C – Programming

>> System Specifications :

Processor > Ryzen 5 5600h (6 cores 12 threads) 3.30Ghz base speed

Memory > 8GB ram

Graphics > NVIDIA GeForce RTX 3050 (4gb Vram)

Storage > 512Gb internal SSD

Software : Visual Studio Code (Version 1.93)

>> Name – Sankalp Pandey

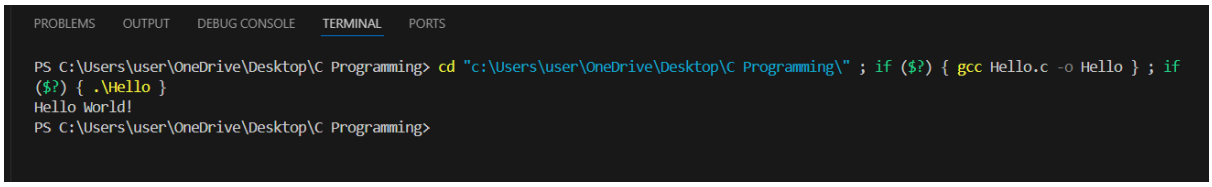
>>Batch – 71

>>Sap_ID- 590015862

C - Programs :

1) Print "Hello World!"

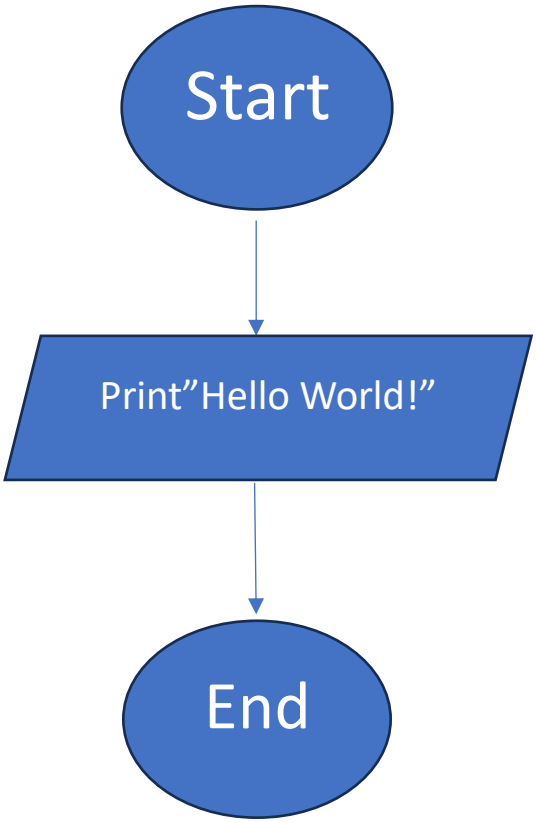
Output :



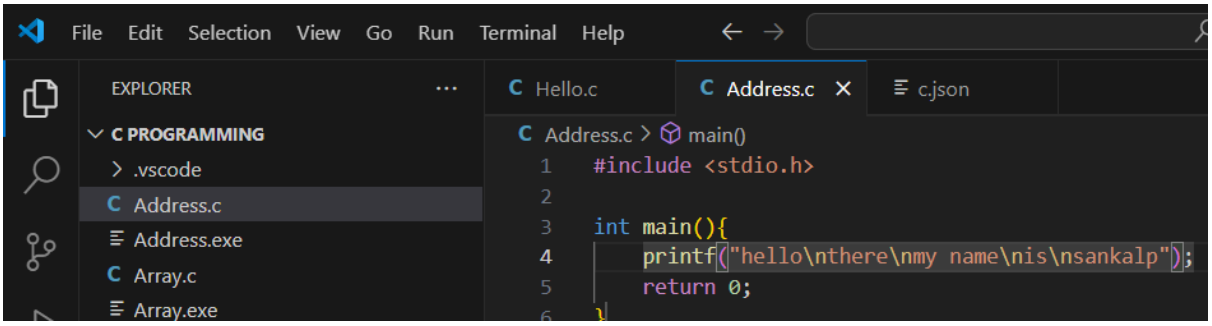
Algorithm :

1) Start
2) Include stdio.h (header)
3) Starting main () function
4) Printing the statement “Hello World!”
5) End

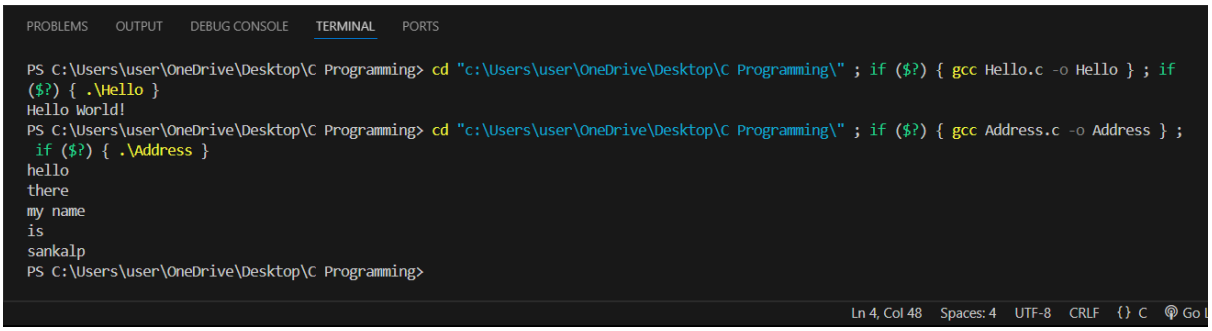
FLOW CHART :



2) Print the address in multiple lines (new line)



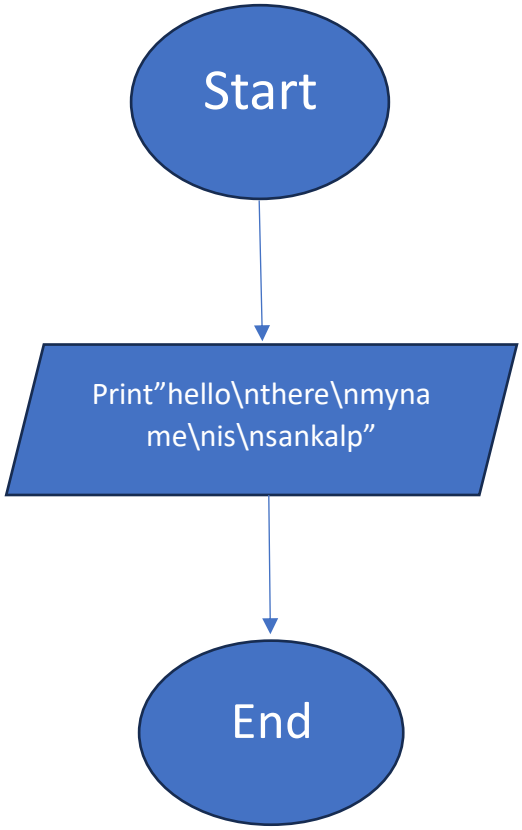
Output :



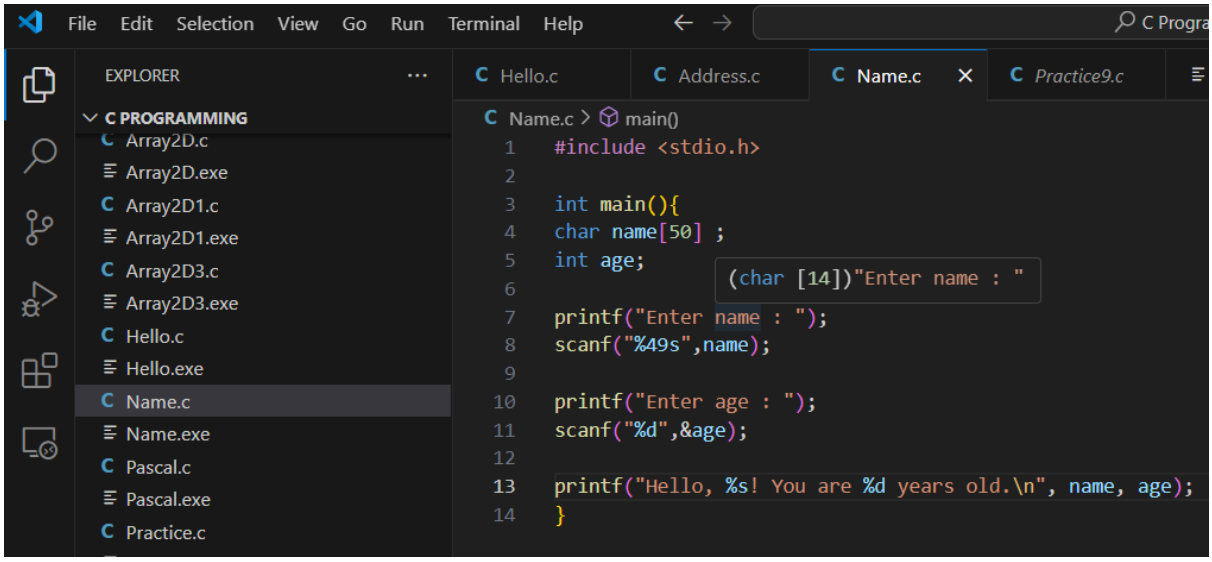
Algorithm :

1) Start
2) Include header file
3) Starting main() function
4) Enter address (use printf)
5) End

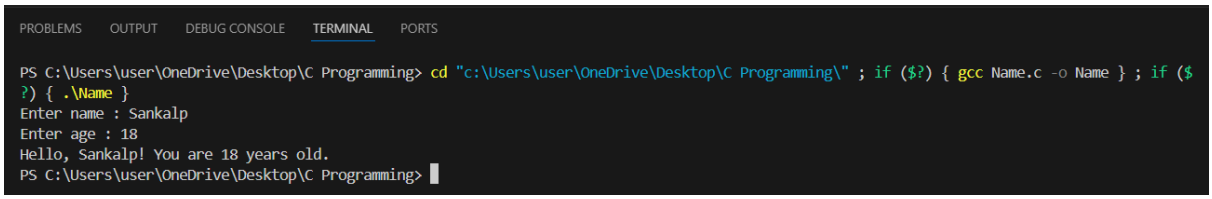
FLOW CHART :



3) that prompts the user to enter their name and age.



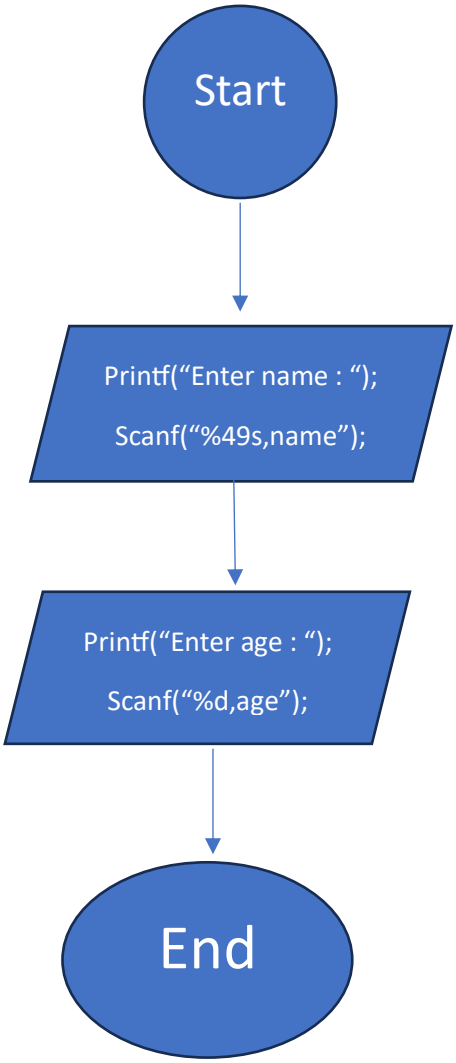
Output :



Algorithm :

1) Start
2) Include header file
3) Starting main() function
4) Use an input function to ask user to input their name and store it in a string variable.
5) Use an input function to ask the user to input their age and store it in an integer variable.
6) End

Flow Chart :



4) add two numbers, take number from user.

Terminal Help C Programming

Hello.c | Address.c | Name.c | Sum.c | c.json

Sum.c > main()

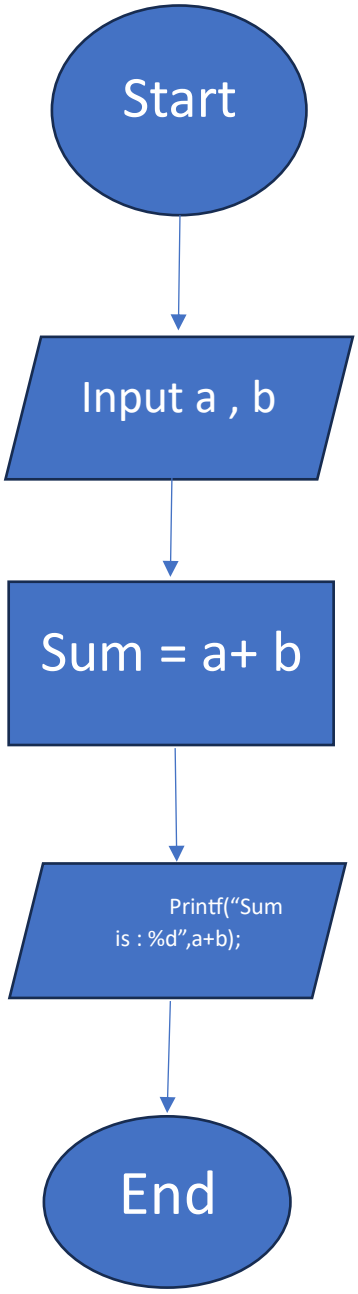
```c
# include <stdio.h>
int main()
{
int a,b;
printf("Enter a =");
scanf("%d",&a);

printf("Enter b =");
scanf("%d",&b);

printf("Sum is : %d",a + b);
return 0;
}
```

Output :

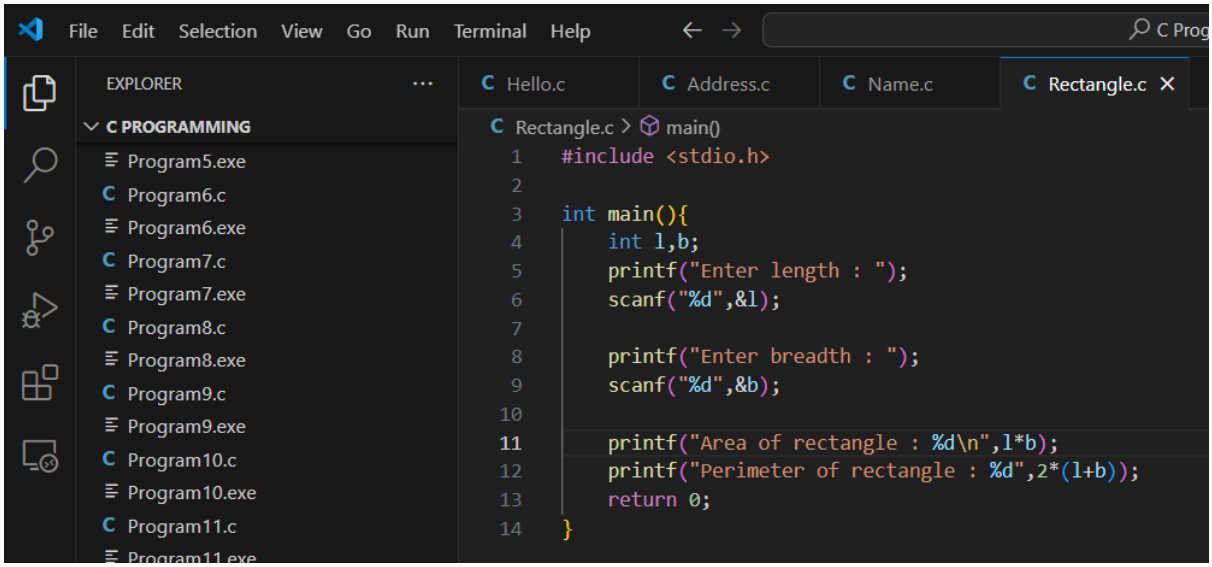
Algorithm :

1) Start
2) Include header file
3) Use two inputs a and b to calculate their sum
4) Perform basic calculation as sum = a+b
5) End

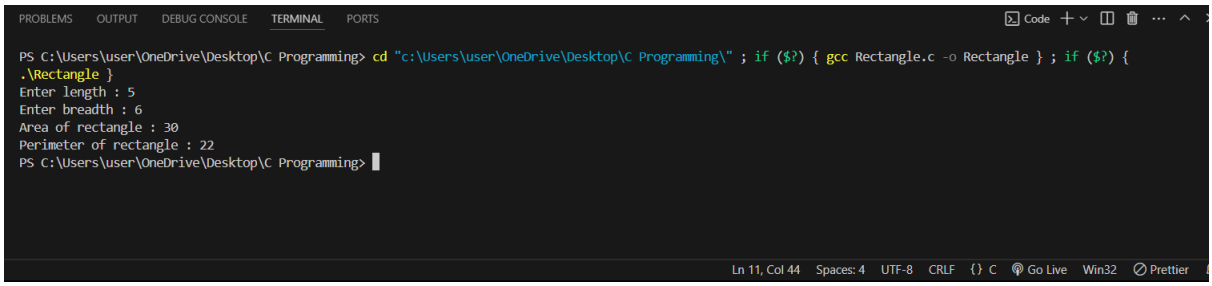
Flowchart :



5) calculate the area and perimeter of a rectangle based on the input length and width.

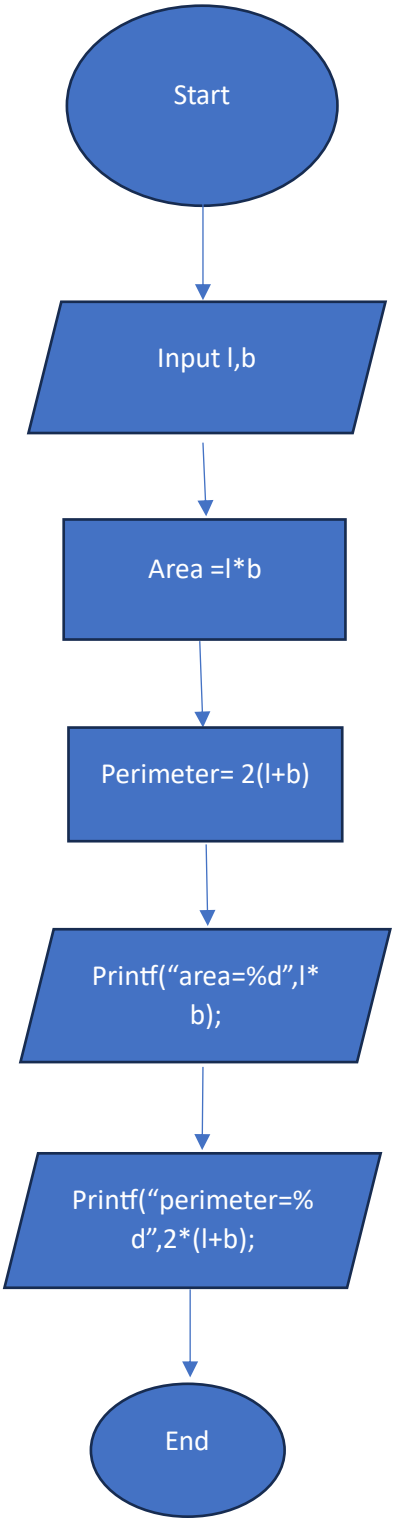


Output :



Algorithm :

1) Start
2) Include header file
3) Read length , breadth
4) Calculate area = length * breadth
5) Calculate perimeter as = 2 * (length + breadth)
6) Print area and perimeter
7) End


Start
Input l,b
Area =l*b
Perimeter= 2(l+b)
Printf(“area=%d”,l*
b);
Printf(“perimeter=%
d”,2*(l+b);
End

6) convert temperature from Celsius to Fahrenheit using the formula : F= (C*9/5)+32

Terminal Help C Programming

Hello.c | Address.c | Name.c | Rectangle.c | Temperature.c

Temperature.c > main()

```c
#include <stdio.h>

int main(){
    float celsius,fahrenheit;
    printf("Enter temperature in celcius : ");
    scanf("%f",&celsius);

    fahrenheit = (celsius * 9 / 5) + 32;

    printf("Temperature in Fahrenheit: %0.2f\n",fahrenheit);
    return 0;
}
```

