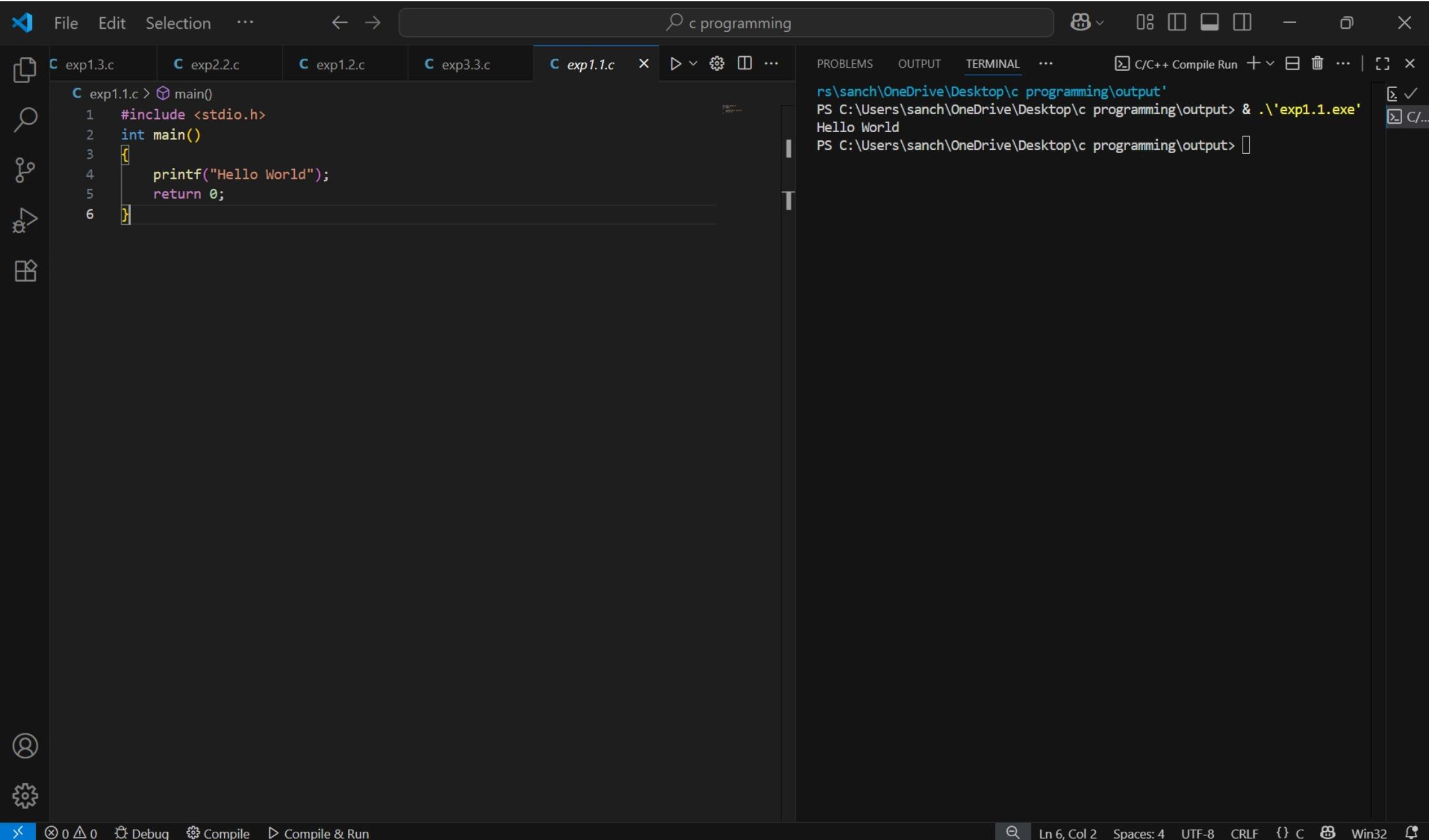


Exp = 1.1

→ Write a C program to print "Hello World"

⇒

```
# include < stdio.h >
int main ()
{
    printf ("Hello World");
    return 0 ;
}
```



Exp = 1.2

→ Write a program to print address in new line

→

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

```
int main()
```

{

```
    printf ("Name : Sanchita Luthra \n");
```

```
    printf ("Address : Bidholi \n");
```

```
    printf ("Dehradun, uttarakhand \n");
```

```
    printf ("India \n");
```

```
    return 0
```

}

The screenshot shows a dark-themed code editor interface with several panels and icons.

