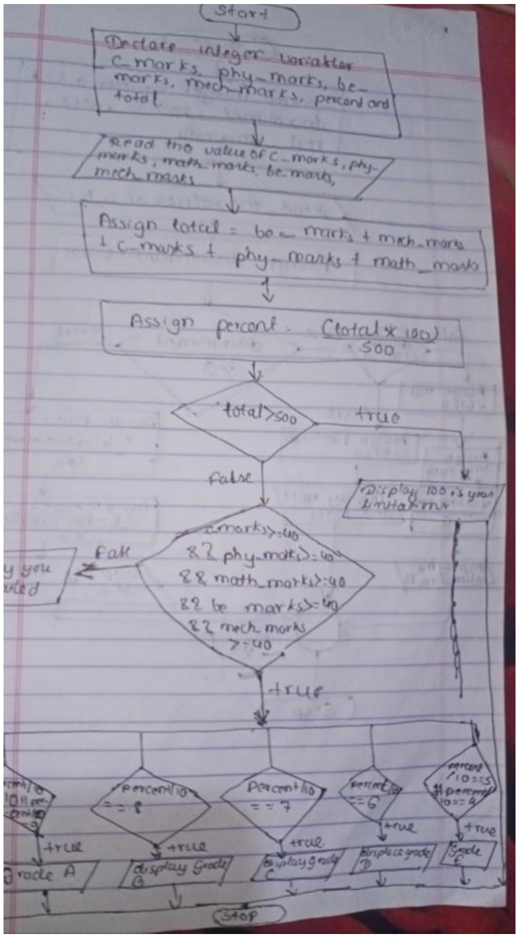
Step 5 : Assign percent = $\frac{(total * 100)}{500}$

Step 6 : If total > 500 then display 100 is your limited marks else ~~run~~ goto step 7

Step 7 : if c_marks >= 40 && phy_marks >= 40 && math_marks >= 40 && be_marks >= 40 then got to step 8 else display you have failed

Step 8 : If percent/10 is 10 or 9 then ~~print~~ display grade A else if the value is 8 print grade B else if the value is 7 print grade C else if the value is 6 print display grade D else if ~~grade E display gr~~ the value is 5 ~~and~~ or 4 then display grade E else display you have program error

Flowchart:

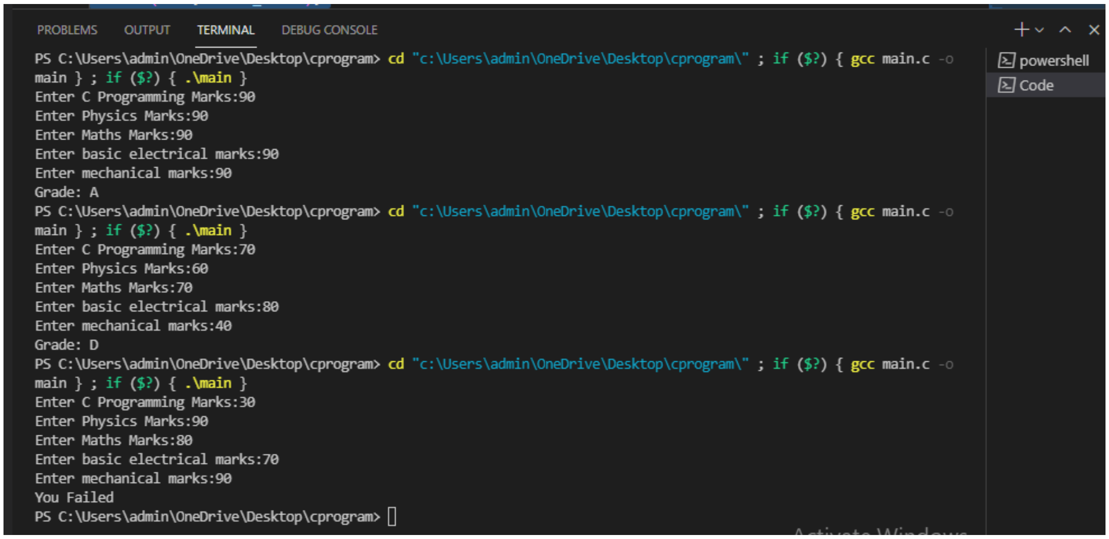

Start
Declare integer variables c_marks, phy_marks, be_marks, mech_marks, percent and total
Read the value of c_marks, phy_marks, math_marks, be_marks, mech_marks
Assign total = be_marks + mech_marks + c_marks + phy_marks + math_marks
Assign percent (total × 100) / 500
total>500
true
false
Display 100 is your limitation
c_marks>=40 && phy_marks>=40 && math_marks>=40 && be_marks>=40 && mech_marks>=40
false
y you
iled
true
percent/10 == 8
percent/10 == 7
percent/10 == 6
percent/10 == 5 || percent/10 == 4
true
Grade A
display Grade B
display grade C
display grade D
Grade E
Stop