Output :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code

```
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc Temperature.c -o Temperature } ; if ($?
) { .\Temperature }
Enter temperature in celcius : 45
Temperature in Fahrenheit: 113.00
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Ln 10, Col 61 Spaces: 4 UTF-8 CRLF {} C Go Live Win32

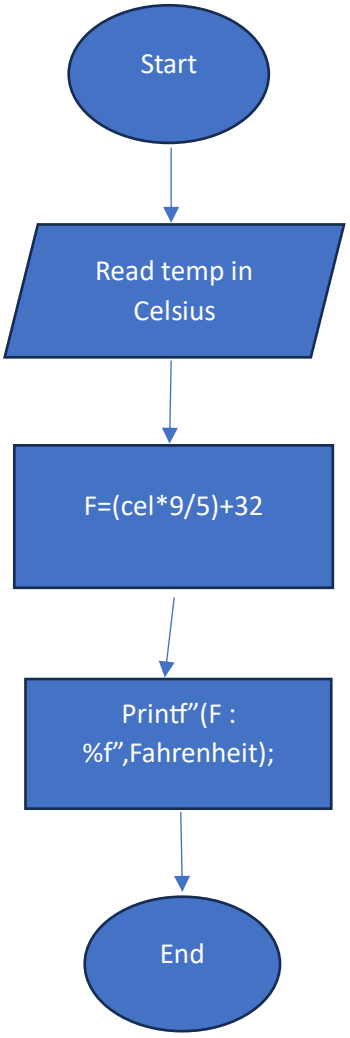
Algorithm :

1) Start
2) Include header file
3) Read temperature in cels ius
4) Calculate the temperature from Celsius to fahrenheit using formula f=(cel*9/5)+32

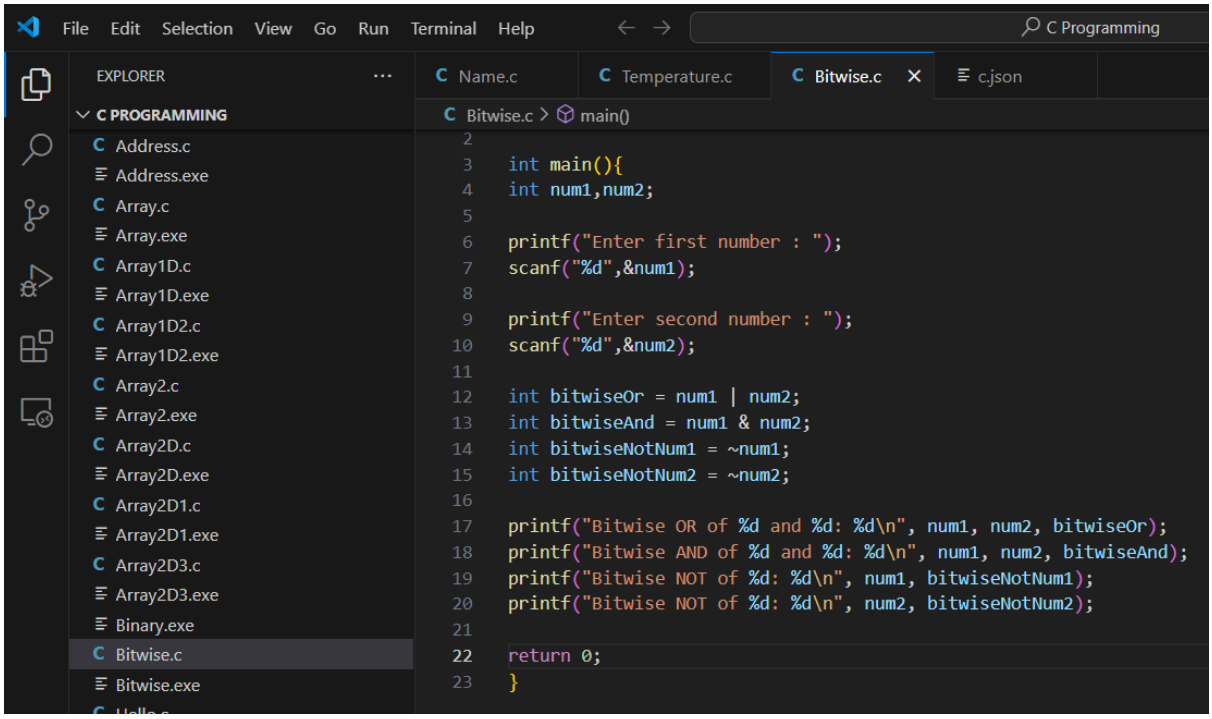
5) Print the temperature in Fahrenheit

6) End

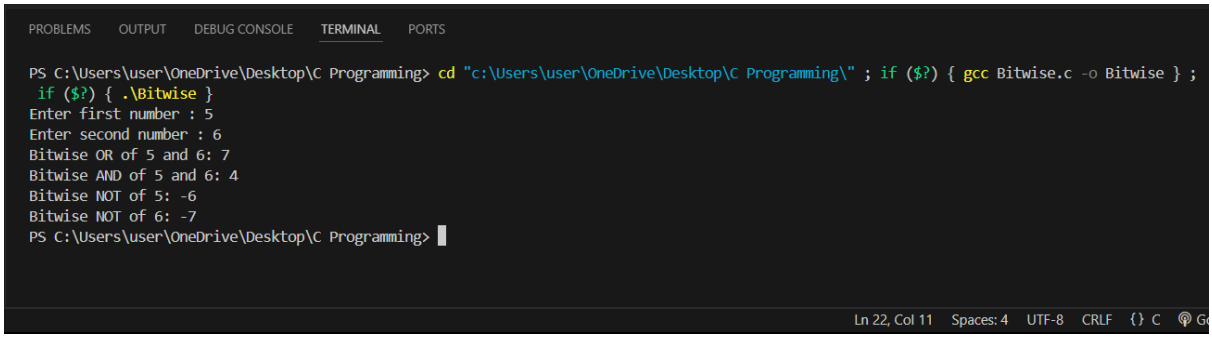
Flowchart :



7) Write a C program to apply bitwise OR, AND, and NOT operators on bit level.

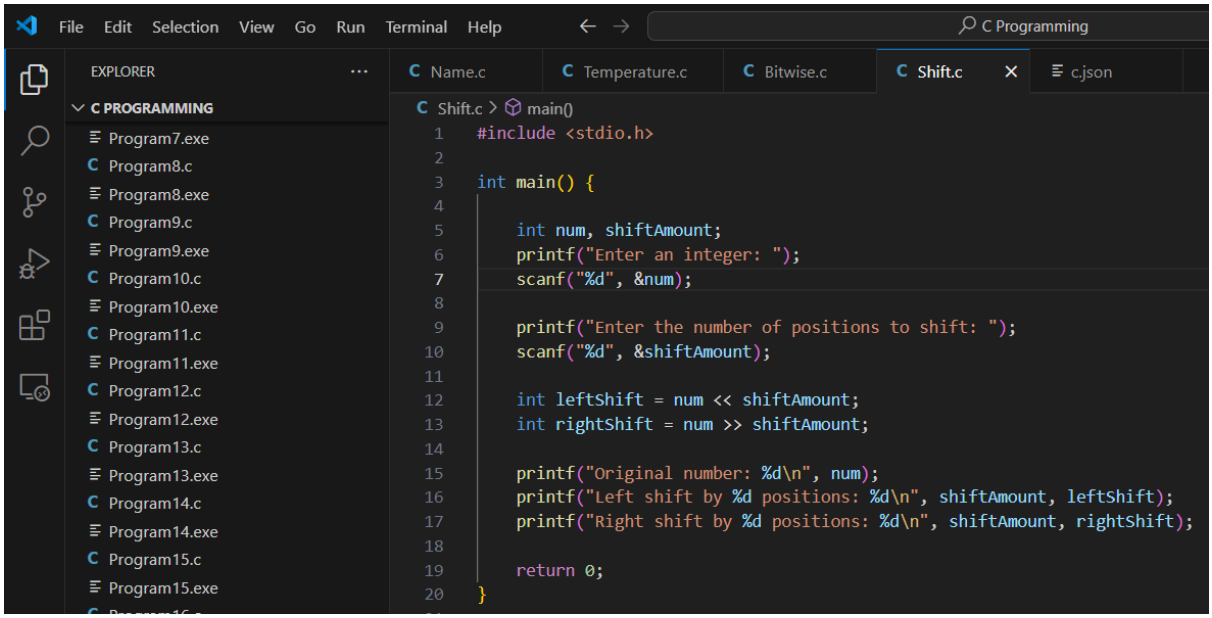


Output :

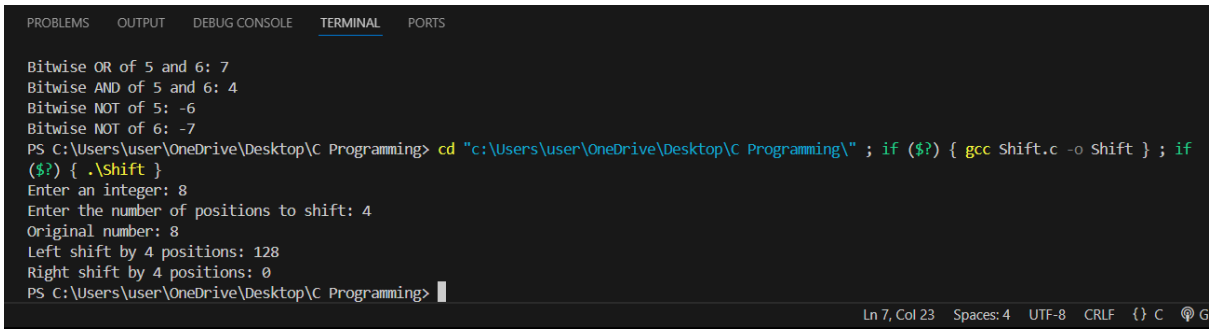


Algorithm :

8) Write a C program to apply left and right shift operators.

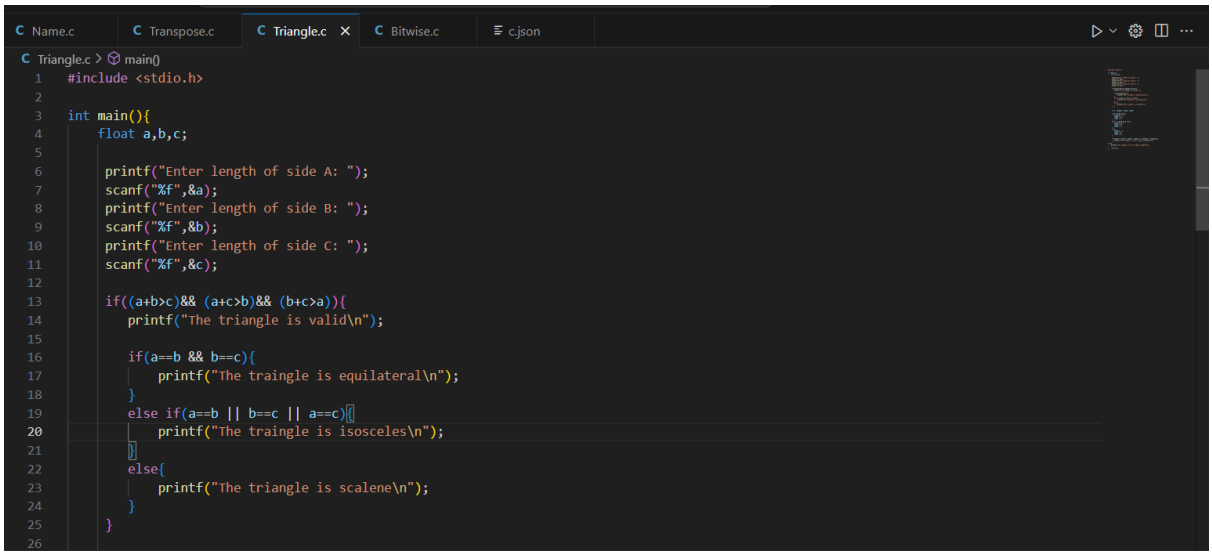


Output :



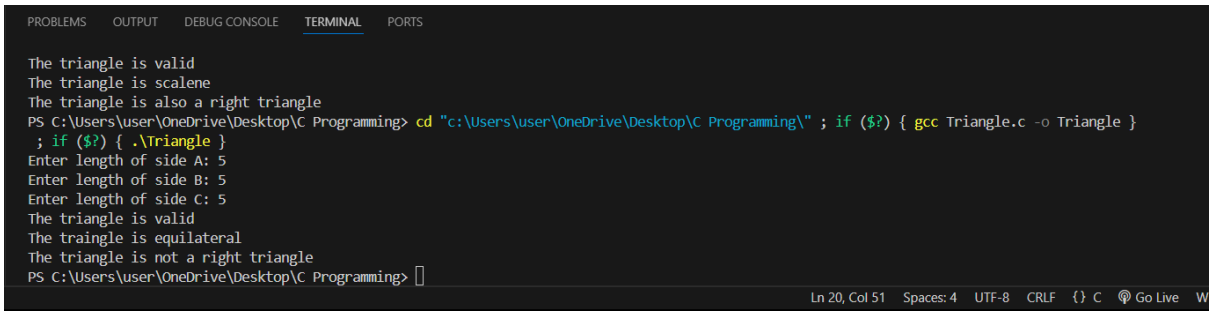
Experiment : Conditional Statements

1. WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle, or scalene. Take sides of the triangle as input from a user.



```c
#include <stdio.h>

int main(){
    float a,b,c;

    printf("Enter length of side A: ");
    scanf("%f",&a);
    printf("Enter length of side B: ");
    scanf("%f",&b);
    printf("Enter length of side C: ");
    scanf("%f",&c);

    if((a+b>c)&& (a+c>b)&& (b+c>a)){
       printf("The triangle is valid\n");

       if(a==b && b==c){
           printf("The traingle is equilateral\n");
       }
       else if(a==b || b==c || a==c){
           printf("The traingle is isosceles\n");
       }
       else{
           printf("The triangle is scalene\n");
       }
    }
```

Output :



```
The triangle is valid
The triangle is scalene
The triangle is also a right triangle
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc Triangle.c -o Triangle }
 ; if ($?) { .\Triangle }
Enter length of side A: 5
Enter length of side B: 5
Enter length of side C: 5
The triangle is valid
The traingle is equilateral
The triangle is not a right triangle
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Step 1: Declare three variables a,b,c of triangle.

Step 2: Enter three sides .

Step 3: If a == b && b == c Go to step 6

Step 4: If a == b || b == c|| c== a Go to Step 7

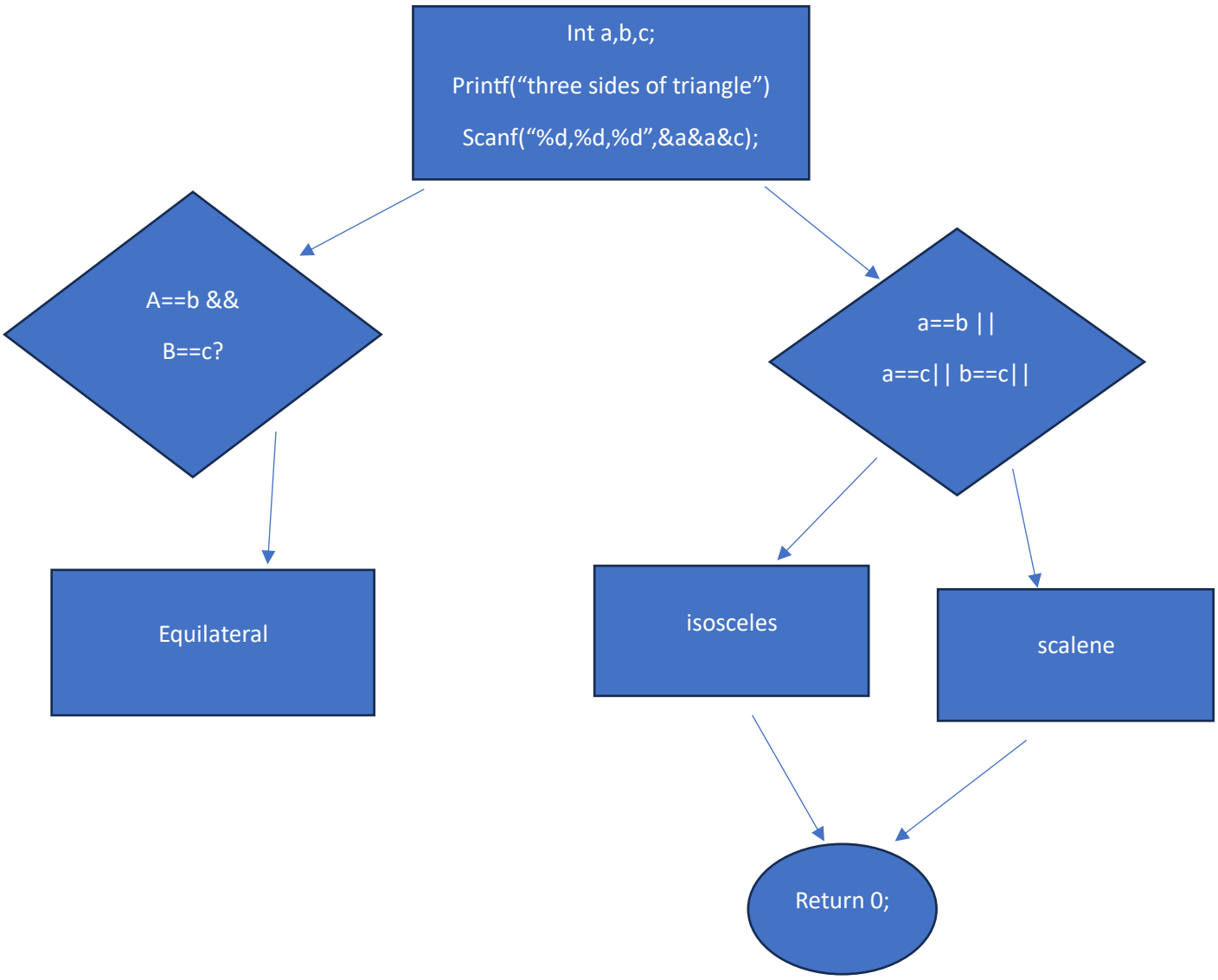
Step 5: else Go to step 8

Step 6: Print the triangle is equilateral.

Step 7: Print the triangle is isosceles.

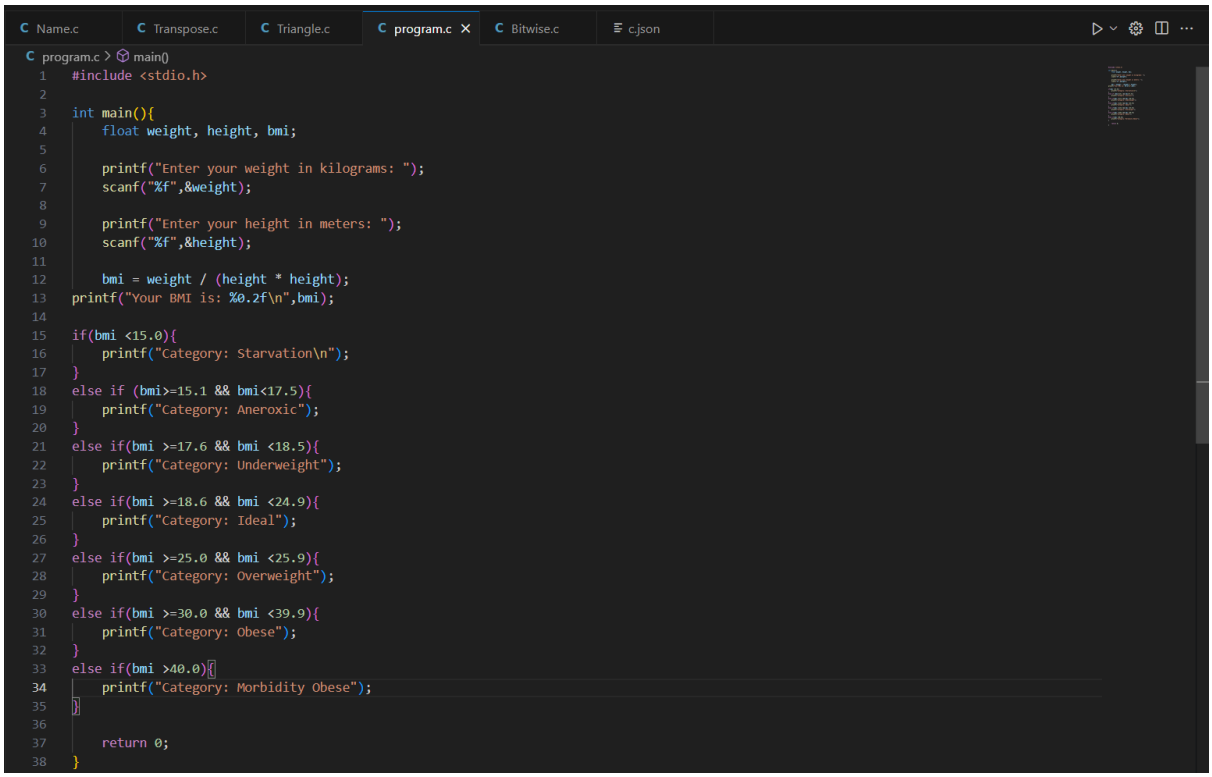
Step 8: Print the triangle is scalene.

Flowchart :

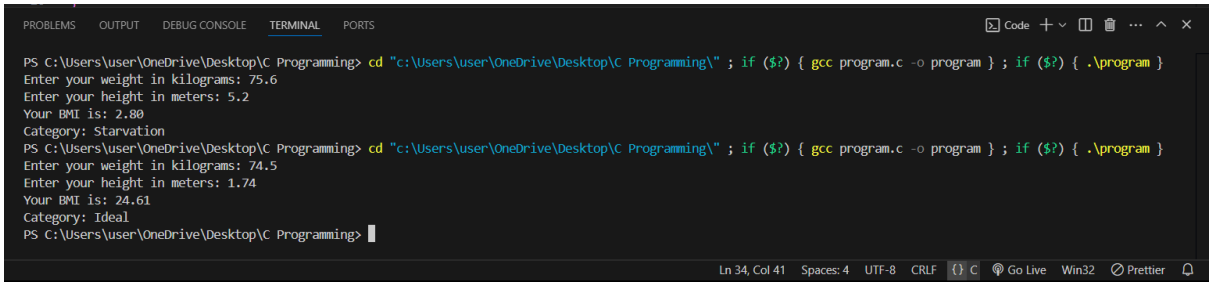


2. WAP to compute the BMI Index of the person and print the BMI values as per the following ranges. You can use the following formula to compute BMI= weight(kgs)/Height(Mts)*Height(Mts). BMI Starvation is

	BMI
Starvation	<15
Anorexic	15.1 to 17.5
Underweight	17.6 to 18.5
Ideal	18.6 to 24.9
Overweight	25 to 25.9
Obese	30 to 39.9
Morbidity Obese	40 above



Output :



Algorithm :

Step 1:Start

Step2:input weight in kgs and height in meters(in float)

Step3:use bmi formula : bmi=weight/(height*height)

Step4: check conditions If bmi<=15 print “starvation” Else if15.1=<=17.5 print” anorexic” Else if17.6=<=18.5 print” underweight” Else if18.6=<=24.9print”ideal” Else if25=<=25.9print”overweight” Else if30=<=39.9 print” obese” Else Print” Morbidity Obese”

Step 5-display the bmi category which matches the condition .