Top Bar: File, Edit, Selection, View, Go, ... (with arrows), a search bar containing "c programming", and various window control icons (minimize, maximize, close).

Sidebar Icons: Document, Magnifying glass, Open folder, Open recent, Help, Settings.

Left Panel: A list of files: exp1.3.c, exp2.2.c, and exp1.2.c (the active tab).

Code Editor: The content of the active file, exp1.2.c, is displayed:exp1.2.c > main()
1 #include <stdio.h>
2 int main()
3 {
4 printf("Name : Sanchita Luthra\n");
5 printf("Address : Bidholi\n");
6 printf(" Dehradun , Uttarakhand\n");
7 printf(" India ");
8 return 0;
9 }

Terminal: The terminal panel shows the command-line session:PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp1.2.exe
Name : Sanchita Luthra
Address : Bidholi
 Dehradun , Uttarakhand
 India
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>

Bottom Bar: Icons for Debug, Compile, Compile & Run, and status information (Ln 9, Col 2, Spaces: 2, UTF-8, CRLF, {}, C, Win32, a lock icon).

$$\text{Exp} = 1.3$$

→ Write a C program that prompts the user to enter their name & age

→

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

```
int main()
```

```
{
```

```
    char name;
```

```
    printf("enter your name \n");
```

```
    scanf("%s", &name);
```

