

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 multiple tabs and a terminal window.

Code Editor:

- Tab bar: exp1.3.c, exp2.2.c, exp1.2.c, exp3.3.c (active), exp3.2.c
- File content (exp3.3.c):

```
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 }
```

Terminal:

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

Bottom Bar:

- Icons: File, Edit, Selection, ..., Back, Forward, Search, Problems, Terminal, ...
- Text: PROBLEMS TERMINAL ...
- Text: C/C++ Compile Run
- Text: Line 21, Column 9, Spaces: 4, UTF-8, CRLF, {}, C, Win32, etc.

Exp 3.1.4

→ According to the gregorian calender, it was Monday on date 01/01/31. If any year is input through the keyboard write a program to find out what is the day on 1st january of this year

→

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

```
int main()
```

{

```
int year, i, leapyears = 0, normalyears,  
totalshifts, dayIndex;  
char *days[] = {"Monday", "Tuesday", "Wednesday",  
"Thursday", "Friday", "Saturday", "Sunday"};
```

```
printf("Enter a year : ");  
scanf("%d", &year);
```

```
for (i=1 ; i < year ; i++) {  
if ((i % 400 == 0) || (i % 4 == 0 && i % 100 != 0))  
leapyears++
```

Teacher's Signature _____

{

 $\text{normalYears} = (\text{Year} - 1) - \text{leapYears};$ $\text{totalShifts} = (\text{normalYears} * 1 + \text{leapYears} * 2) \% 7;$ $\text{dayIndex} = \text{totalShifts} \% 7;$

if (year <= 2025)

{

printf("On 01/01/%d, it was %s\n", year,
days[dayIndex]);

{

else

{

printf("On 01/01/%d, it will be %s\n", year,
days[dayIndex]);

{

return 0;

{

The screenshot shows a C programming environment with the following details:

- File:** exp4.c
- Code Content:** A C program to calculate the day of the week for a given year. It includes leap year logic and uses an array of days of the week.

```
exp_3.1.c > C exp4.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     int year, i, leapYears = 0, normalYears, totalShifts, dayIndex;
6     char *days[] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};
7
8     printf("Enter a year : ");
9     scanf("%d", &year);
10
11    for (i = 1; i < year; i++) {
12        if ((i % 400 == 0) || (i % 4 == 0 && i % 100 != 0))
13            leapYears++;
14    }
15
16    normalYears = (year - 1) - leapYears;
17
18    totalShifts = (normalYears * 1 + leapYears * 2) % 7;
19
20    dayIndex = totalShifts % 7;
21
22    if (year<=2025)
23    {
24        printf("On 01/01/%d, it was %s\n", year, days[dayIndex]);
25    }
26    else
27    {
28        printf("On 01/01/%d, it will be %s\n", year, days[dayIndex]);
29    }
30
31
32    return 0;
33
34 }
```

- Output Window:** Shows command-line output from a terminal window.

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output> &.\'exp4.exe'
Enter a year : 2021
On 01/01/2021, it was Friday
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output> cd 'c:\Users\sanch\OneD
rive\Desktop\c programming\exp_3.1.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output> &.\'exp4.exe'
Enter a year : 2029
On 01/01/2029, it will be Monday
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output> 
```

Exp - 3.1.5

→

WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles should be three

→

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

```
int main ()
```

{

```
int l1 , b1 , l2 , b2 , l3 , b3 , p1 , p2 ,  
p3 , max ;
```

```
printf ("Enter length and breadth of rectangle 1: ");  
scanf ("%d %d" , &l1 , &b1) ;
```

```
printf ("Enter length and breadth of rectangle 2: ");  
scanf ("%d %d" , &l2 , &b2) ;
```

```
printf ("Enter length and breadth of rectangle 3: ");  
scanf ("%d %d" , &l3 , &b3) ;
```

Teacher's Signature _____

$$P_1 = 2 * (l_1 + b_1);$$

$$P_2 = 2 * (l_2 + b_2);$$

$$P_3 = 2 * (l_3 + b_3);$$

max = ($P_1 > P_2$) ? (($P_1 > P_3$) ? $P_1 : P_3$) : (($P_2 > P_3$) ? $P_2 : P_3$);

if (max == p1)

printf ("Rectangle 1 has the highest perimeter:
%d\n", p1);

else if (max == p2)

printf ("Rectangle 2 has the highest perimeter:
%d\n", p2);

else

printf ("Rectangle 3 has the highest perimeter: %d
\\n", p3);

return 0;

}

The screenshot shows a C/C++ development environment with the following details:

- File Bar:** File, Edit, Selection, View, Go, ...
- Search Bar:** c programming
- Toolbars:** Standard toolbar icons.
- Project Explorer:** Shows files exp4.c and exp5.c.
- Code Editor:** The code for `exp5.c` is displayed, which calculates the perimeters of three rectangles and prints the one with the highest perimeter.
- Output Window:** Displays the terminal output of the program execution.
- Status Bar:** Shows the current line (Ln 20, Col 69), spaces used (Spaces: 4), encoding (UTF-8), file type (CRLF), and window title (Win32).