Step6-End

Flowchart :


Start
Input three points of coordinates
X1,y1,x2,y2,x3,y3
If collinear==0
Print “the points are collinear”
Print “ the points are not collinear”
Return 0;

3. WAP to check if three points (x1,y1), (x2,y2) and (x3,y3) are collinear or not.

Name.c | Transpose.c | Triangle.c | Temperature.c | program.c | Collinear.c | Bitwise.c | c.json

Collinear.c > main()

```c
#include <stdio.h>

int main(){
    float x1,y1,x2,y2,x3,y3;

    printf("Enter coordinates of point 1 (x1, y1): ");
    scanf("%f%f",&x1, &y1);
printf("Enter coordinates of point 2 (x2, y2): ");
scanf("%f%f",&x2, &y2);
printf("Enter coordinates of point 3 (x3, y3): ");
scanf("%f%f",&x3, &y3);

float area = x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2);
printf("Area = %.2f\n",area);

if(area == 0){
    printf("The points are collinear\n");
}
else{
    printf("The points are not collinear\n");
}
    return 0;
}
```

Output :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Area = 0.00The points are collinear
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc Collinear.c -o Collinear } ; if ($?) { .\Collinear
 }
Enter coordinates of point 1 (x1, y1): 2
3
Enter coordinates of point 2 (x2, y2): 4
5
Enter coordinates of point 3 (x3, y3): 6
7
Area = 0.00
The points are collinear
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Ln 19, Col 6 Spaces: 4 UTF-8 CRLF {} C Go Live Win32 Prettier

Algorithm :

Step1-Start

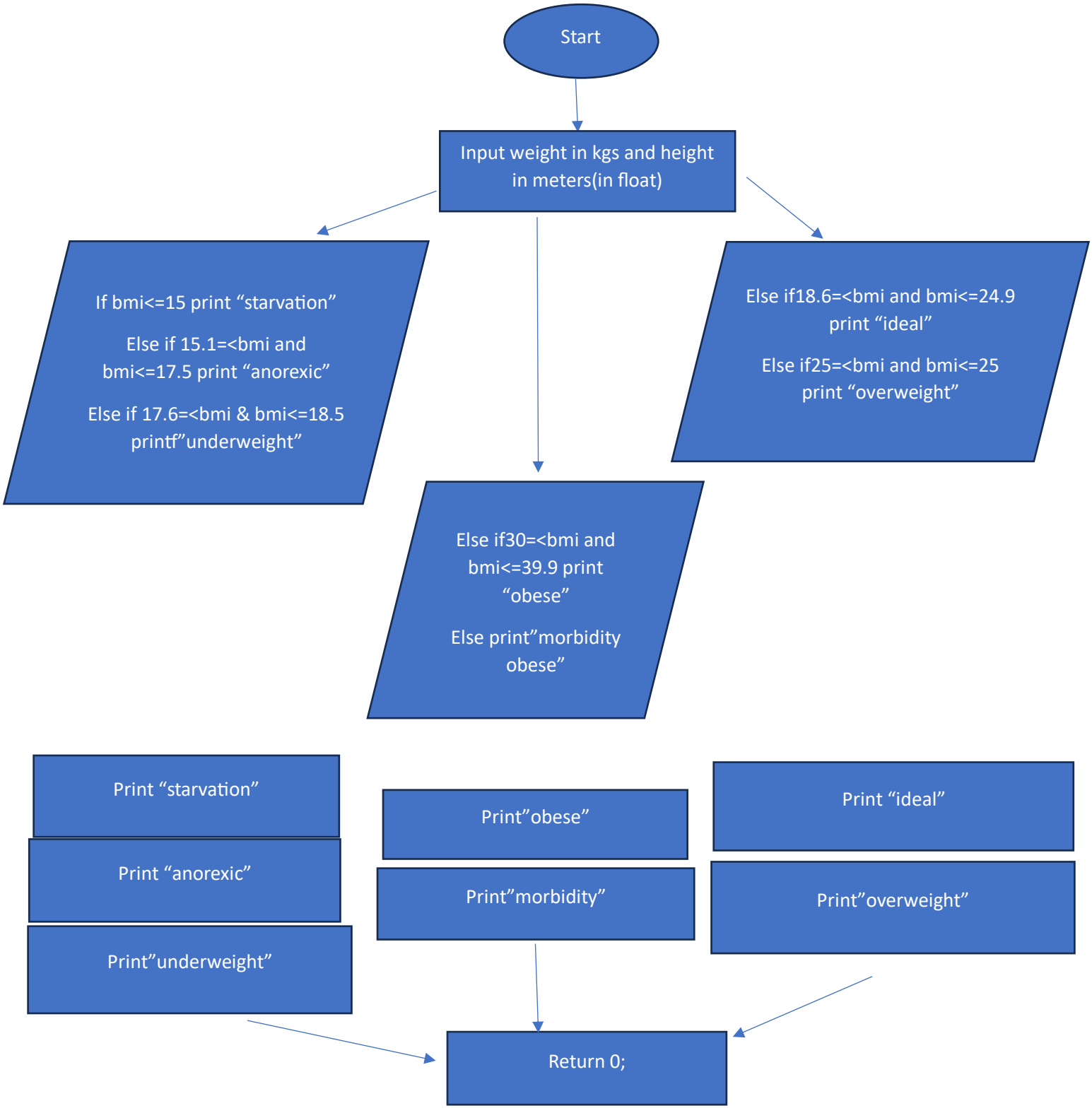
Step2-input all three points of coordinates x1,y1,x2,y2,x3,y3

Step3-use formula – ((x1 * (y2- y3) + x2 * (y3- y1) + x3 * (y1- y2))

Step4-check condition If(collinear==0) Print “points are collinear” Else Print ”the points are not collinear”

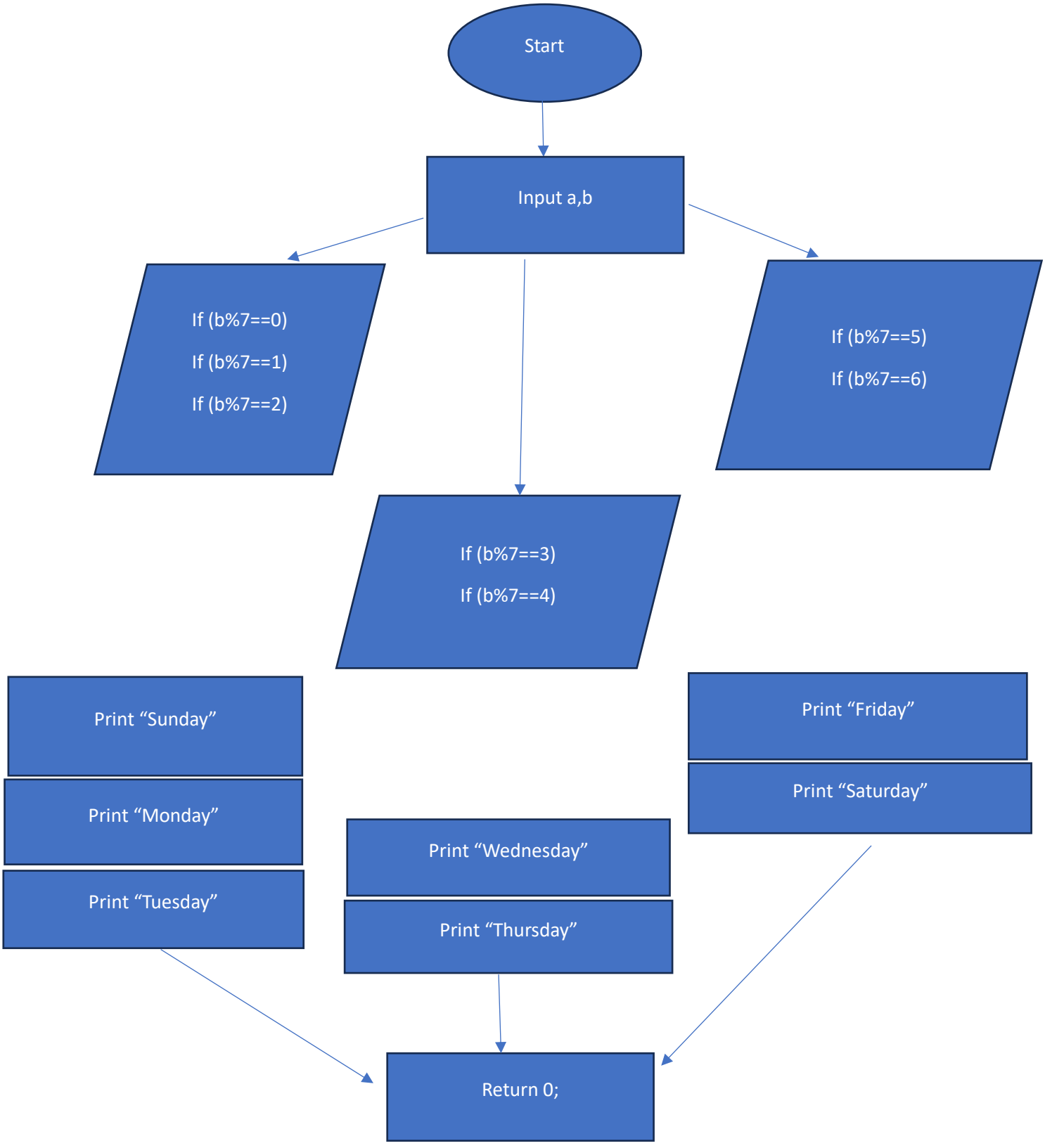
Step5- End

Flowchart :



4. According to the gregorian calendar, it was Monday on the date 01/01/01. If Any year is input through the keyboard write a program to find out what is the day on 1st January of this year.

Flowchart :



```c
#include <stdio.h>

int main(){
    int yearGiven = 2001;

    int year;
    printf("Enter the year: ");
    scanf("%d",&year);

    int difference = year - yearGiven;

    int leapYear = difference/4;

    int nonLeapYear = difference - leapYear;

    int days = (leapYear * 366) + (nonLeapYear * 365) +1;

    if(days % 7 ==0)
    {
        printf("Sunday");
    }
    if(days % 7 ==1)
    {
        printf("Monday");
    }
    if(days % 7 ==2)
    {
        printf("Tuesday");
    }
    if(days % 7 ==3)
    {
        printf("Wednesday");
    }
    if(days % 7 ==4)
    {
        printf("Thursday");
    }
    if(days % 7 ==5)
    {
```

Output :

```
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc calendar.c -o calendar } ; if ($?) { .\calendar }
Enter the year: 2005
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc calendar.c -o calendar } ; if ($?) { .\calendar }
Enter the year: 2005
Saturday
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc calendar.c -o calendar } ; if ($?) { .\calendar }
Enter the year: 2003
Wednesday
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Algorithm :

Step1-Start

Step2-input a, b (a for year ,b to calculating, what is the day on 1st January of this year)

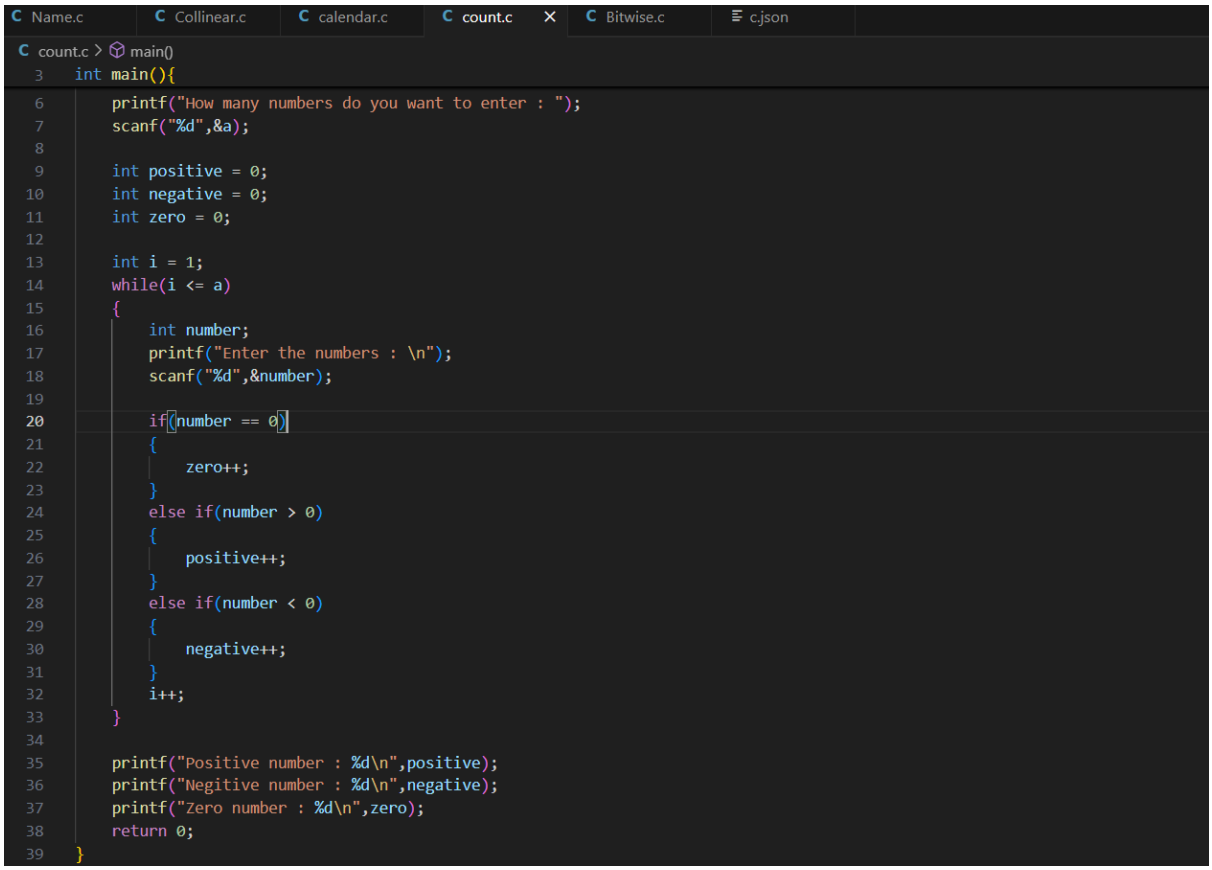
Step3-use formula –) b=(a+(a/4)-(a/100)+(a/400))%7

Step4-check condition If(b%7==0) Print” Sunday” If(b%7==1)Print” Monday” If(b%7==2)print” Tuesday” If(b%7==3)print” Wednesday” If(b%7==4)print “Thursday” If(b%7==5)print “Friday” If(b%7==6)print “Saturday”

Step5- End

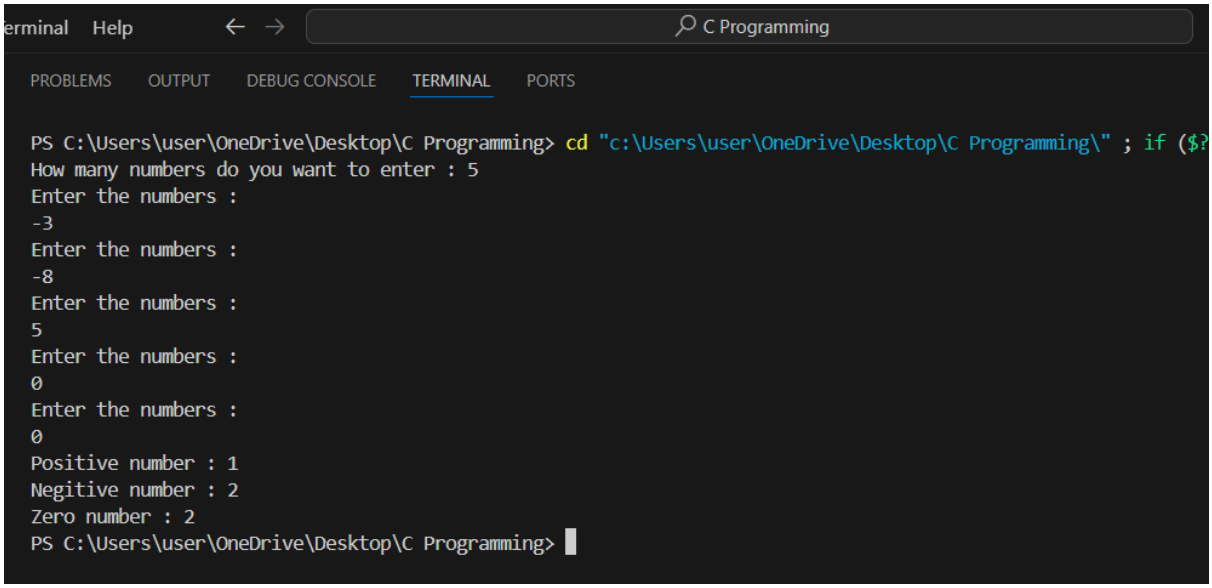
Experiment : Loops

1. WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative, and Zeroes entered.



```c
int main(){
    printf("How many numbers do you want to enter : ");
    scanf("%d",&a);

    int positive = 0;
    int negative = 0;
    int zero = 0;

    int i = 1;
    while(i <= a)
    {
        int number;
        printf("Enter the numbers : \n");
        scanf("%d",&number);

        if(number == 0)
        {
            zero++;
        }
        else if(number > 0)
        {
            positive++;
        }
        else if(number < 0)
        {
            negative++;
        }
        i++;
    }

    printf("Positive number : %d\n",positive);
    printf("Negitive number : %d\n",negative);
    printf("Zero number : %d\n",zero);
    return 0;
}
```

Output :



Algorithm :

Step1-Start

Step2-input a for how many number you want to enter

Step3-take Initialize counters: positive=0 negative =0 zero =0

Step4-start a loop for accepting numbers

Step5-now input the numbers inside loop

Step6-check condition

if number>0 then positive ++;

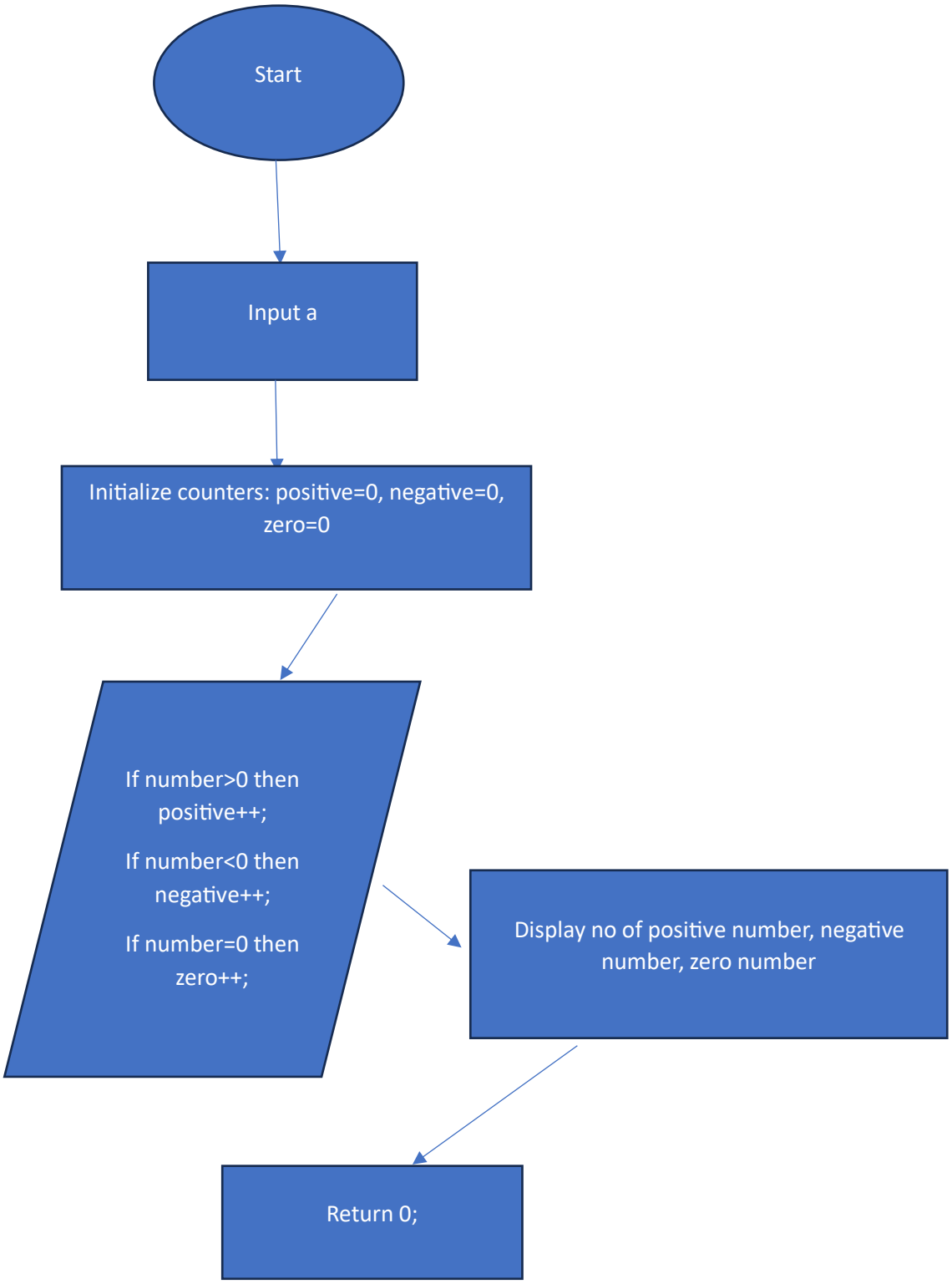
if number<0 then negative++;

if number=0 then zero++;

Step-8now display the output after loop ends it will give you number of positive number negative numbers and number of zeros in the code

Step-9-End

Flowchart :



WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting. Num * 1 = Num.

Terminal Help C Programming

Name.c | Collinear.c | calendar.c | count.c | Multiplication.c | Bitwise.c | c.json

Multiplication.c > main()