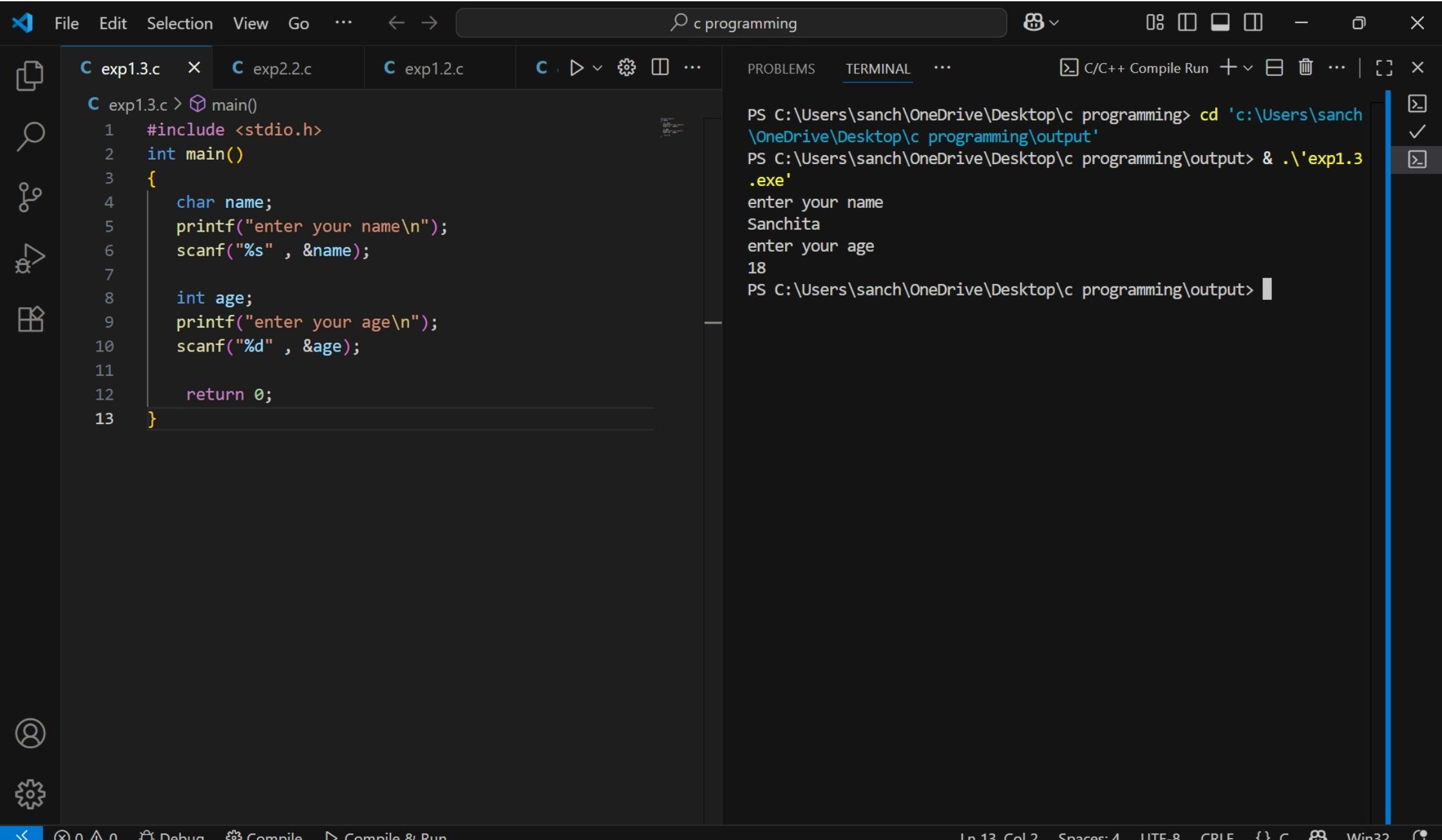
```
    int age;
```

```
    printf("enter your age \n");
```

```
    scanf("%d", &age);
```

```
    return 0;
```

```
}
```



$$\text{Exp} = 1.4$$

→ Write a C program to add two numbers, take number from user

→

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

```
int main()
```

```
{
```

```
int a, b, c;
```

```
printf ("enter first number \n");
```

```
scanf ("%d", &a);
```

```
printf ("enter second number \n");
```

```
scanf ("%d", &b);
```

$$c = a + b$$

```
printf ("the sum of two numbers is : %d \n", c);
```

```
return 0;
```

```
}
```

A screenshot of a dark-themed code editor, likely Visual Studio Code, displaying a C programming file named `exp1.4.c`. The code prompts the user to enter two numbers and prints their sum.

```
C exp1.4.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     int a,b,c;
5
6
7     printf("enter first number\n");
8     scanf("%d" , &a);
9
10    printf("enter second number\n");
11    scanf("%d" , &b);
12
13    c = a + b;
14    printf("the sum of two number is : %d\n" ,c );
15
16    return 0;
17 }
```

The terminal tab shows the execution of the program:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp1.4.exe
enter first number
26
enter second number
14
the sum of two number is : 40
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

At the bottom, the status bar indicates the current position is Line 17, Column 2, with 2 spaces, using UTF-8 encoding, and the file is a Win32 application.

Exp = 2.1

→ Write a C program to find the area & perimeter of rectangle based on its length & width

→

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

```
int main ()
```

```
{
```

```
    int length , width , area , perimeter ;
```

```
    printf (" enter the length of rectangle \n " ) ;
```

```
    scanf ("%d" , &length ) ;
```

```
    printf (" enter the width of rectangle \n " ) ;
```

```
    scanf ("%d" , &width ) ;
```

```
    area = length * width ;
```

```
    perimeter = 2 * ( length + width ) ;
```

```
    printf (" The area of rectangle is %d \n " , area ) ;
```

```
    printf (" The perimeter of rectangle is %d " , perimeter ) ;
```

```
    return 0 ;
```

```
}
```