```
exp_3.1.c > exp5.c > main()
2
3 int main() {
4     int l1, b1, l2, b2, l3, b3, p1, p2, p3, max;
5
6     printf("Enter length and breadth of rectangle 1: ");
7     scanf("%d %d", &l1, &b1);
8
9     printf("Enter length and breadth of rectangle 2: ");
10    scanf("%d %d", &l2, &b2);
11
12    printf("Enter length and breadth of rectangle 3: ");
13    scanf("%d %d", &l3, &b3);
14
15
16    p1 = 2 * (l1 + b1);
17    p2 = 2 * (l2 + b2);
18    p3 = 2 * (l3 + b3);
19
20    max = (p1 > p2) ? ((p1 > p3) ? p1 : p3) : ((p2 > p3) ? p2 : p3);
21
22    if (max == p1)
23        printf("Rectangle 1 has the highest perimeter: %d\n", p1);
24    else if (max == p2)
25        printf("Rectangle 2 has the highest perimeter: %d\n", p2);
26    else
27        printf("Rectangle 3 has the highest perimeter: %d\n", p3);
28
29    return 0;
30 }
```

PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output> & .\exp5.exe
Enter length and breadth of rectangle 1: 2 3
Enter length and breadth of rectangle 2: 4 3
Enter length and breadth of rectangle 3: 8 2
Rectangle 3 has the highest perimeter: 20
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.1.c\output>

Exp 3.2.1

→

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

→

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

```
int main()
```

{

```
int positive = 0, negative = 0, zero = 0, num,  
cont;
```

```
do
```

{

```
printf ("enter number \n");  
scanf ("%d", &num);
```

```
if (num > 0)
```

```
positive++;
```

```
else if (num < 0)
```

```
negative++;
```

else

zero ++

printf (" if you want to enter another press 1 else
press 0 : ");

scanf ("%d", &cont);

}

while (cont == 1)

printf (" positive number entered : %d\n", positive);

printf (" negative number entered : %d\n", negative);

printf (" zero number entered : %d\n", zero);

return 0;

}

A screenshot of a C programming environment in a dark-themed IDE. The interface includes a top navigation bar with File, Edit, Selection, and other standard options. A search bar is positioned at the top center. Below the search bar, there are tabs for exp4.c and exp1.c, with exp1.c currently active. The main area contains the source code for exp1.c, which counts positive, negative, and zero numbers entered by the user. The code uses standard input/output functions like printf and scanf. To the right of the code editor is a terminal window showing the execution of the program. The terminal output shows the user entering several numbers (43, 89566, -5956, 0, 0, -95) and the program responding with prompts and counts (positive, negative, zero numbers). The bottom of the screen features a toolbar with icons for file operations, a status bar indicating the current line and column (Ln 15, Col 20), and various system status indicators.

```
exp_3.2.c > C exp1.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     int positive=0 , negative=0 , zero=0 , num , cont ;
6
7     do
8     {
9         printf("enter number\n");
10        scanf("%d", &num);
11
12        if (num>0)
13            positive++;
14        else if (num<0)
15            negative++;
16        else
17            zero++;
18
19        printf("if you want to enter another number press 1 else press 0 : ");
20        scanf("%d", &cont);
21    }
22    while (cont==1);
23
24    printf("positive number entered : %d\n", positive);
25    printf("negative number entered : %d\n", negative);
26    printf("zero number entered : %d\n", zero);
27
28    return 0;
29 }
```

```
ing> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output> & .\'exp1.exe'
enter number
43
if you want to enter another number press 1 else press 0 : 1
enter number
89566
if you want to enter another number press 1 else press 0 : 1
enter number
-5956
if you want to enter another number press 1 else press 0 : 1
enter number
0
if you want to enter another number press 1 else press 0 : 1
enter number
0
if you want to enter another number press 1 else press 0 : 1
enter number
-95
if you want to enter another number press 1 else press 0 : 0
positive number entered : 2
negative number entered : 2
zero number entered : 2
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output>
```

Exp 3.2.2

→

WAP to print the multiplication table of the number entered by the user it should be in the correct formatting $\text{Num} * 1 = \text{Num}$

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

```
int main()
```

```
{
```

```
    int num, i, result ;
```

```
    printf (" enter a number : \n ");
```

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