```c
#include <stdio.h>

int main(){
    int num;

    printf("Enter number to print its multiplication table : ");
    scanf("%d",&num);

    printf("Multiplication table of %d \n",num);

    for(int i = 1; i<=10; i++){
        printf("%d * %d = %d \n",num,i,num * i);
    }
    return 0;
}
```

Output :

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
Multiplication table of 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Ln 12, Col 49 Spaces: 4 U

Algorithm :

Step1-Start

Step2-input a number

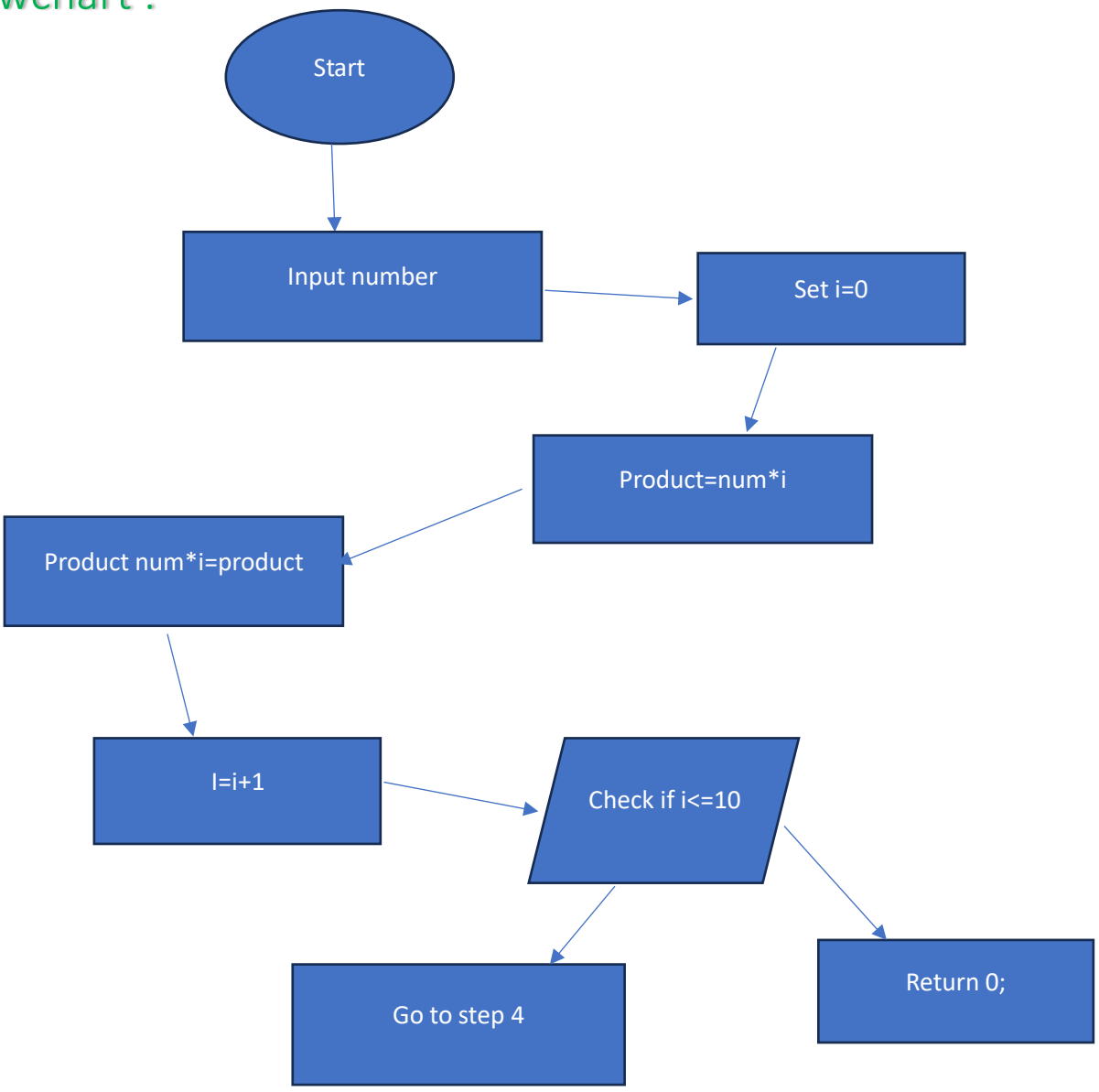
Step3-initialize a loop from 1 to 10

Step4-Multiply the number: For each iteration of the loop (from 1 to 10), calculate the product of the number (num) and the current iteration index (i).

Step5-Display the result in the format: num * i = product.

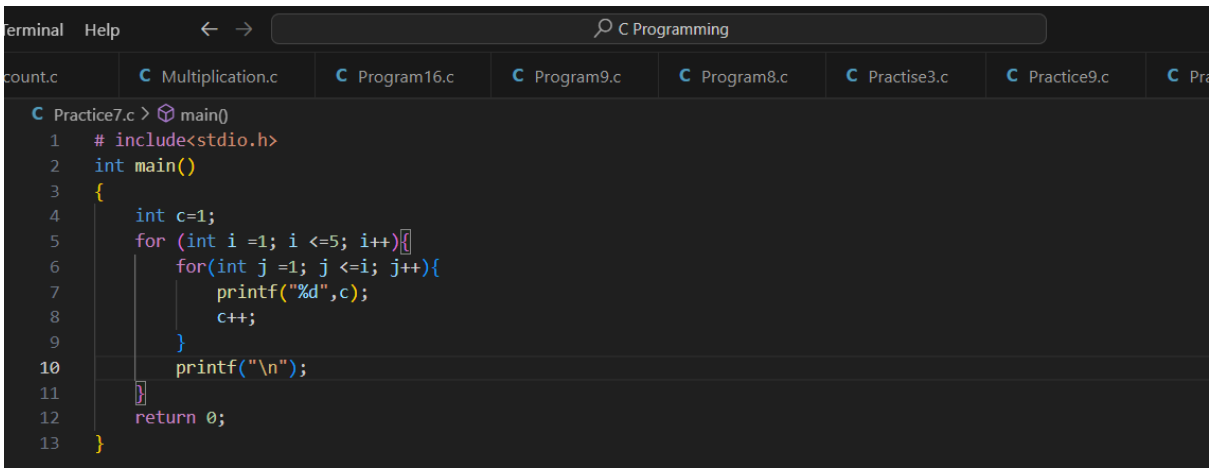
Step 6- End

Flowchart :

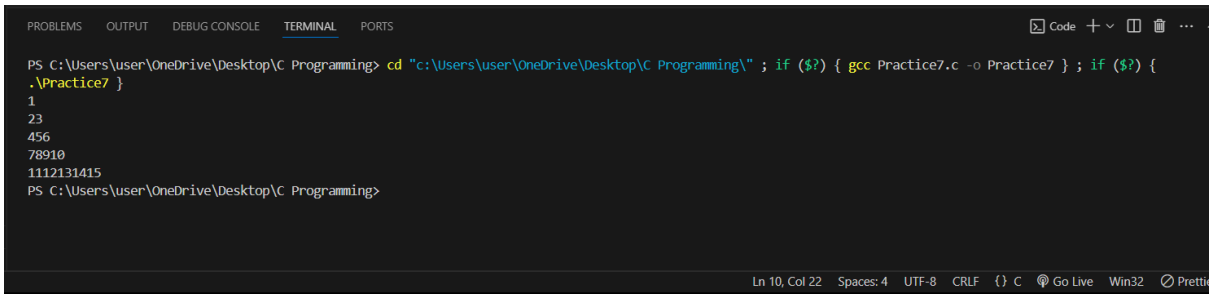


2. WAP to generate the following patterns (we can done 7 of them in lectures, finish all):

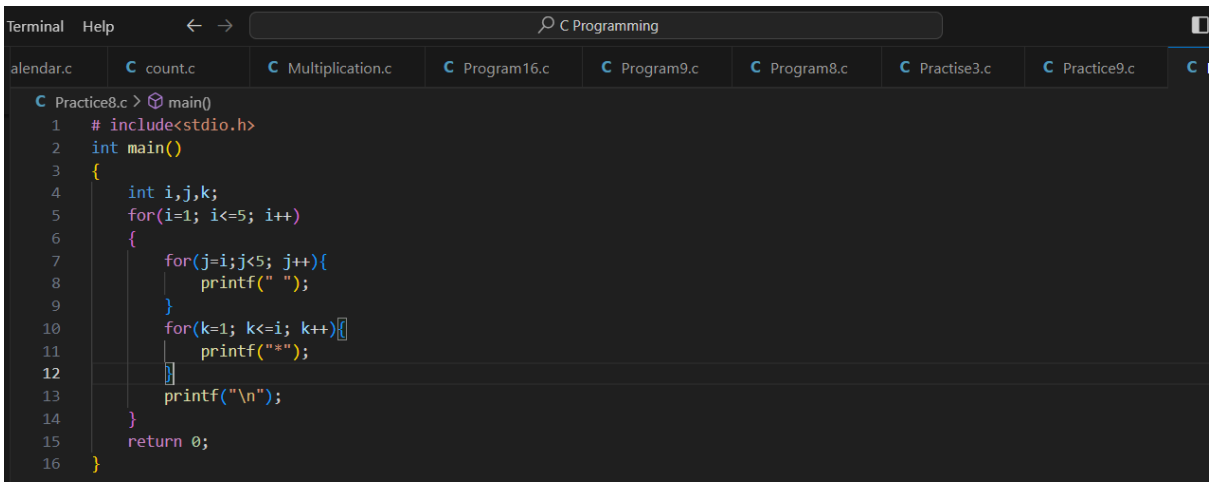
1.



Output :



2.



Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc Practice8.c -o Practice8
.\Practice8 }
    *
   **
  ***
 ****
*****
PS C:\Users\user\OneDrive\Desktop\C Programming>

Ln 12, Col 10  Spaces: 4  UTF-8  CRLF  {} C
```

3.

Terminal Help C Programming

Name.c | Collinear.c | calendar.c | count.c | Multiplication.c | Program16.c | Program9.c

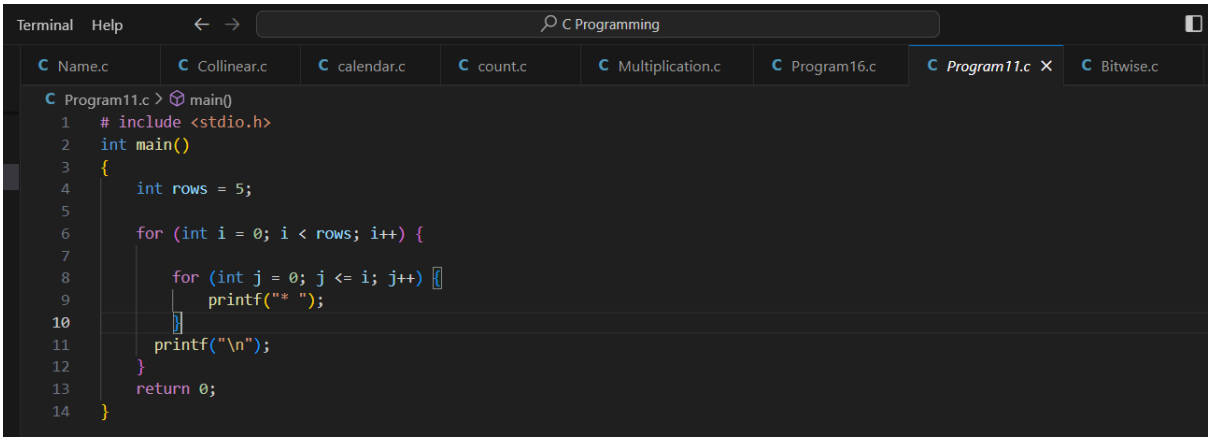
Practise3.c > main()

```c
# include<stdio.h>
int main()
{
    int i,j,rows;
    printf("Enter number of rows : ");
    scanf("%d",&rows);

    for(i=rows; i>=1; i--){
        for(j=1; j<=i; j++){
            printf("*");
        }
        printf("\n");
    }

}
```

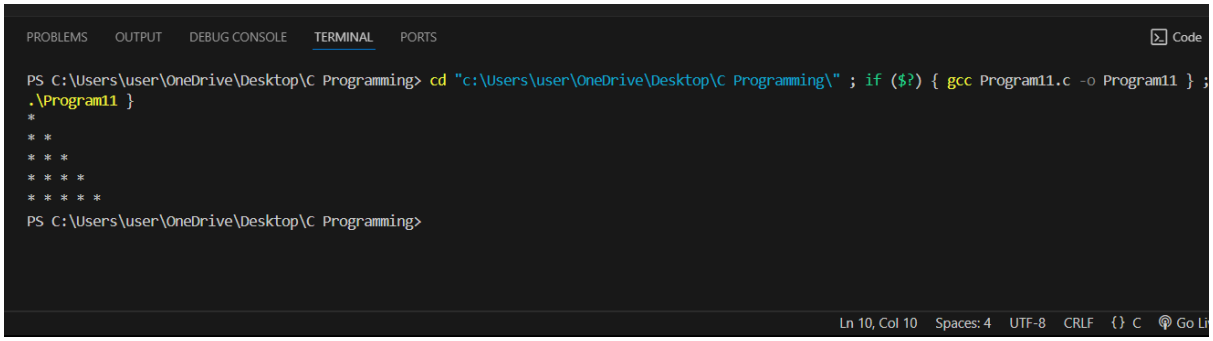
Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc Practise3.c -o Practise3
.\Practise3 }
Enter number of rows : 5
*****
****
***
**
*
PS C:\Users\user\OneDrive\Desktop\C Programming>

Ln 9, Col 29  Spaces: 4  UTF-8  CRLF  {} C
```

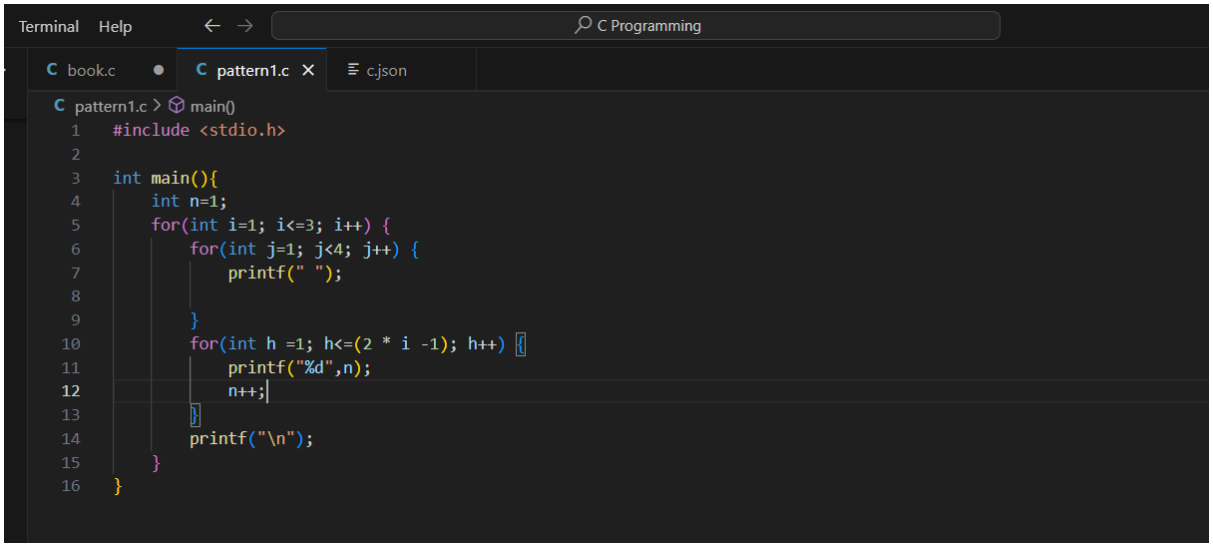
4.


Terminal Help
C Programming
Name.c
Collinear.c
calendar.c
count.c
Multiplication.c
Program16.c
Program11.c
Bitwise.c
Program11.c > main()
include <stdio.h>
int main()
{
int rows = 5;
for (int i = 0; i < rows; i++) {
for (int j = 0; j <= i; j++) {
printf("* ");
}
printf("\n");
}
return 0;
}

Output :


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Code
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc Program11.c -o Program11 } ;
.\Program11 }
*
* *
* * *
* * * *
* * * * *
PS C:\Users\user\OneDrive\Desktop\C Programming>
Ln 10, Col 10
Spaces: 4
UTF-8
CRLF
{} C

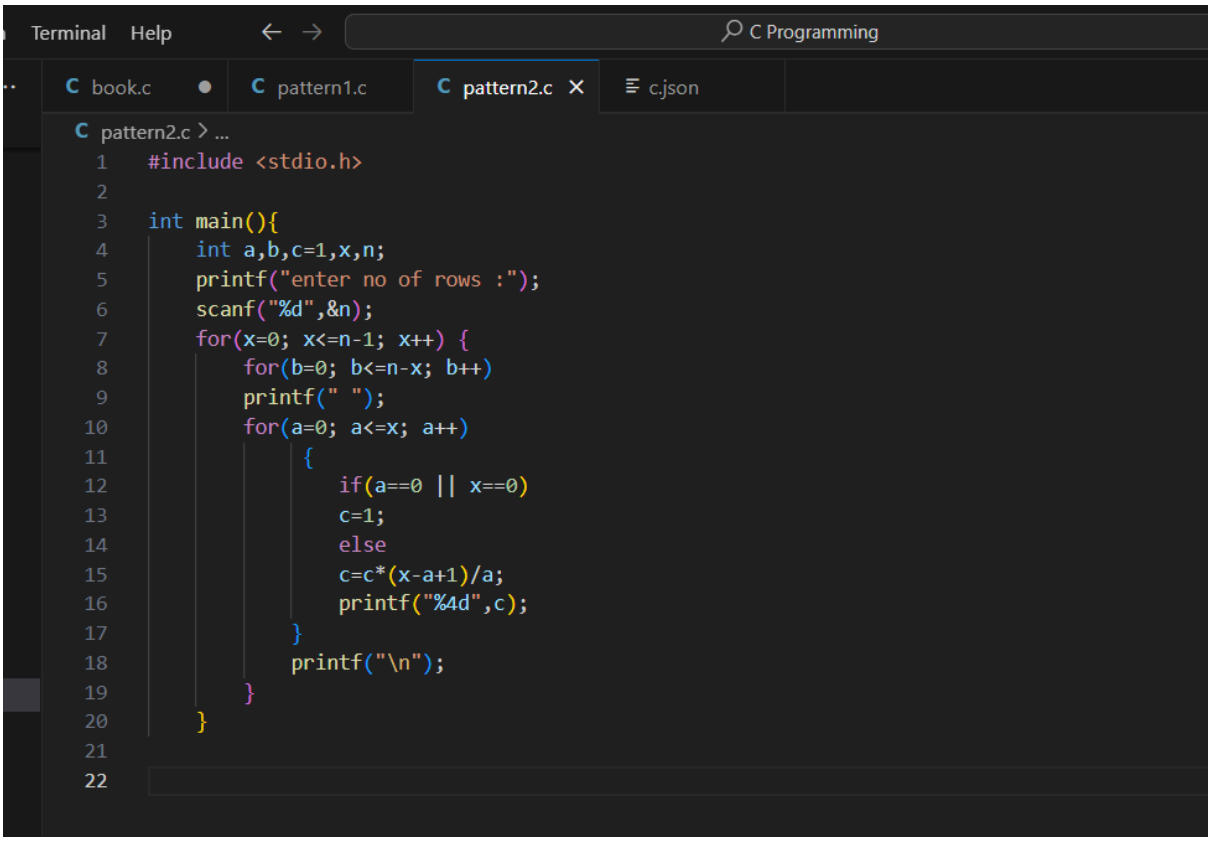
5.


Terminal Help
C Programming
book.c
pattern1.c
c.json
pattern1.c > main()
#include <stdio.h>
int main(){
int n=1;
for(int i=1; i<=3; i++) {
for(int j=1; j<4; j++) {
printf(" ");
}
for(int h =1; h<=(2 * i -1); h++) {
printf("%d",n);
n++;
}
printf("\n");
}
}

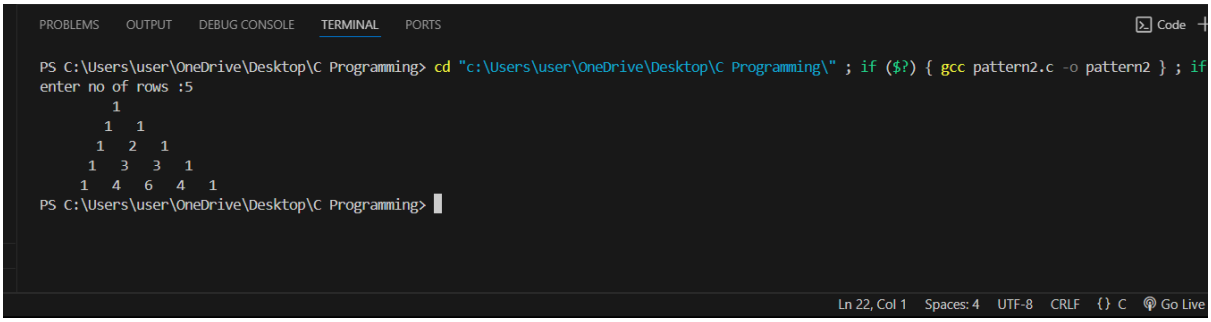
Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc pattern1.c -o pattern1
   1
   234
   56789
PS C:\Users\user\OneDrive\Desktop\C Programming>

Ln 12, Col 17  Spaces: 4  UTF-8  CRLF  {} C
```

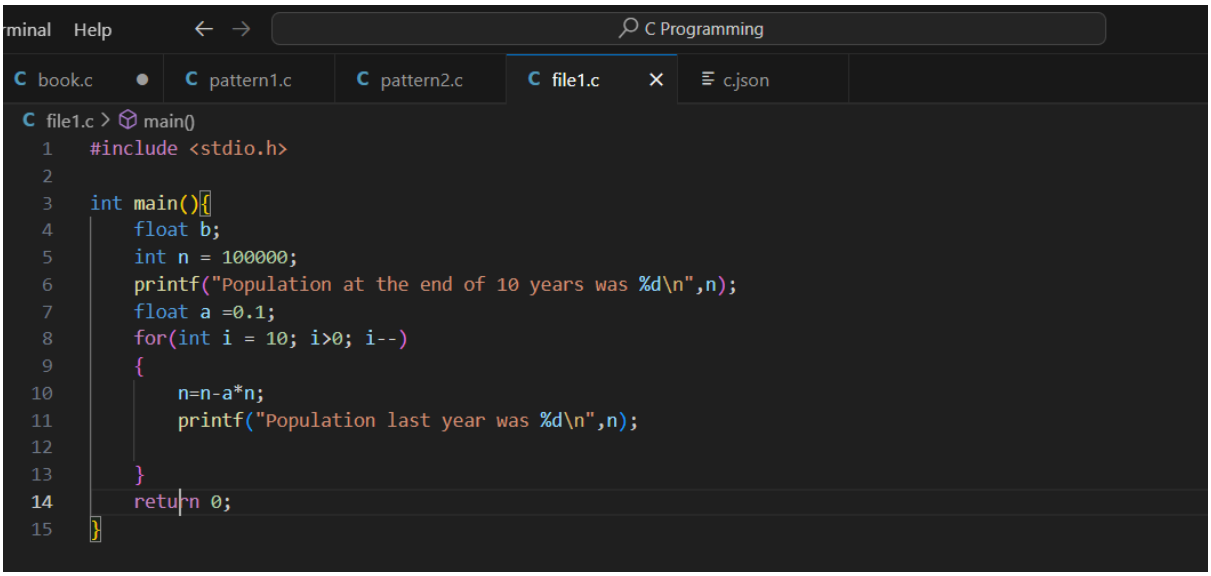
6.