Teacher's Signature _____

The screenshot shows a dark-themed code editor interface with the following details:

- File Bar:** File, Edit, Selection, View, Go, ...
- Search Bar:** c programming
- Toolbar:** Includes icons for file operations like Open, Save, Find, and others.
- Left Sidebar:** Icons for Selection, View, Go, and other tools.
- Editor Tabs:** .2.c, exp3.1.c, exp1.4.c, exp2.1.c (active tab).
- Editor Content:** The code for `exp2.1.c` is displayed. It includes declarations for `length`, `width`, `area`, and `perimeter`. It prompts the user to enter the length and width of a rectangle, calculates the area and perimeter, and prints the results.

```
C exp2.1.c > main()
1 #include <stdio.h>
2 int main() {
3     int length,width,area,perimeter;
4     printf("enter the length of rectangle\n");
5     scanf("%d" ,&length);
6
7     printf("enter the width of rectangle\n");
8     scanf ("%d" ,&width);
9
10    area = length*width;
11    perimeter = 2*(length + width);
12
13    printf("the area of rectangle is %d \n",area);
14    printf("the perimeter of rectangle is %d ",perimeter);
15    return 0;
16 }
```

- Terminal:** Shows the command-line session output. The user navigates to the directory `c:\Users\sanch\OneDrive\Desktop\c programming\output`, runs the executable `p2.1.exe`, and enters the values 8 and 4 when prompted for length and width respectively. The output shows the calculated area (32) and perimeter (24).

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\'exp2.1.exe'
enter the length of rectangle
8
enter the width of rectangle
4
the area of rectangle is 32
the perimeter of rectangle is 24
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

- Bottom Bar:** Includes icons for Debug, Compile, Compile & Run, and status indicators for rows, columns, spaces, encoding, and file type (Win32).

Exp = 2.2

→ Write a C program to convert temperature from celsius to fahrenheit Using formula $F = (c * 9/5) + 32$

→

include < stdio.h >

int main()

{

int celsius, fahrenheit

printf(" enter temperature in celsius in ");

scanf(" %d ", &celsius);

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

printf (" Temperature in fahrenheit is %d ", fahrenheit);

return 0 ;

}

Teacher's Signature _____

The screenshot shows a dark-themed code editor interface, likely Visual Studio Code, with several windows open. The main window displays a C program for converting Celsius to Fahrenheit. The terminal window shows the execution of the program and its output.

Code Editor:

```
exp2.2.c > ...
1 #include <stdio.h>
2
3 int main() {
4     int celsius,fahrenheit;
5
6     printf("enter temperature in celsius\n");
7     scanf("%d" , &celsius);
8
9     fahrenheit = (celsius*9/5)+32;
10    printf("Temperature in fahrenheit is %d" ,fahrenheit);
11
12    return 0;
13 }
```

Terminal Output:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\'exp2.2.exe'
enter temperature in celsius
60
Temperature in fahrenheit is 140
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

Bottom Status Bar:

Ln 14, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32

Exp = 3.1.1

→ 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 triangle as input from user

→ #include < stdio.h >

mt main ()

{

mt a, b, c ;

printf ("enter first side of the triangle : \n");

scanf ("%d", &a);

printf ("enter second side of the triangle : \n");

scanf ("%d", &b);

printf ("enter third side of the triangle : \n");

scanf ("%d", &c);

if (a+b > c && b+c > a && c+a > b)

{

printf ("the triangle is valid \n");

```
if (a==b && b==c)
{
    printf ("It is an equilateral triangle \n");
}
else if (a==b || b==c || a==c)
{
    printf ("It is an isosceles triangle \n");
}
else if (a*a + b*b == c*c || b*b + c*c == a*a || a*a + c*c == b*b)
{
    printf ("It is a right angled triangle \n");
}
else {
    printf ("It is a scalene triangle \n");
}
else {
    printf ("the triangle is not valid");
}
return 0;
```

A screenshot of a C programming environment in a dark-themed IDE. The interface includes a top bar with File, Edit, Selection, and other standard menu items. A search bar is positioned above the tabs. Below the tabs, there are several code editor panes and a terminal window.