Code:

```
#include<stdio.h>

int main()

{

  int c_marks, phy_marks, math_marks, be_marks,mech_marks,total, percent;

  printf("Enter C Programming Marks:");

  scanf("%d", &c_marks);

  printf("Enter Physics Marks:");

  scanf("%d", &phy_marks);

  printf("Enter Maths Marks:");

  scanf("%d", &math_marks);

  printf("Enter basic electrical marks:");
```

```
scanf("%d", &be_marks);
printf("Enter mechanical marks:");
scanf("%d", &mech_marks);
total = be_marks + mech_marks + c_marks + phy_marks + math_marks;
percent = (total* 100)/500;
if(total > 500){
  printf("100 is your limited marks");
}
else{
  if(c_marks>=40 && phy_marks >=40 && math_marks>=40 && mech_marks >=40 && be_marks>=40)
{
  switch (percent/10)
{
case 10:
case 9:
  printf("Grade: A");
  break;
case 8:
  printf("Grade: B");
  break;
case 7:
  printf("Grade: C");
  break;
case 6:
  printf("Grade: D");
  break;
case 5:
case 4:
   printf("Grade: E");
   break;


  default:
   printf("You have program error");
   break;
  }
  }
  else{
   printf("You Failed");
  }


  }
  return 0;
}
```

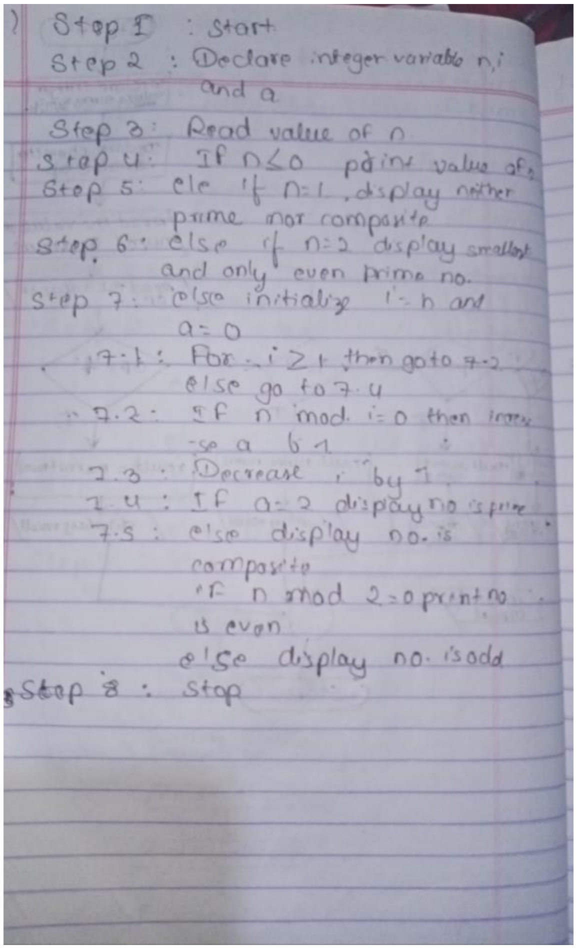
Output:



```
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter C Programming Marks:90
Enter Physics Marks:90
Enter Maths Marks:90
Enter basic electrical marks:90
Enter mechanical marks:90
Grade: A
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter C Programming Marks:70
Enter Physics Marks:60
Enter Maths Marks:70
Enter basic electrical marks:80
Enter mechanical marks:40
Grade: D
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter C Programming Marks:30
Enter Physics Marks:90
Enter Maths Marks:80
Enter basic electrical marks:70
Enter mechanical marks:90
You Failed
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```

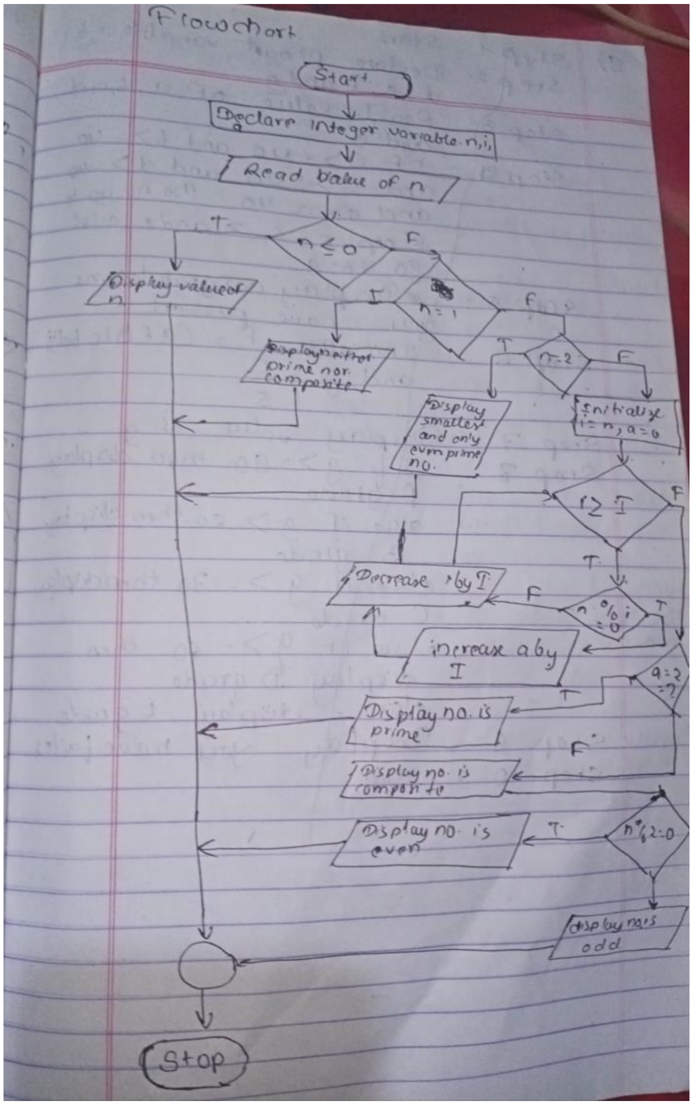
10.WAP to input a number from user. If user enters a number less than or equal to zero then program should just display the number. If user enters 1 the program should display output as neither prime nor composite, if user enters 2 the program should display output as smallest and only even prime number. If user enters any number greater than 2 the program should check whether the number is prime or not, also if the number is not prime the program should display whether it is even or odd.

Algorithm:



```
Step 1 : Start
Step 2 : Declare integer variable n,i and a
Step 3: Read value of n
Step 4: If n≤0 print value of n
Step 5: ele if n=1, display neither prime nor composite
Step 6: else if n=2 display smallest and only even prime no.
Step 7: else initialize i=n and a=0
    7.1: For i≥1 then goto 7.2 else go to 7.4
    7.2: If n mod i=0 then increase a by 1
    7.3: Decrease i by 1
    7.4: If a=2 display no is prime
    7.5: else display no. is composite
         if n mod 2=0 print no is even
         else display no. is odd
Step 8: Stop
```

Flowchart:

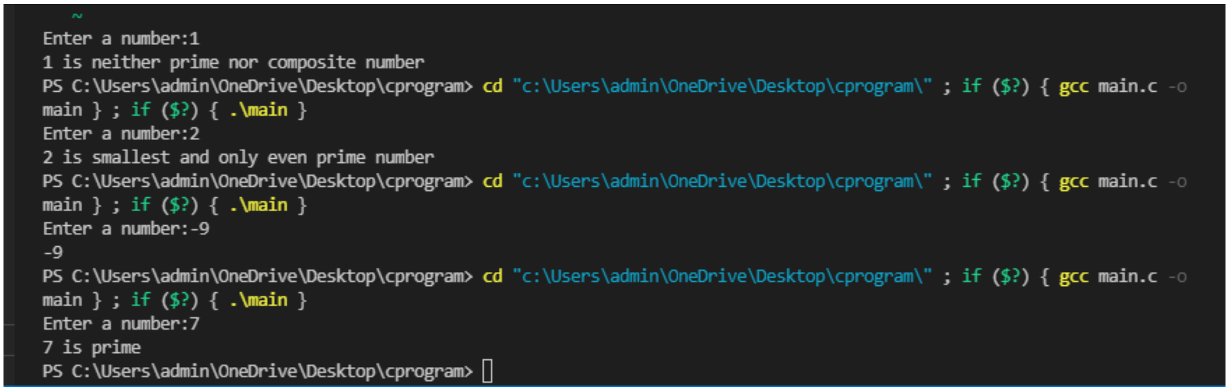


Code:

```
#include <stdio.h>

int main()

{

  int n, i, a = 0;

  printf("Enter a number:");

  scanf("%d", &n);

  if (n <= 0)

  {

    printf("%d", n);
```

```
}
else if (n == 1)
{
  printf("%d is neither prime nor composite number", n);
}
else if (n == 2)
{
  printf("%d is smallest and only even prime number", n);
}
else
{
  for (i = n; i >= 1; i--)
  {
    if ((n % i) == 0)
      a++;
  }
  if (a == 2)
    printf("%d is prime", n);
  else
  {
    printf("%d is composite\n", n);
    if ((n % 2) == 0)
    {
      printf("%d is even number.", n);
    }
    else
    {
      printf("%d is odd number", n);
      }
    }
  }
  return 0;
}
```

Output:



```
   ~
Enter a number:1
1 is neither prime nor composite number
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:2
2 is smallest and only even prime number
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:-9
-9
PS C:\Users\admin\OneDrive\Desktop\cprogram> cd "c:\Users\admin\OneDrive\Desktop\cprogram\" ; if ($?) { gcc main.c -o
main } ; if ($?) { .\main }
Enter a number:7
7 is prime
PS C:\Users\admin\OneDrive\Desktop\cprogram> 
```