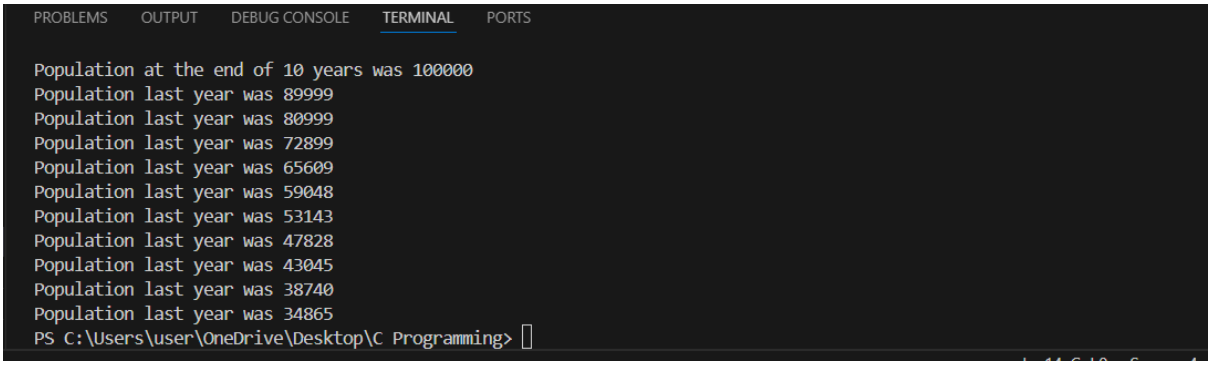
Output :



Q15- The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in the last decade.



Output :



Algorithm :

Step1-Start

Step 2- Initialize the population n to 100,000.

Initialize the decline rate a to 0.1 (10%).

Step 3- Print the initial population n using printf

Step 4- Loop 10 times, decrementing the loop counter i from 10 to 1

Step 5-End

Q16- Ramanujan Number is the smallest number that can be expressed as the sum of two cubes in two different ways. WAP to print all such numbers up to a reasonable limit. Example of Ramanujan number: 1729 12^3 + 1^3 and 10^3 + 9^3. for a number L=20(that is limit)

```c
#include <stdio.h>

int main(){
    int a=0;
    for(int i=0; i<35000; i++) {
        a=0;
        for(int j=1; (j*j*j)<=i; j++){
            for(int k=j; (j*j*j)+(k*k*k)<=i; k++) {
                if(j*j*j+k*k*k==i){
                a++;
                if(a==2) {
                    printf("ramanujan number is %d\n",i);
                }
            }
        }
    }
    }
    return 0;
}
```

Output :

```
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc file2.c -o fi
ramanujan number is 1729
ramanujan number is 4104
ramanujan number is 13832
ramanujan number is 20683
ramanujan number is 32832
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Algorithm :

Step1- Input a=0;

Step2-Initialize loop limit from 0 to 35000

Step 3-Generate Sums of Cubes in i j k using loop

Step 4-check ramanujan number

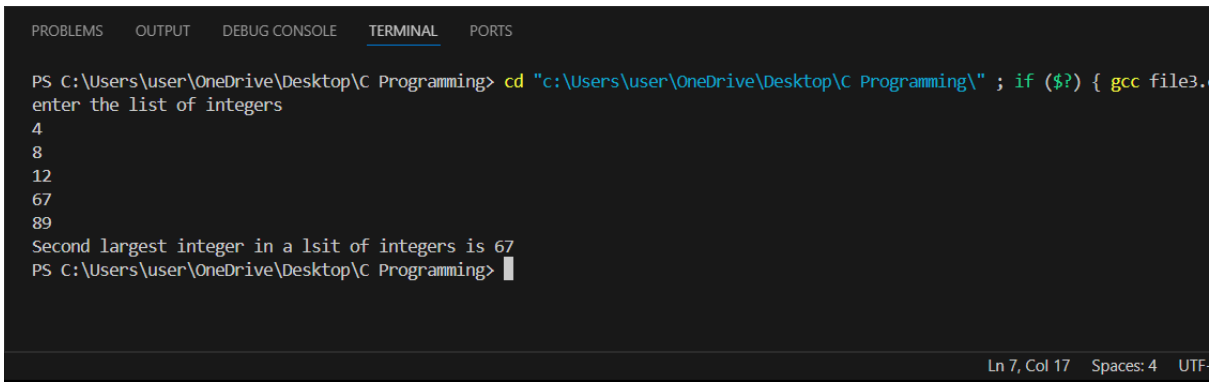
Step5-Print ramanujan number

Step6-End

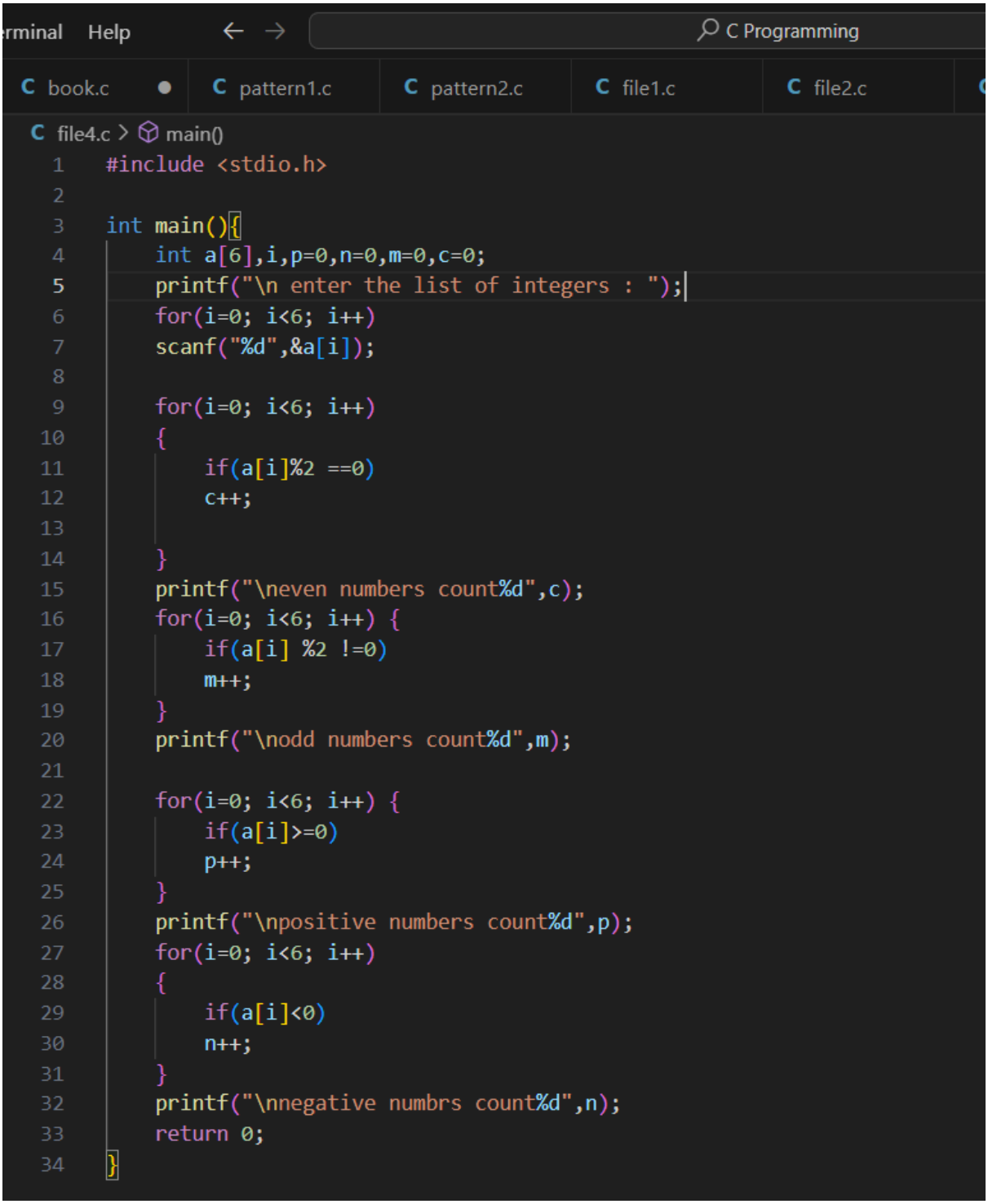
Experiment- Arrays

Q17- WAP to read a list of integers and store it in a 1D array. Write a C program to print the second largest integer in a list of integers.

Output :



Q18- WAP to read a list of integers and store it in a 1D array. Write a C program to count and diaplay positive, negative, odd, and even numbers in an array.




```c
#include <stdio.h>

int main(){
    int a[6],i,p=0,n=0,m=0,c=0;
    printf("\n enter the list of integers : ");
    for(i=0; i<6; i++)
    scanf("%d",&a[i]);

    for(i=0; i<6; i++)
    {
        if(a[i]%2 ==0)
        c++;

    }
    printf("\neven numbers count%d",c);
    for(i=0; i<6; i++) {
        if(a[i] %2 !=0)
        m++;
    }
    printf("\nodd numbers count%d",m);

    for(i=0; i<6; i++) {
        if(a[i]>=0)
        p++;
    }
    printf("\npositive numbers count%d",p);
    for(i=0; i<6; i++)
    {
        if(a[i]<0)
        n++;
    }
    printf("\nnegative numbrs count%d",n);
    return 0;
}
```

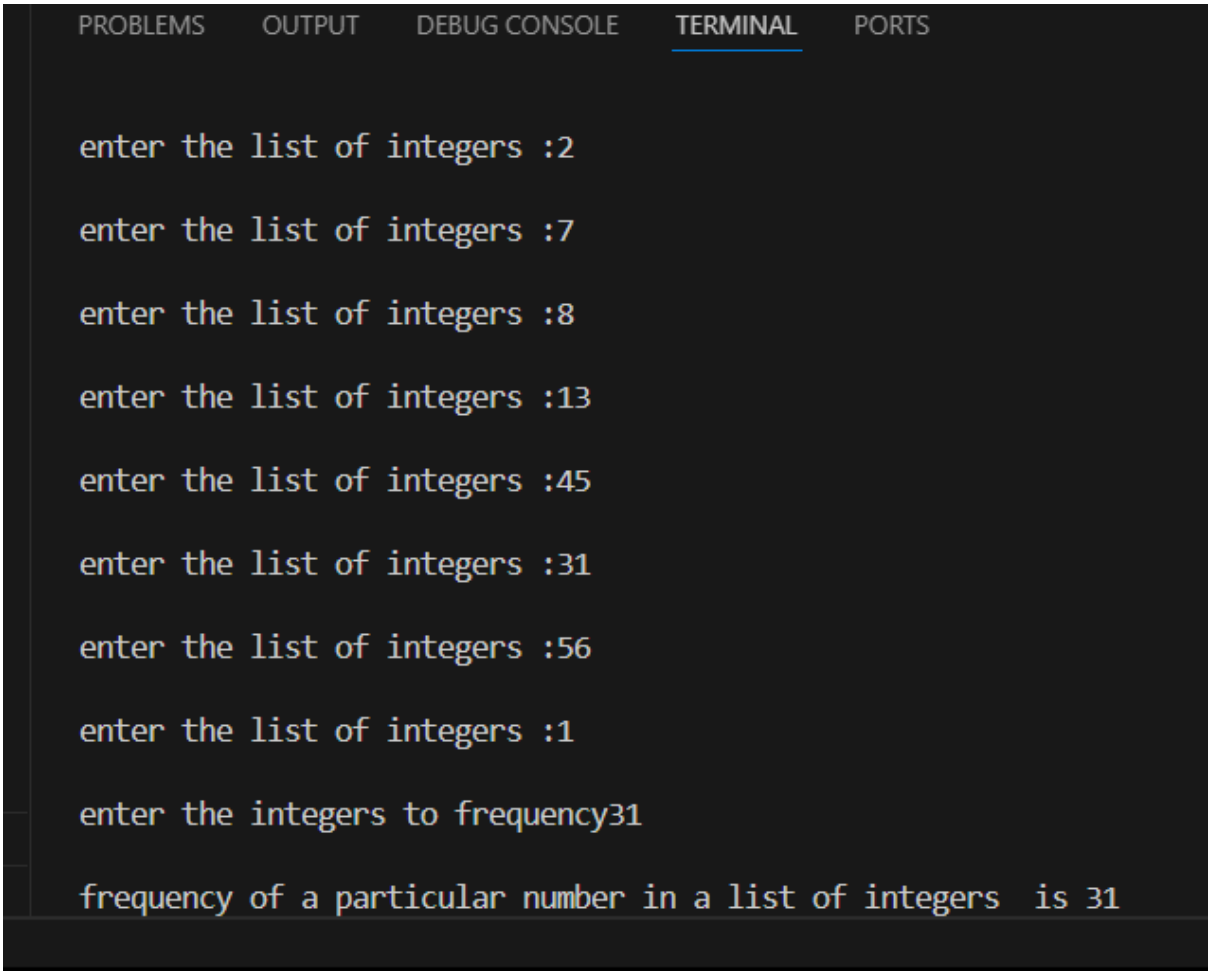
Output :

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

 enter the list of integers : 10
-42
-8
13
11
-24

even numbers count4
odd numbers count2
positive numbers count3
negative numbrs count3
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

Q19- WAP to read a list of integers and store it in a 1D array. Write a C program to find the frequency of a particular number in a list of integers.

```
erminal  Help        C Programming

C book.c ●  C pattern1.c  C pattern2.c  C file1.c  C file2.c  C file3.c  C file4.c

C file5.c > main()
 1  #include <stdio.h>
 2
 3  int main(){
 4      int a[10],i,freq=0,key;
 5      for(i=0; i<10; i++)
 6      {
 7          printf("\nenter the list of integers :");
 8          scanf("%d",&a[i]);
 9      }
10      printf("\nenter the integers to frequency");
11      scanf("%d",&key);
12      for(i=0; i<10; i++)
13      if(a[i]==key)
14      freq++;
15      printf("\nfrequency of a particular number in a list of integers  is %d",key,freq);
16
17      return 0;
18  }
```

Output :



Q20- Develop a C function ISPRIME(num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.

rminal Help C Programming

pattern2.c | file1.c | file2.c | file3.c | file4.c | fil

file7.c > main()

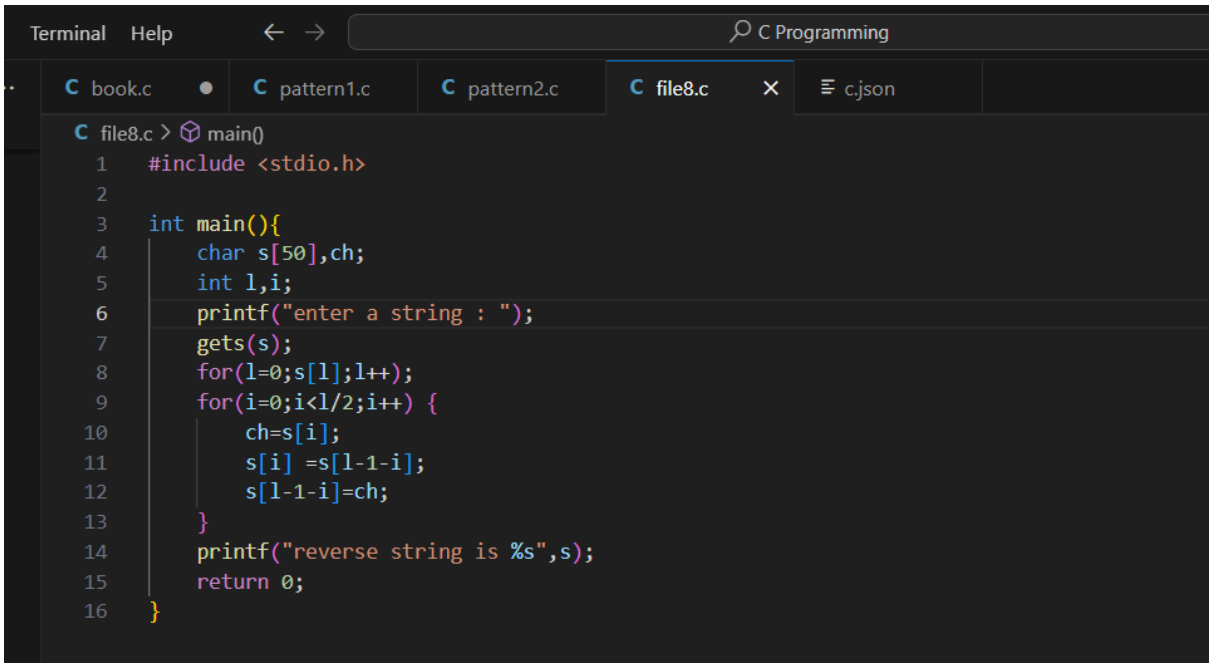
```c
#include <stdio.h>
int isprime(int num) {
    if(num<=1)return 0;
    for(int i=2; i*i<=num; i++) {
        if(num%i==0) return 0;
    }
    return 1;
}
int main(){
    int a,b;
    printf("enter range : ");
    scanf("%d%d",&a,&b);
    printf("Prime numbers between %d and %d are:\n",a,b);
    for(int i=a; i<=b; i++) {
        if(isprime(i)) {
            printf("%d\n",i);
        }
    }
    printf("\n");
    return 0;
}
```

Output :

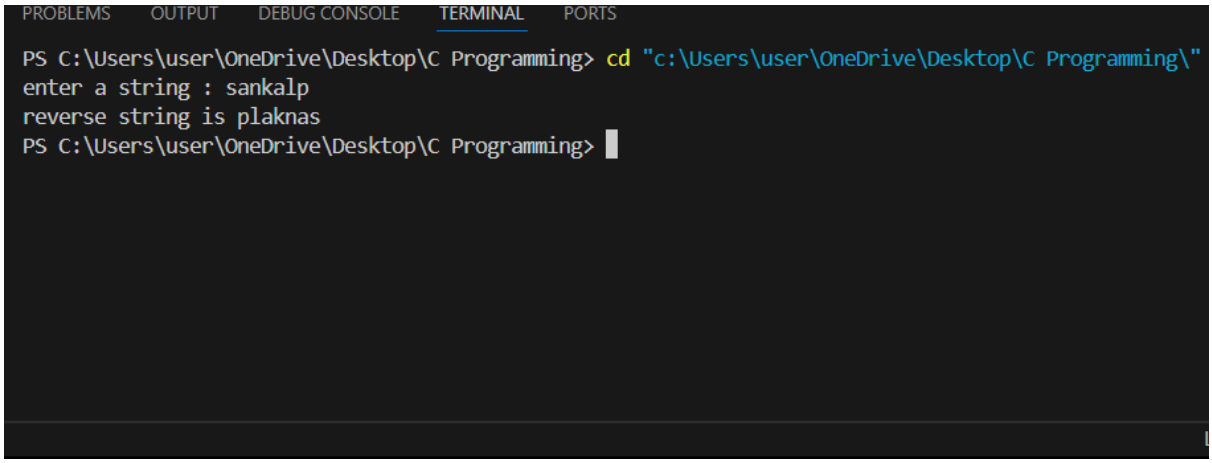
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
enter range : 0
20
Prime numbers between 0 and 20 are:
2
3
5
7
11
13
17
19

PS C:\Users\user\OneDrive\Desktop\C Programming>
```

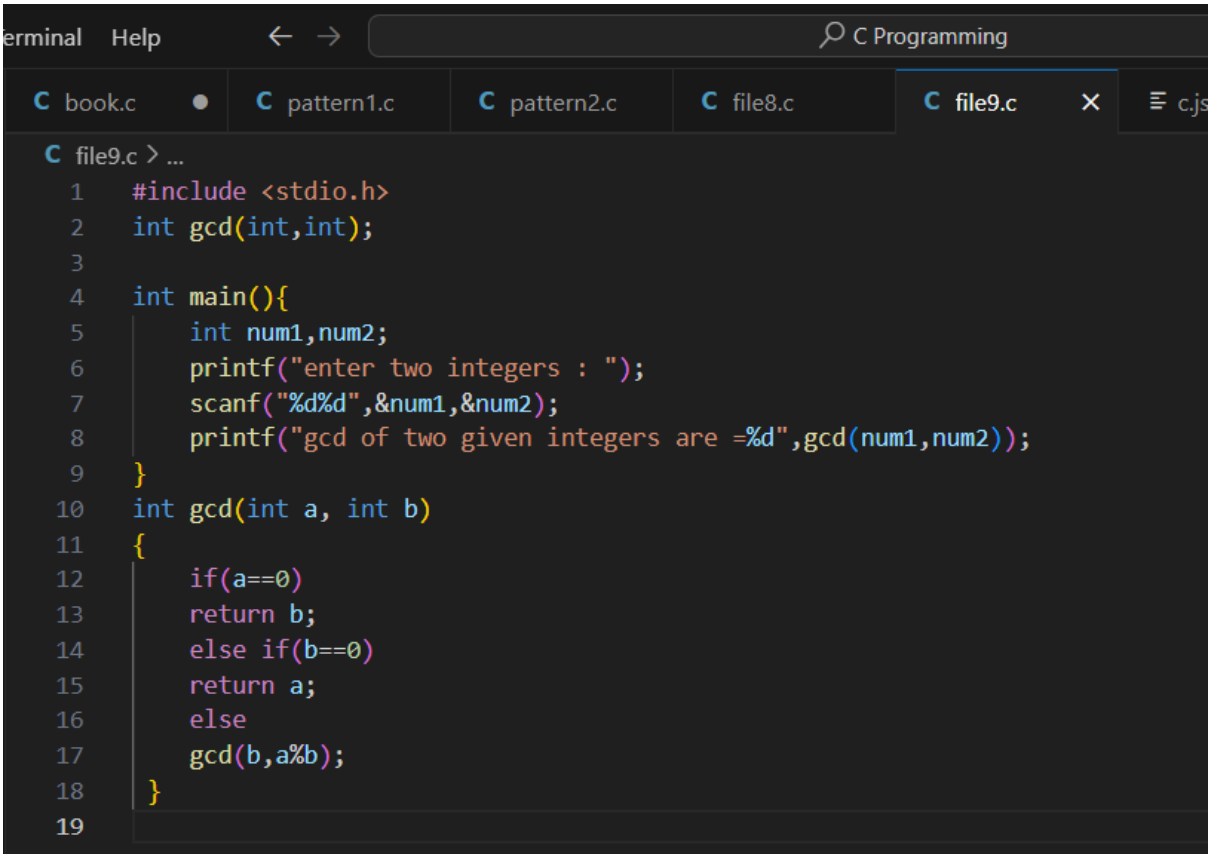
Q21-Develop a function REVERSE(str) that accepts a string argument. Write a C program that invokes this function to find the reverse of a given string.