The tabs at the top show the following files:

- p2.2.c
- C exp1.2.c
- C exp3.3.c
- C exp3.2.c
- C exp3.1.c (selected)
- C exp1.4.c
- C exp2

The terminal window on the right shows the command-line interface for running the program:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp3.1.exe
enter first side of the triangle :
3
enter second side of the triangle :
5
enter third side of the triangle :
10
the triangle is not valid
PS cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp3.1.exe
enter first side of the triangle :
3
enter second side of the triangle :
6
enter third side of the triangle :
6
the triangle is valid
it is an isosceles triangle
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

The main code editor pane contains the following C code:

```
#include <stdio.h>
int main()
{
    int a , b , c;

    printf("enter first side of the triangle :\n");
    scanf("%d" , &a);

    printf("enter second side of the triangle :\n");
    scanf("%d" , &b);

    printf("enter third side of the triangle :\n");
    scanf("%d" , &c);

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

        if( a==b && b==c)
        { printf("it is an equilateral triangle\n");
        }
        else if( a==b || b==c || a==c)
        { printf("it is an isosceles triangle\n");
        }
        else if( a * a + b * b == c * c || b * b + c * c == a * a || a * a + c * c == b * b)
        { printf("it is a right angled triangle\n");
        }
        else {
            printf("it is a scalene triangle\n");
        }
    } else {
        printf("the triangle is not valid");
    }
    return 0;
}
```

At the bottom, there are icons for file operations, a status bar showing line 6, column 52, and various file formats like UTF-8, CRLF, and Win32, along with a settings gear icon.

Exp = 3.1.2

→ WAP to compute the BMI Index of the person of the person and print the BMI values as per the following ranges. You can use the following formula to compute $BMI = \text{weight} / \text{height}^2$



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

```
int main ()
```

```
{
```

```
float weight, height, bmi;
```

```
printf (" Enter weight in Kilograms : \n " );  
scanf (" %f ", & weight );
```

```
printf (" Enter height in meters : \n " );  
scanf (" %.2f ", & height );
```

```
bmi = weight / (height * height) ;
```

```
printf (" YOUR BMI IS : %f \n " , bmi ) ;
```

```
printf (" Category: " );
```

Teacher's Signature _____

```
if (bmi < 15.0)
    printf("starvation\n");
else if (bmi >= 15.1 && bmi <= 17.5)
    printf("Underweight\n");
else if (bmi > 17.6 && bmi < 18.5)
    printf("Anorexic\n");
else if (bmi >= 18.6 && bmi <= 24.9)
    printf("Ideal\n");
else if (bmi >= 25.0 && bmi <= 25.9)
    printf("Overweight\n");
else if (bmi >= 30.0 && bmi <= 39.9)
    printf("Obese\n");
else if (bmi >= 40.0)
    printf("Morbidity Obese\n");
else
    printf("Not categorized by the given ranges\n");
```

return 0;

}

A screenshot of a C programming environment in Visual Studio Code (VS Code). The interface includes a top bar with File, Edit, Selection, and other icons. Below the top bar is a search bar containing "c programming". The left sidebar has icons for file operations like Open, Save, Find, and Settings.