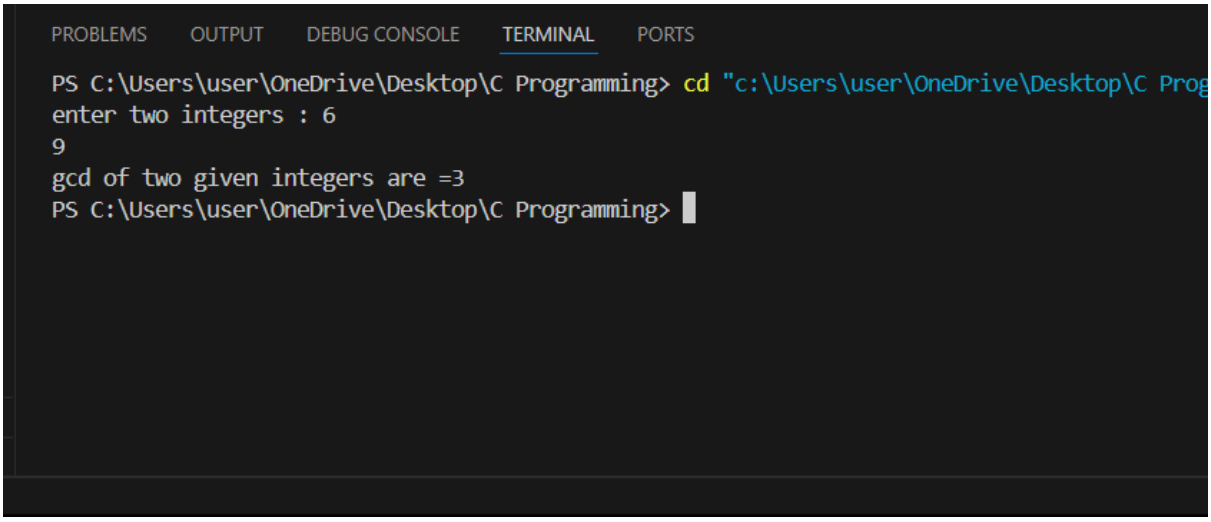
Output :



Q23-Develop a recursive function GCD(num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.



Output :



Q25-Develop a recursive function FIBO(num) that accepts an integer
argument. Write a C program that invokes this function to generate
the Fibonacci sequence up to num.

Terminal Help C Programming

book.c pattern1.c pattern2.c file8.c file9.c

file10.c > ...

```c
#include <stdio.h>
int fibo(int num);

int main(){
    int num, i;
    printf("enter an integer : ");
    scanf("%d",&num);
    printf("fibonacci sequence up to %d is \n",num);
    for(i=0; i<num; i++)
    printf("%d",fibo(i));
}
  int fibo(int num)
  {
    if(num==0)
    return 0;
    else if(num==1)
    return 1;
    else
    return fibo(num-1)+fibo(num-2);
  }
```

Output :

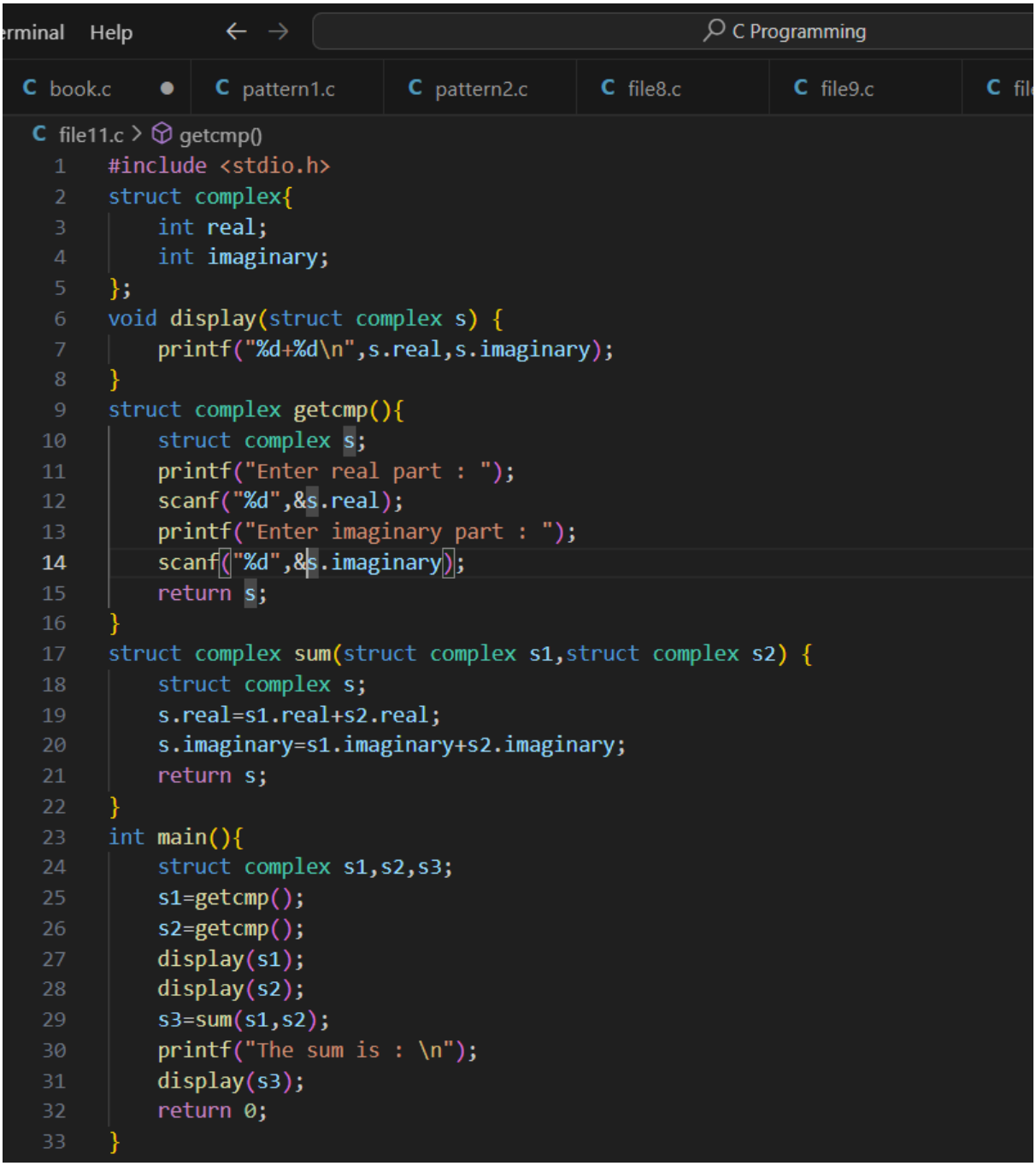
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programming\" ; if ($?) { gcc fi
enter an integer : 6
fibonacci sequence up to 6 is
011235
PS C:\Users\user\OneDrive\Desktop\C Programming>
```

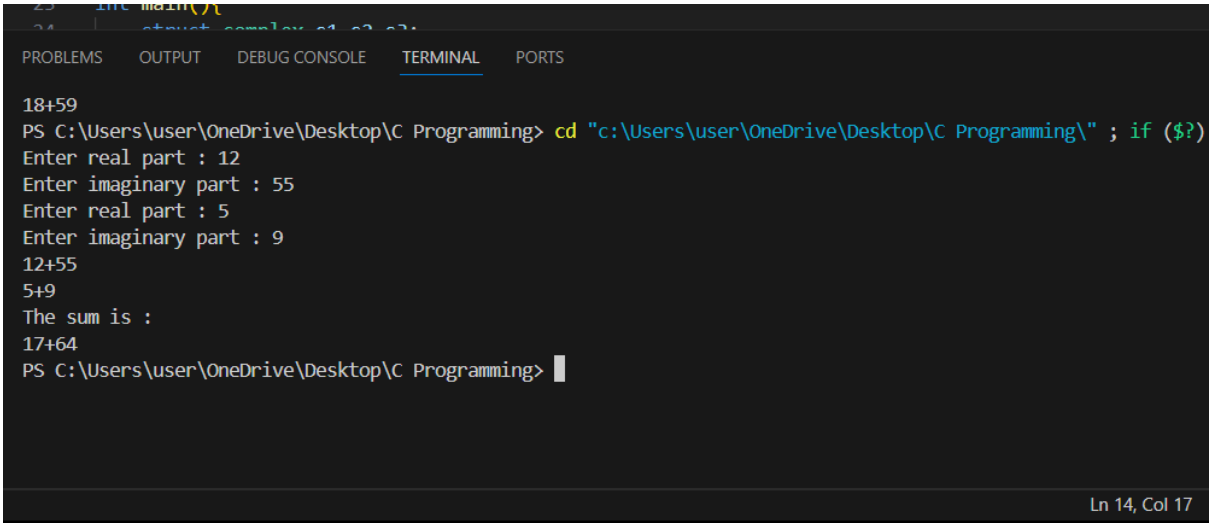
Ln 20, Col 5 Spaces: 4

Experiment- Structures

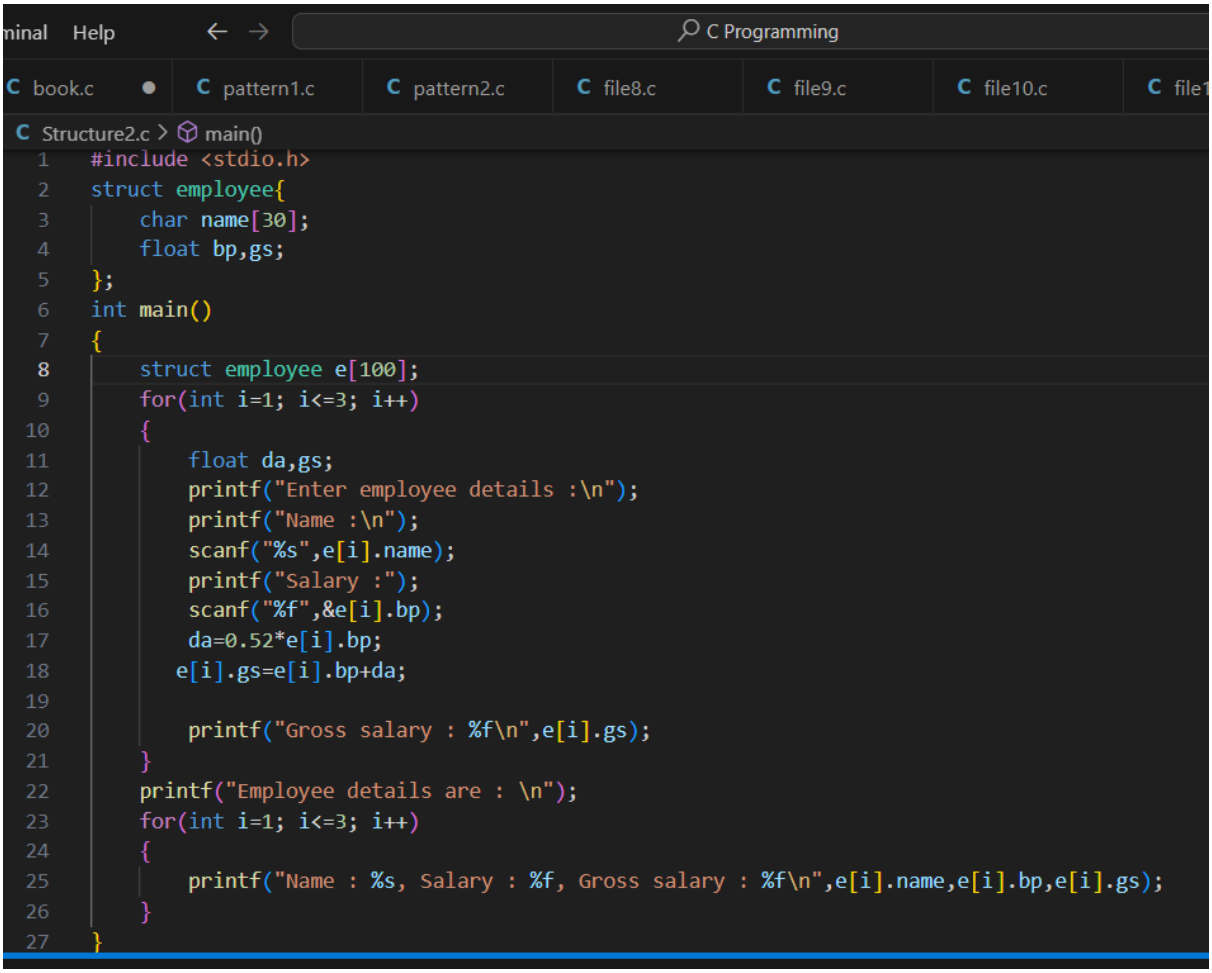
Q26- 1. Write a C program that uses functions to perform the following operations: a. Reading a complex number. b. Writing a complex number. c. Addition and subtraction of two complex numbers



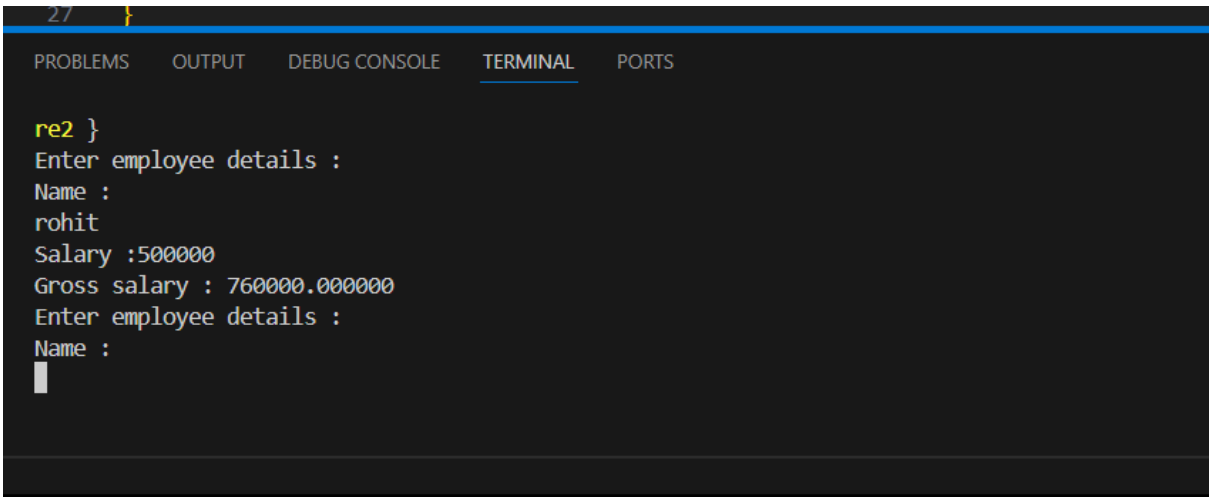
Output :



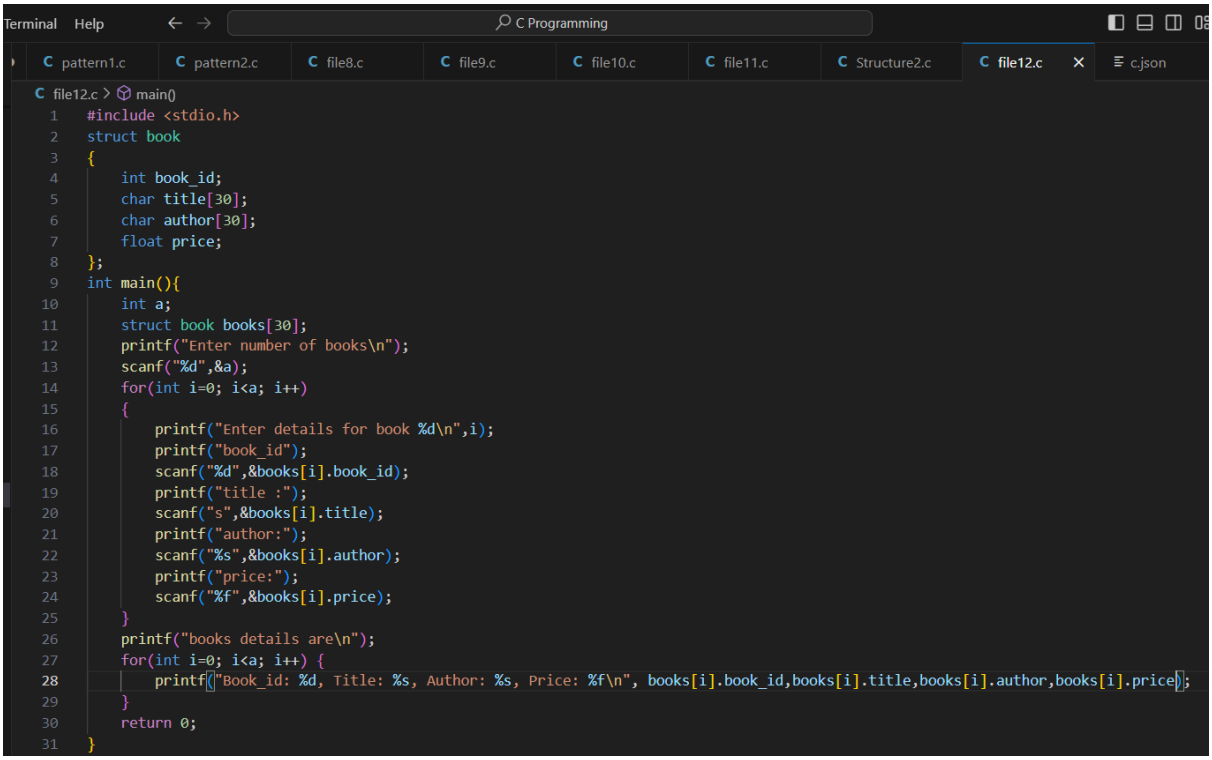
Q27- Write a C program to compute the monthly pay of 100 employees using each employee_s name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.


minal Help
C Programming
book.c
pattern1.c
pattern2.c
file8.c
file9.c
file10.c
Structure2.c > main()
#include <stdio.h>
struct employee{
char name[30];
float bp,gs;
};
int main()
{
struct employee e[100];
for(int i=1; i<=3; i++)
{
float da,gs;
printf("Enter employee details :\n");
printf("Name :\n");
scanf("%s",e[i].name);
printf("Salary :");
scanf("%f",&e[i].bp);
da=0.52*e[i].bp;
e[i].gs=e[i].bp+da;
printf("Gross salary : %f\n",e[i].gs);
}
printf("Employee details are : \n");
for(int i=1; i<=3; i++)
{
printf("Name : %s, Salary : %f, Gross salary : %f\n",e[i].name,e[i].bp,e[i].gs);
}
}

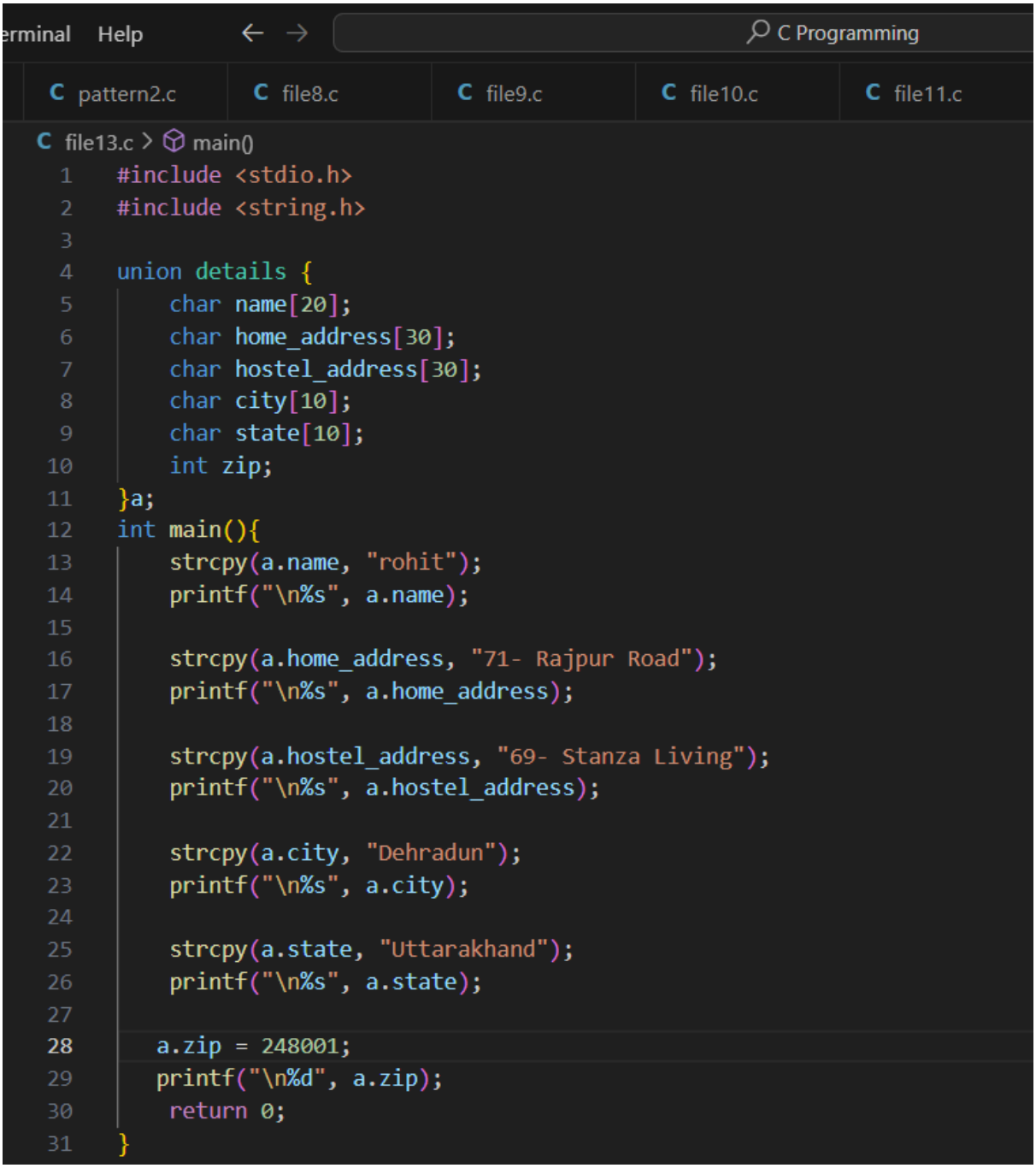
Output :


}
PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
re2 }
Enter employee details :
Name :
rohit
Salary :500000
Gross salary : 760000.000000
Enter employee details :
Name :

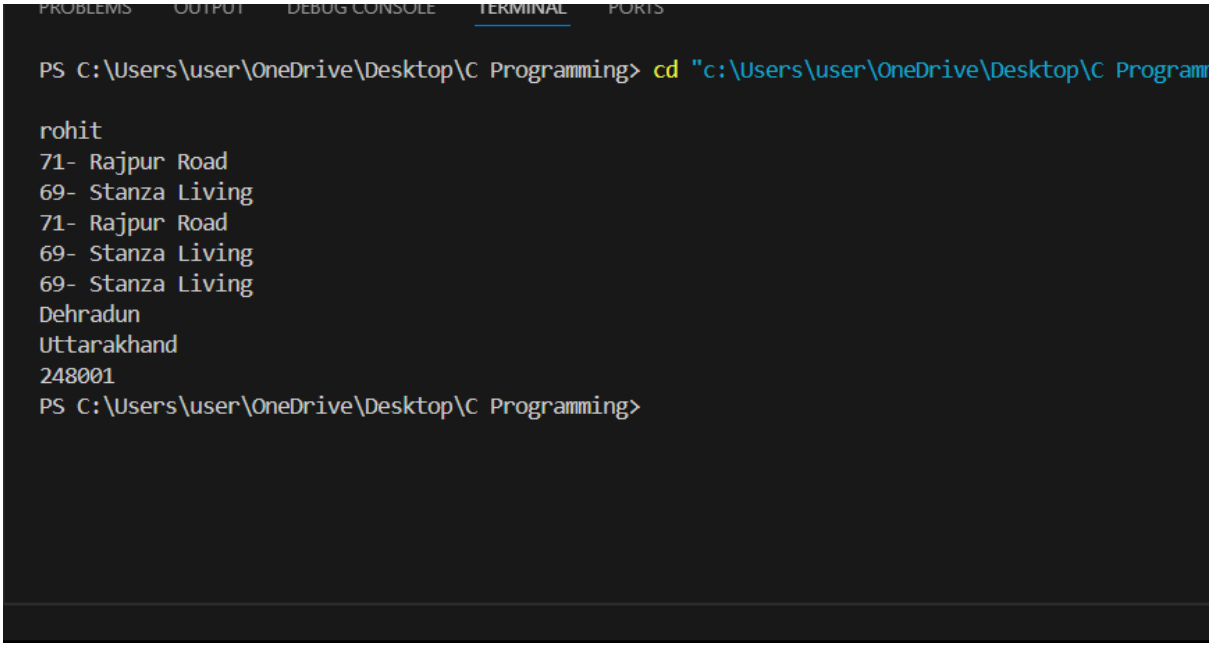
Q28-Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details



Q29- Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

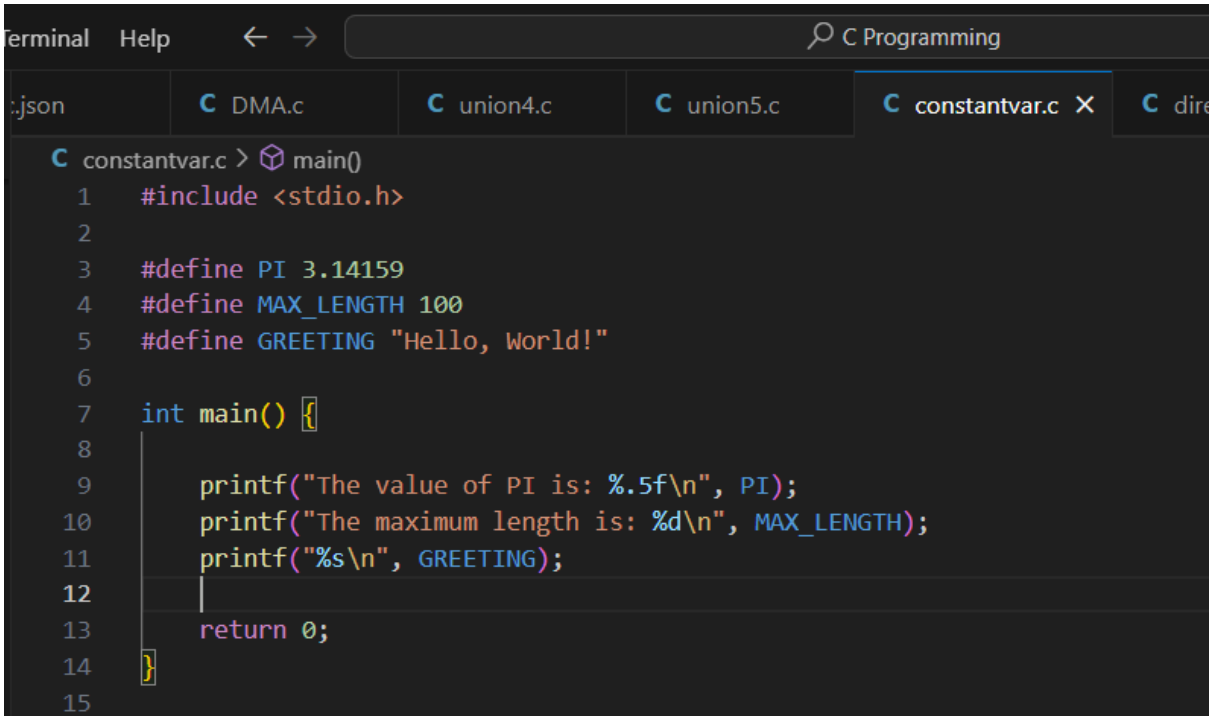

erminal Help
C Programming
pattern2.c
file8.c
file9.c
file10.c
file11.c
file13.c > main()
#include <stdio.h>
#include <string.h>
union details {
char name[20];
char home_address[30];
char hostel_address[30];
char city[10];
char state[10];
int zip;
}a;
int main(){
strcpy(a.name, "rohit");
printf("\n%s", a.name);
strcpy(a.home_address, "71- Rajpur Road");
printf("\n%s", a.home_address);
strcpy(a.hostel_address, "69- Stanza Living");
printf("\n%s", a.hostel_address);
strcpy(a.city, "Dehradun");
printf("\n%s", a.city);
strcpy(a.state, "Uttarakhand");
printf("\n%s", a.state);
a.zip = 248001;
printf("\n%d", a.zip);
return 0;
}

Output :

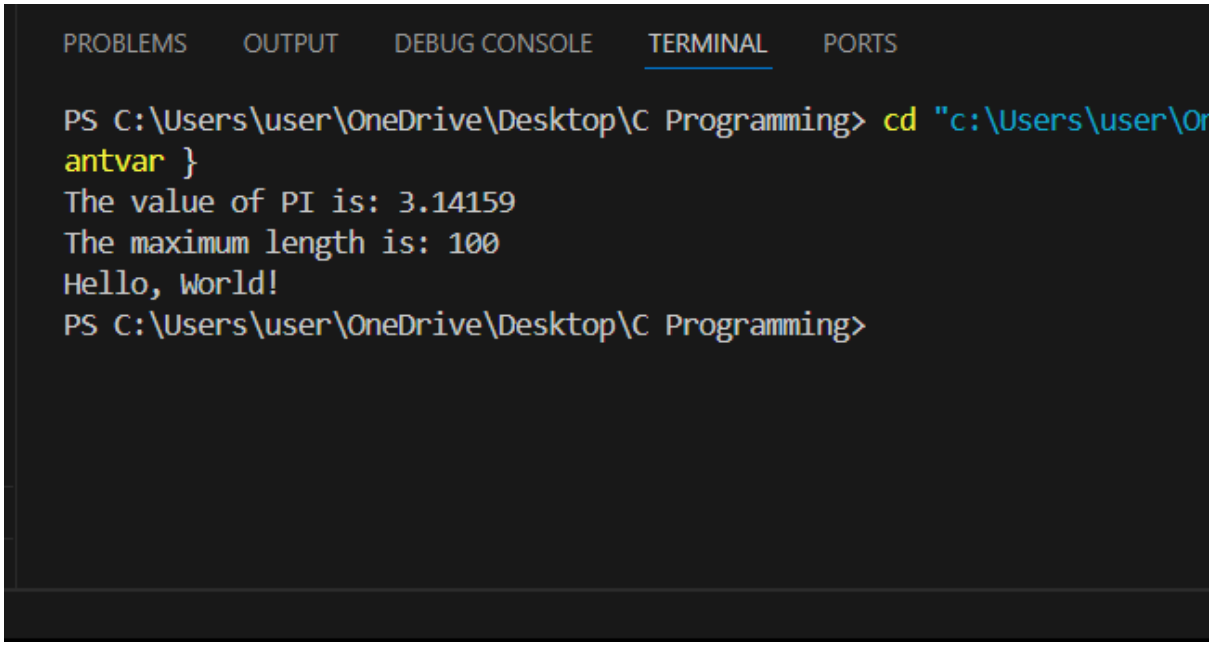

PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Programm
rohit
71- Rajpur Road
69- Stanza Living
71- Rajpur Road