The main area shows several tabs for C files: exp1.2.c, exp3.3.c, exp3.2.c (active), and exp3.1.c. The code editor displays the content of exp3.2.c, which calculates BMI based on user input for weight and height, and categorizes the result. The terminal tab shows the command-line output of running the program, including the BMI calculation and category determination.

```
C exp3.2.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     float weight , height , bmi;
6
7     printf("Enter weight in kilograms: \n");
8     scanf("%f", &weight);
9
10    printf("Enter height in meters: \n");
11    scanf("%f", &height);
12
13    bmi = weight / (height * height);
14
15    printf("YOUR BMI IS : %f\n", bmi);
16    printf("Category: ");
17
18    if (bmi < 15.0)
19        printf("Starvation\n");
20    else if (bmi >= 15.1 && bmi <= 17.5)
21        printf("Anorexic\n");
22    else if (bmi >= 17.6 && bmi <= 18.5)
23        printf("Underweight\n");
24    else if (bmi >= 18.6 && bmi <= 24.9)
25        printf("Ideal\n");
26    else if (bmi >= 25.0 && bmi <= 25.9)
27        printf("Overweight\n");
28    else if (bmi >= 30.0 && bmi <= 39.9)
29        printf("Obese\n");
30    else if (bmi >= 40.0)
31        printf("Morbidity Obese\n");
32    else
33        printf("Not categorized by the given ranges\n");
34
35    return 0;
36 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS C/C++ Compile Run

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp3.2.exe
Enter weight in kilograms:
70
Enter height in meters:
1.75
YOUR BMI IS : 22.857143
Category: Ideal
PS C:\Users\sanch\OneDrive\Desktop\c programming\output>
```

Ln 34, Col 1 Spaces: 4 UTF-8 CRLF {} C Win32

Exp = 3.1.3

→ WAP to check if three points (x_1, y_1) , (x_2, y_2) & (x_3, y_3) are collinear or not

→

include <stdio.h>

int main ()

{

float $x_1, y_1, x_2, y_2, x_3, y_3$;
float area ;

printf ("Enter coordinates of first point (x_1, y_1) : \n");
scanf ("%f %f", & x_1 , & y_1) ;

printf ("Enter coordinates of second point (x_2, y_2) : \n");
scanf ("%f %f", & x_2 , & y_2) ;

printf ("Enter coordinates of third point (x_3, y_3) : \n");
scanf ("%f %f", & x_3 , & y_3) ;

area = $(x_1 * (y_2 - y_3) + x_2 * (y_3 - y_1) + x_3 * (y_1 - y_2)) / 2$

Teacher's Signature _____

if (area == 0) .

 printf ("The points are collinear.\n");

else

 printf ("The points are not collinear.\n");

return 0;

}

The screenshot shows a dark-themed code editor interface with several windows open. The main window displays a C program named `exp3.3.c` which calculates the area of a triangle given three vertices and checks if they are collinear. The code uses standard input-output functions from `<stdio.h>`. The terminal window shows the execution of the program, where the user inputs coordinates for three points. In the first run, points (1, 1), (2, 3), and (3, 5) are entered, resulting in a collinear output. In the second run, points (0, 0), (2, 3), and (4, 1) are entered, resulting in a non-collinear output.

```
C exp3.3.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     float x1 , y1 , x2 , y2 , x3 , y3;
6     float area;
7
8     printf("Enter coordinates of first point (x1 y1):\n");
9     scanf("%f %f", &x1 , &y1);
10
11    printf("Enter coordinates of second point (x2 y2):\n");
12    scanf("%f %f", &x2 , &y2);
13
14    printf("Enter coordinates of third point (x3 y3):\n");
15    scanf("%f %f", &x3 , &y3);
16
17    area = (x1 * (y2 - y3) + x2 * (y3 - y1) + x3 * (y1 - y2)) / 2.0;
18
19    if (area == 0)
20        printf("The points are collinear.\n");
21    else
22        printf("The points are not collinear.\n");
23
24    return 0;
25 }
```

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp3.3.exe
Enter coordinates of first point (x1 y1):
1 1
Enter coordinates of second point (x2 y2):
2 3
Enter coordinates of third point (x3 y3):
3 5
The points are collinear.
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\output> & .\exp3.3.exe
Enter coordinates of first point (x1 y1):
0 0
Enter coordinates of second point (x2 y2):
2 3
Enter coordinates of third point (x3 y3):
4 1
The points are not collinear.
```