69- Stanza Living
69- Stanza Living
Dehradun
Uttarakhand
248001
PS C:\Users\user\OneDrive\Desktop\C Programming>

30) Write a program to define some constant variable in preprocessor.

Input:

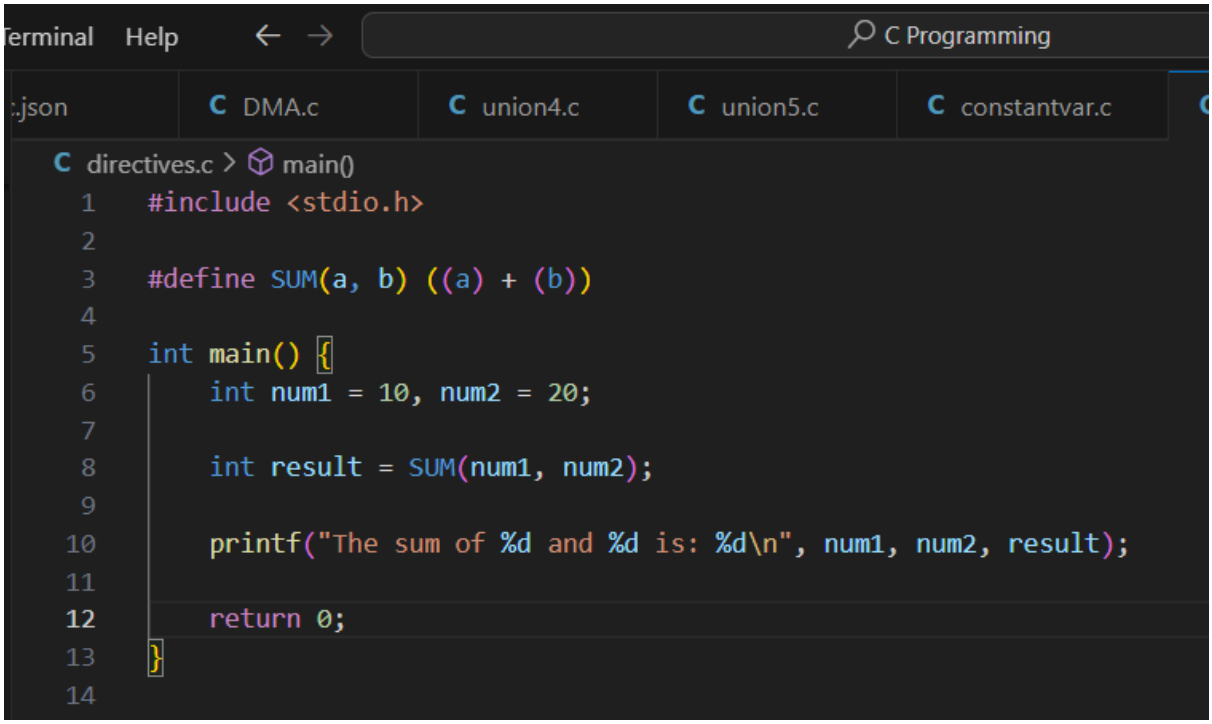


Output :

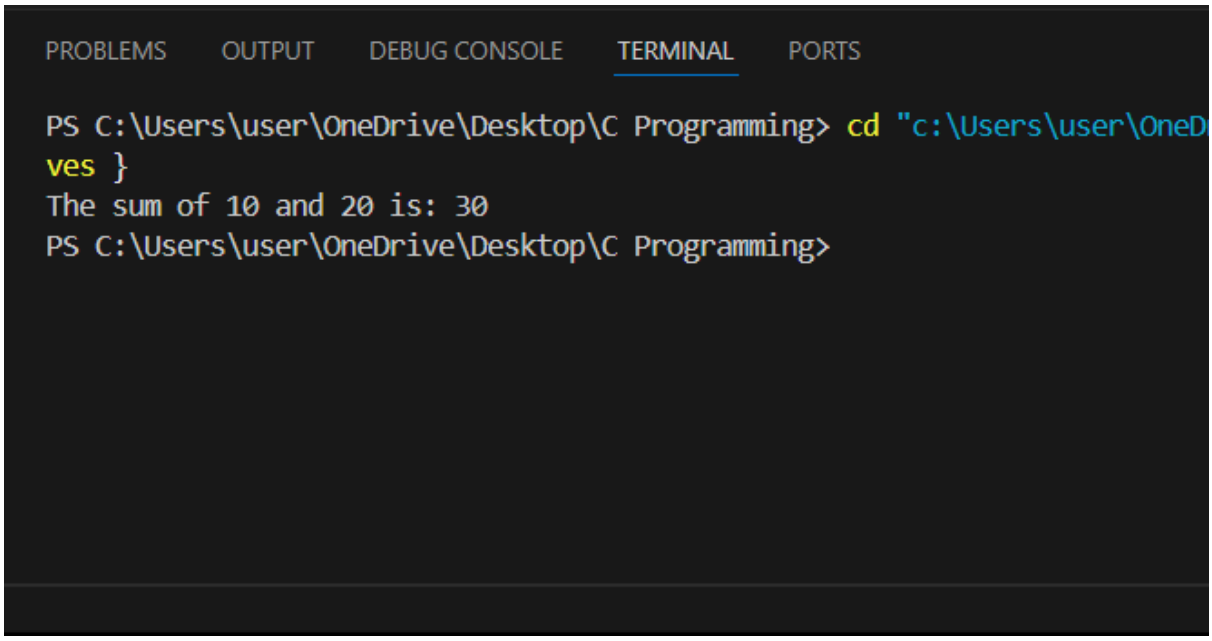


32) Write a program to define a function in directives.

Input :

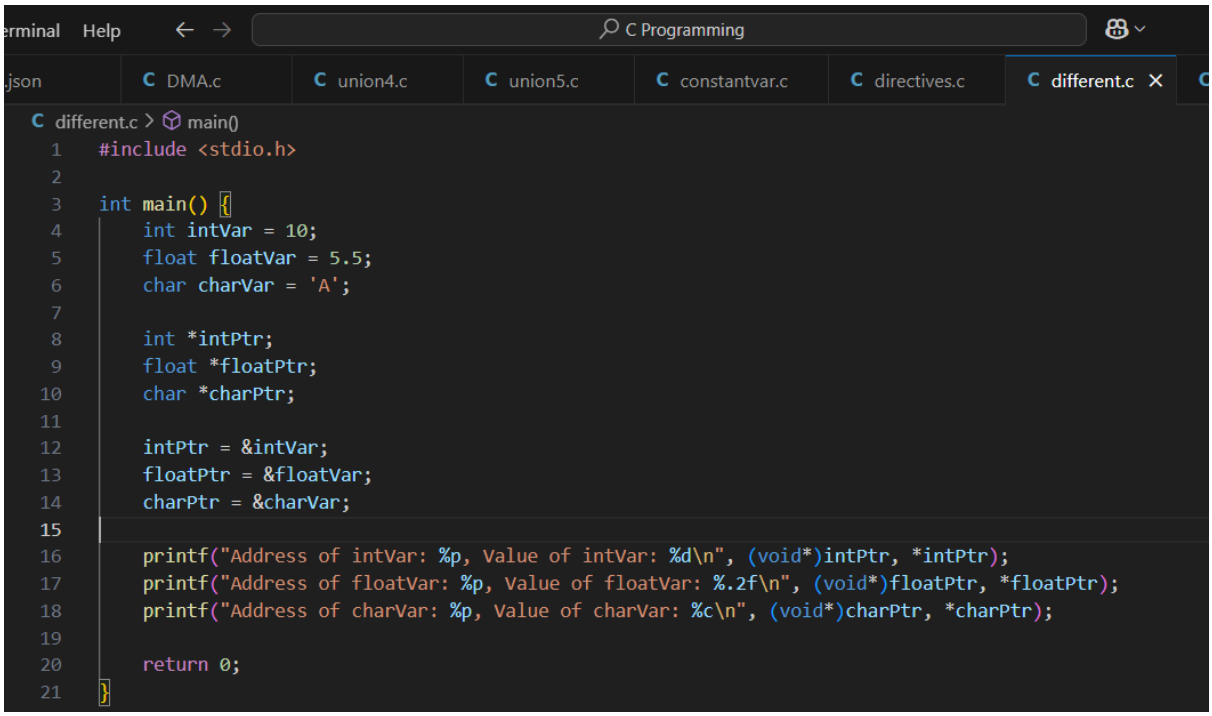


Output :

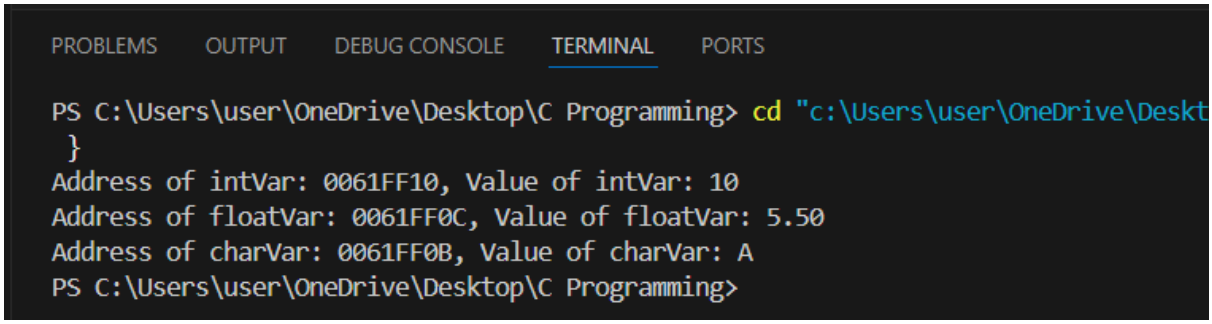


33) Declare different types of pointers (int, float, char) and initialize them with the addresses of variables. Print the values of both the pointers and the variables they point to

Input :



Output :

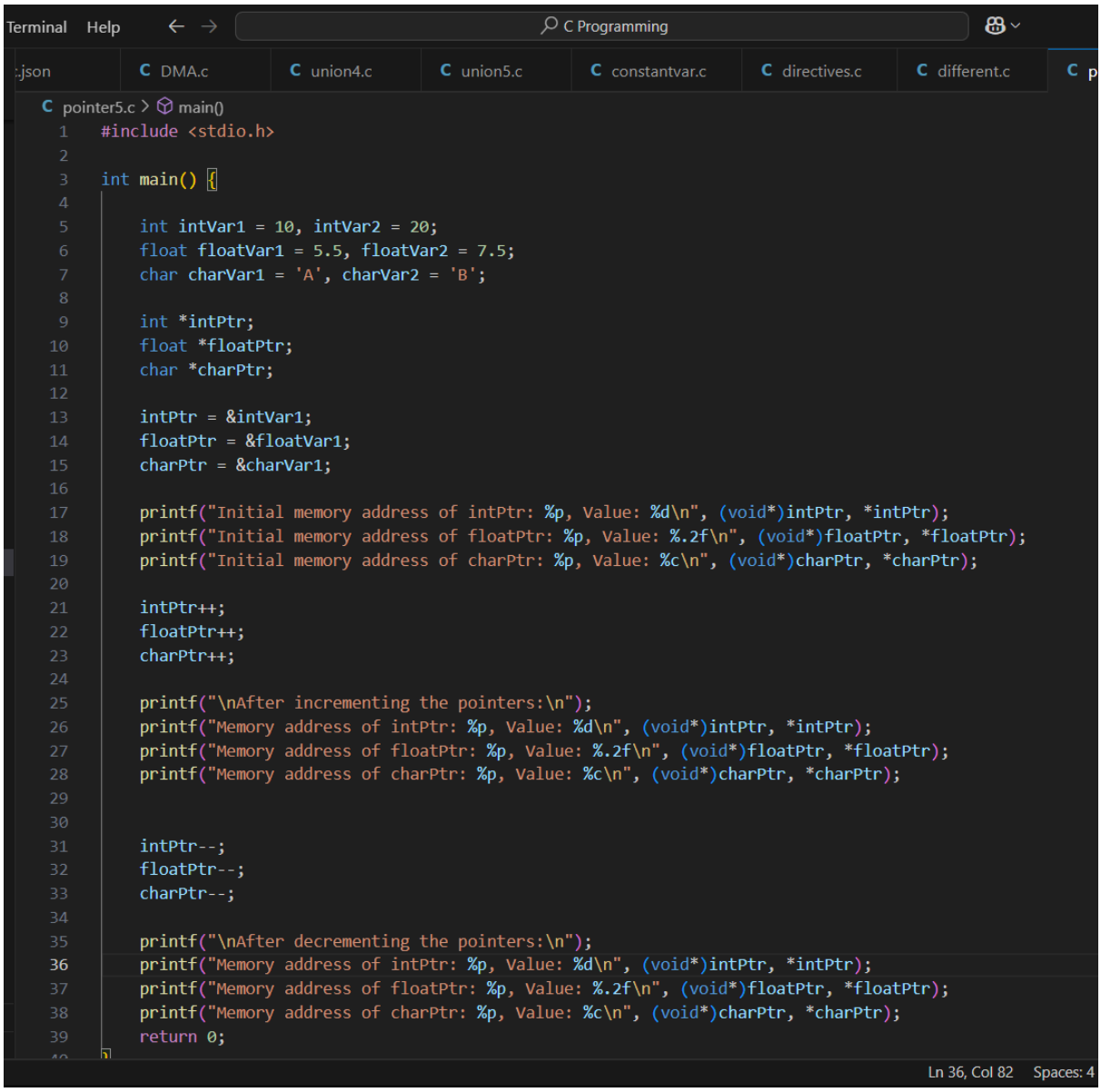


34) Perform pointer arithmetic (increment and decrement) on pointers of different

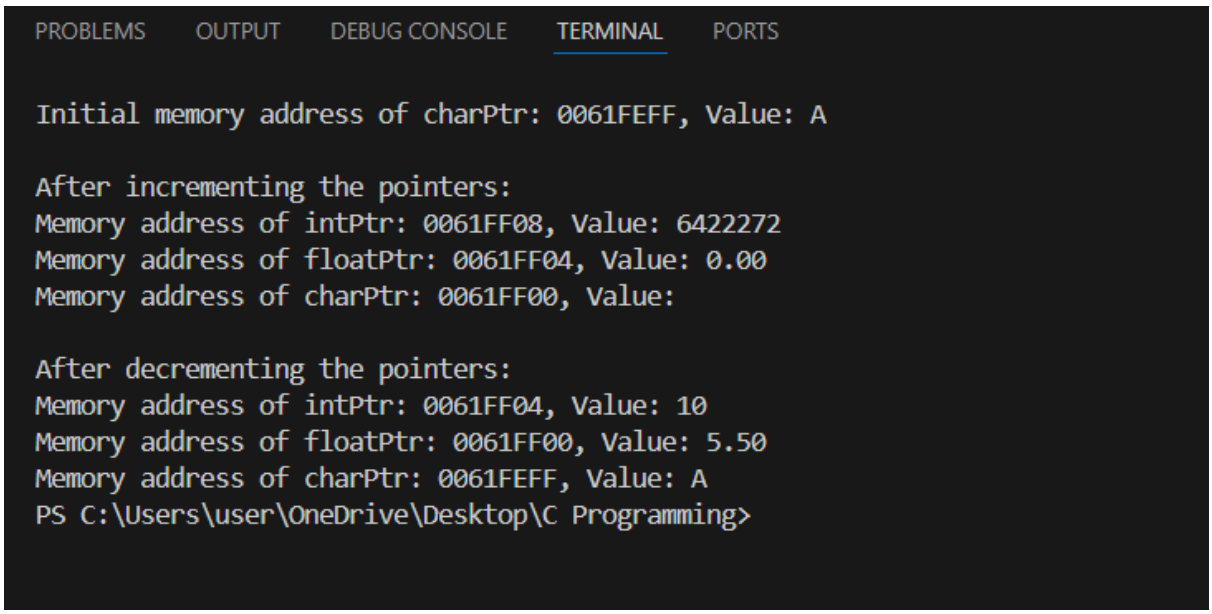
data types. Observe how the memory addresses change and the effects on

data access.

Input :



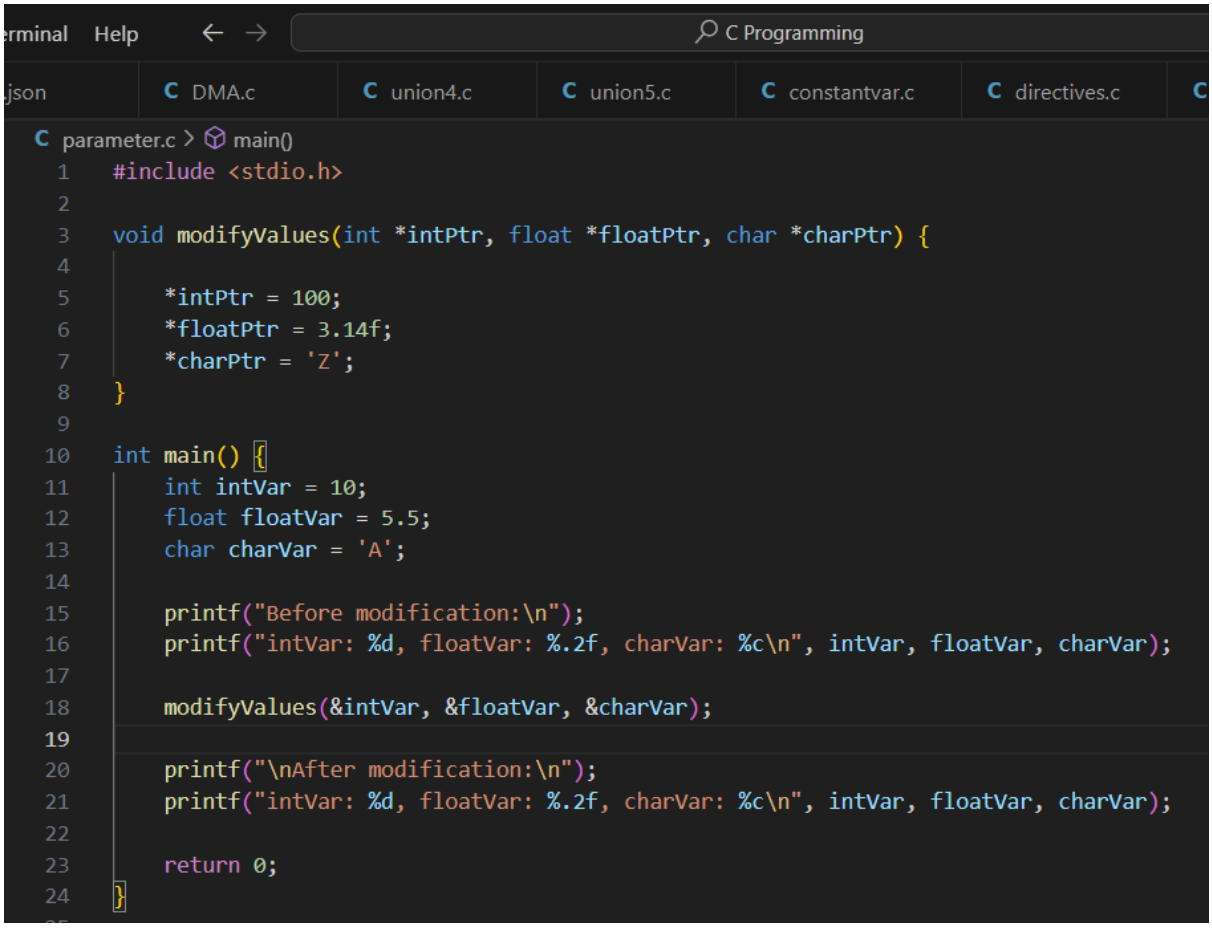
Output :



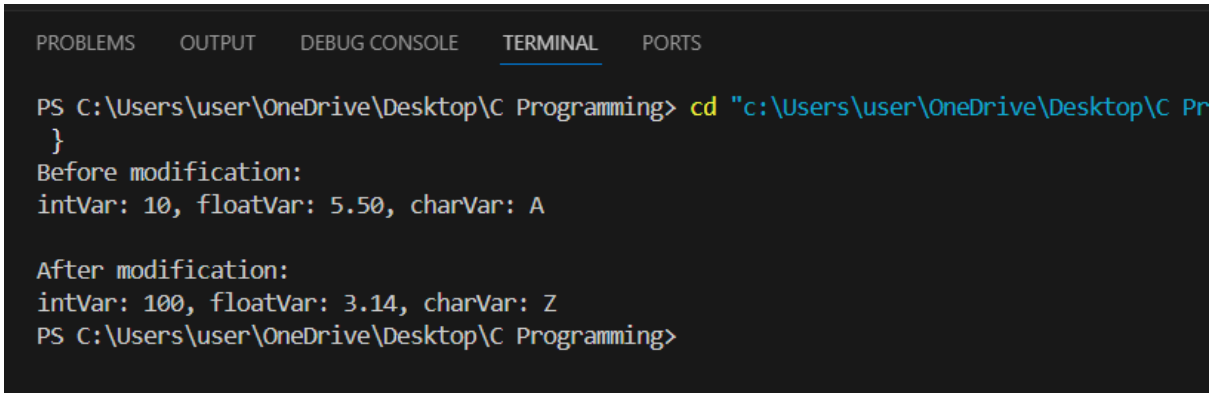
35) Write a function that accepts pointers as parameters. Pass variables by

reference using pointers and modify their values within the function.

Input :


rminal Help
C Programming
json
C DMA.c
C union4.c
C union5.c
C constantvar.c
C directives.c
C parameter.c > main()
#include <stdio.h>
void modifyValues(int *intPtr, float *floatPtr, char *charPtr) {
*intPtr = 100;
*floatPtr = 3.14f;
*charPtr = 'Z';
}
int main() {
int intVar = 10;
float floatVar = 5.5;
char charVar = 'A';
printf("Before modification:\n");
printf("intVar: %d, floatVar: %.2f, charVar: %c\n", intVar, floatVar, charVar);
modifyValues(&intVar, &floatVar, &charVar);
printf("\nAfter modification:\n");
printf("intVar: %d, floatVar: %.2f, charVar: %c\n", intVar, floatVar, charVar);
return 0;
}

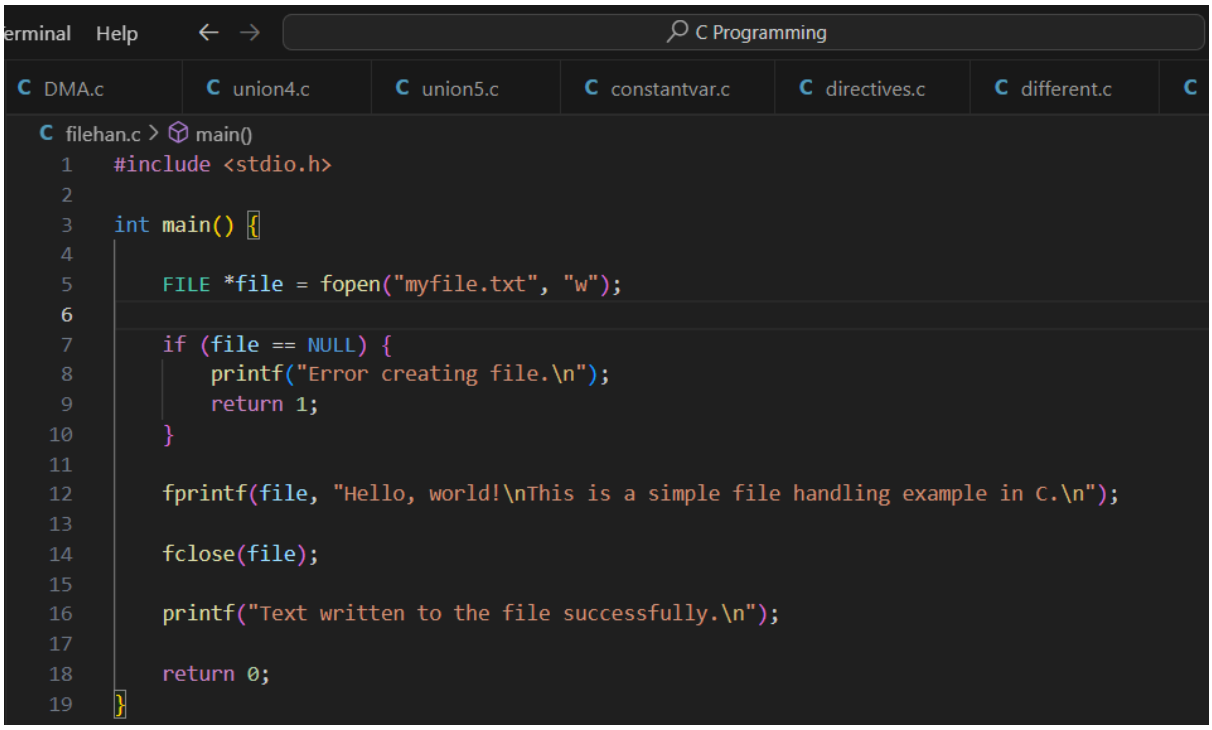
Output :


PROBLEMS
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
PS C:\Users\user\OneDrive\Desktop\C Programming> cd "c:\Users\user\OneDrive\Desktop\C Pr
}
Before modification:
intVar: 10, floatVar: 5.50, charVar: A
After modification:
intVar: 100, floatVar: 3.14, charVar: Z
PS C:\Users\user\OneDrive\Desktop\C Programming>

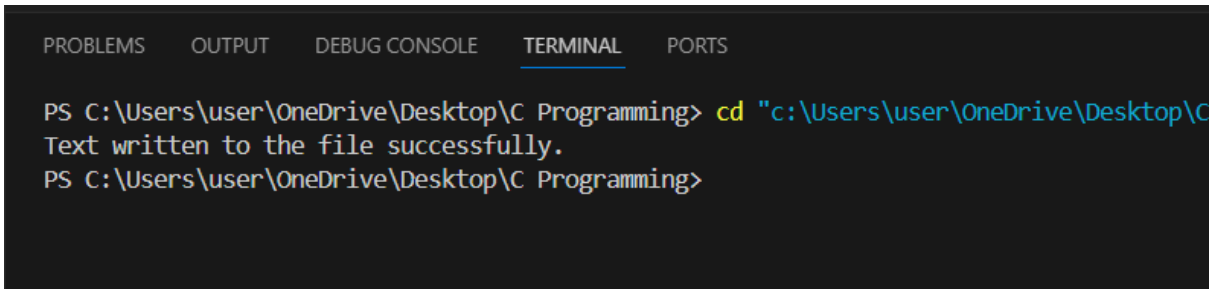
36) File handling in C

1) Write a program to create a new file and write text into it.

Input :

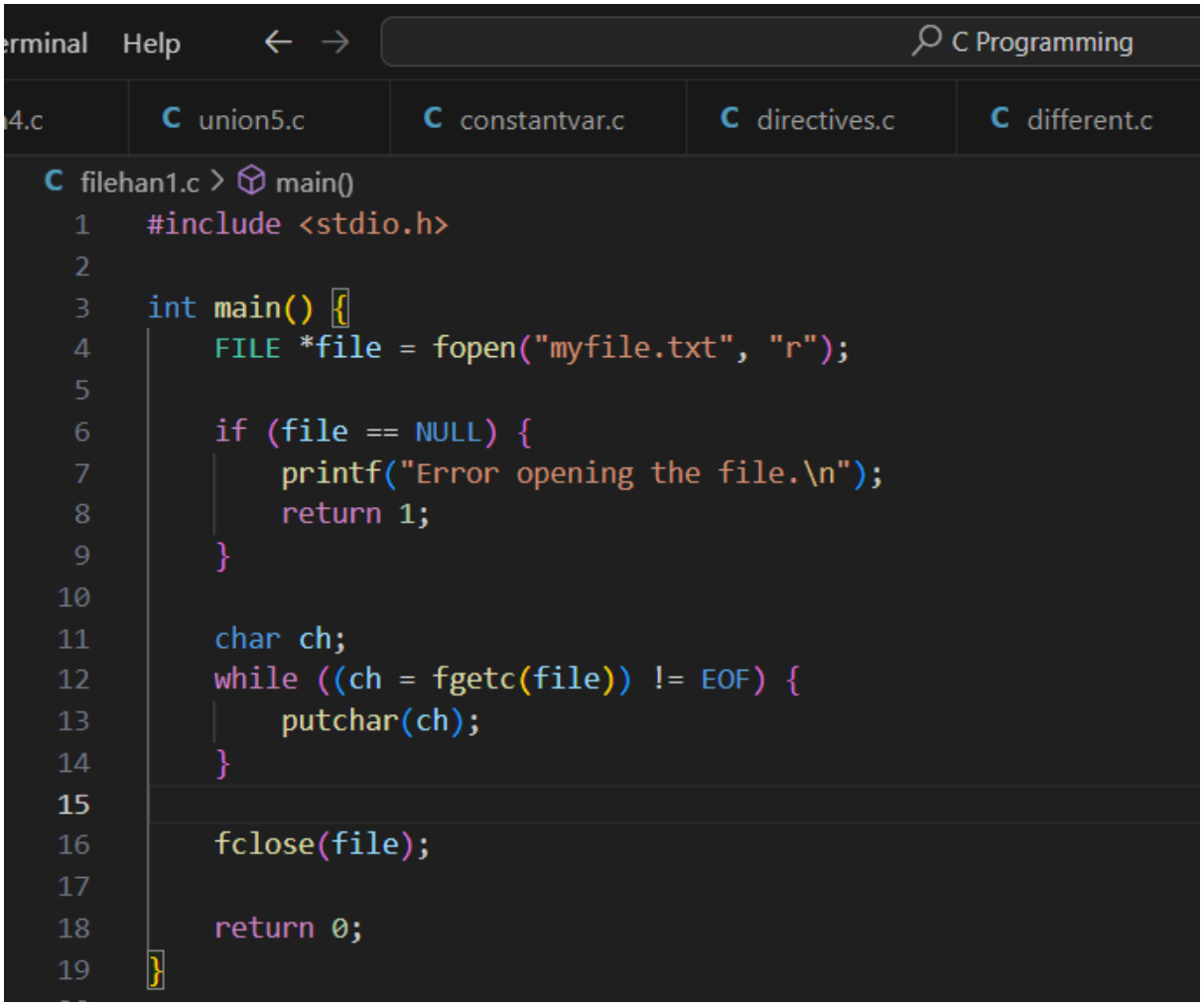


Output :

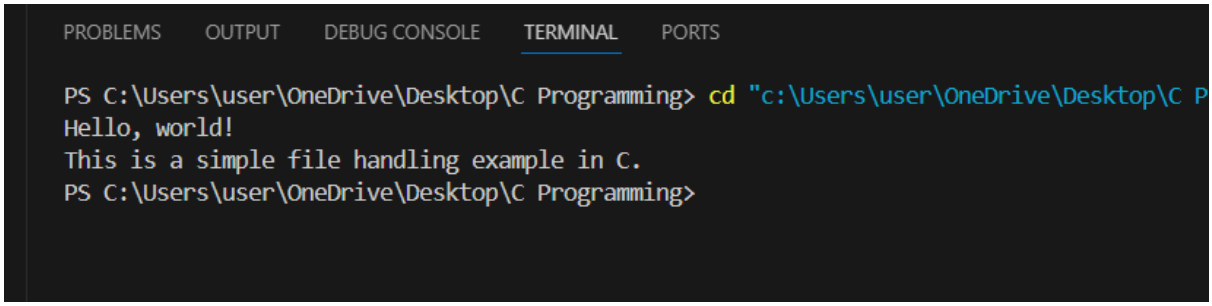


2) Open an existing file and read its content character by character, and then close the file.

Input :

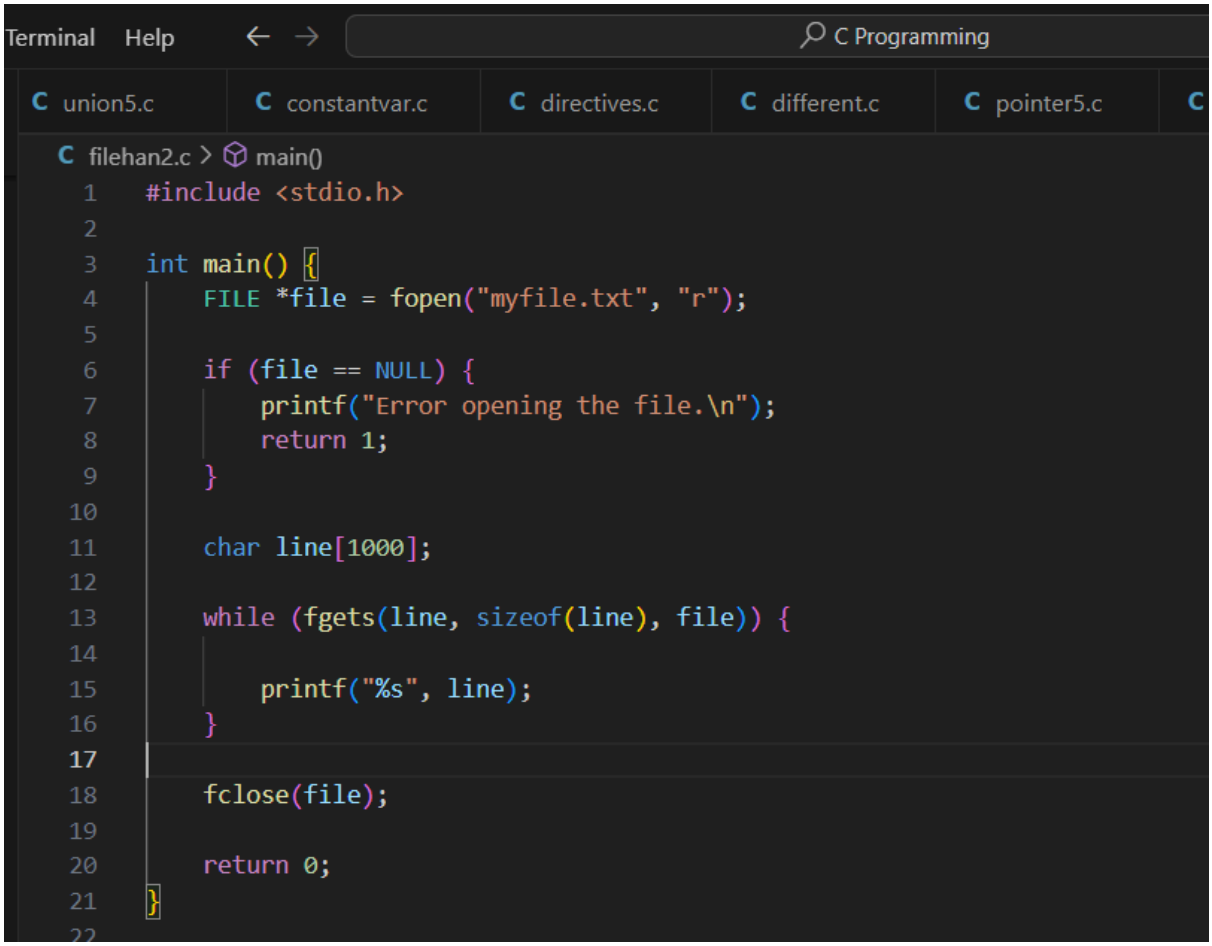


Output :



3) Open a file, read its content line by line, and display each line on the console.

Input :



Output :