```
    printf (" multiplication table of %d is \n ", num) ;
```

```
    for ( i=1 ; i<=10 ; i++ )
```

```
{
```

```
    result = num + i ;
```

```
    printf (" %d * %d = %d \n ", num, i, result) ;
```

```
}
```

```
return 0 ;
```

```
}
```

Teacher's Signature _____

The screenshot shows a dark-themed code editor interface with several panels:

- Left Panel:** Includes icons for file operations (New, Open, Save, Find, Replace, Copy, Paste, Delete, Undo, Redo), a search bar, and a terminal window.
- Top Bar:** Contains the application name (VS Code), file menu (File, Edit, Selection, View, Go, ...), a search bar (c programming), and various system icons (minimize, maximize, close).
- Code Editor:** Displays two files: `exp4.c` and `exp2.c`. The `exp4.c` file contains the following C code:

```
exp_3.2.c > exp2.c > main()
1 #include <stdio.h>
2 int main()
3 {
4     int num , i , result;
5     printf("enter a number: \n");
6     scanf("%d" , &num);
7
8     printf("multiplication table of %d is\n", num);
9
10    for ( i = 1; i <= 10; i++)
11    {
12        result = num * i;
13        printf("%d * %d = %d \n" , num , i , result);
14    }
15    return 0;
16 }
```

- Terminal:** Shows the command-line interface output for the `exp2.c` program. It includes the current directory (`C:\Users\sanch\OneDrive\Desktop\c programming`), the command (`cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output'`), the execution of the program (`PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output> & .\exp2.exe`), and the multiplication table of 12.
- Bottom Bar:** Includes icons for file operations, a status bar showing the current line (Ln 16, Col 2), character count (Spaces: 4), encoding (UTF-8), line endings (CRLF), file type ({} C), and system information (Win32).

Exp - 3.2.3

→ WAP to generate the following output

a 1
 2 3
 4 5 6

b 1
 1 1
 1 2 1
 1 3 3 1
 1 4 6 4 1

→

include < stdio.h >

```
int main () {  
    int i, j, num;  
  
    printf (" Part (a) : \n ");  
    num = 1 ;  
    for ( i = 1 ; i <= 3 ; i++ ) {  
        for ( j = 1 ; j <= i ; j++ ) {  
            printf (" %d ", num );
```

Teacher's Signature _____

```
    num ++ ;  
}
```

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

```
printf ("In Part (b) : \n");
```

```
for (i=0; i < 5; i++) {
```

```
    num = 1 ;
```

```
    for (j=0; j <= i; j++) {
```

```
        printf (" .d", num);
```

```
        num = num * (i-j)/(j+1);
```

```
}
```

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

```
}
```

```
return 0 ;
```

```
}
```

A screenshot of a C/C++ development environment, likely Visual Studio Code, showing a file named `exp4.c` open in the editor. The code contains two parts, (a) and (b), which print numerical patterns to the console. The terminal window shows the execution of the program and its output.

The code in `exp4.c`:

```
exp_3.2.c > exp3.c > main()
1 #include <stdio.h>
2
3 int main() {
4     int i, j, num;
5
6     printf("Part (a):\n");
7
8     num = 1;
9     for (i = 1; i <= 3; i++) {
10         for (j = 1; j <= i; j++) {
11             printf("%d ", num);
12             num++;
13         }
14         printf("\n");
15     }
16
17     printf("\nPart (b):\n");
18     for (i = 0; i < 5; i++) {
19         num = 1;
20         for (j = 0; j <= i; j++) {
21             printf("%d ", num);
22             num = num * (i - j) / (j + 1);
23         }
24         printf("\n");
25     }
26
27     return 0;
28 }
```

The terminal output:

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output> & .\exp3.exe
Part (a):
1
2 3
4 5 6

Part (b):
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output>
```

Status message:

Compilation successful.

File menu: File Edit Selection View Go ...

Search bar: c programming

Toolbar icons: Save, Open, Find, Replace, Copy, Paste, Cut, Undo, Redo, Find in files, Open recent, Open folder, Open workspace, Open settings, Open extensions, Open themes, Open help, Open about, Open command palette.

Editor tabs: exp4.c, exp3.c

Editor status: Line 17, Column 29, Spaces: 4, UTF-8, CRLF, {} C, Win32

Exp- 3.2.4

→

The population of 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

→

include < stdio.h >

mt main ()

{

mt year;

float population = 100000;

printf (" Population in the last decade : \n");

for (year = 1 ; year <= 10 ; year ++)

{

population = population + population * 0.1 ;

printf (" year %.d : %.0f \n ", year , population);

return 0 ;

{

A screenshot of a dark-themed code editor, likely Visual Studio Code, displaying a C programming file. The left pane shows the code for `exp4.c`, which calculates population growth over 10 years starting from 1,000,000. The right pane shows the terminal output where the program is run, displaying the population for each year from 1 to 10.

```
exp_3.2.c > exp4.c > main()
1 #include <stdio.h>
2
3 int main()
4 {
5     int year;
6     float population = 100000;
7
8     printf("Population growth in the last decade :\n");
9
10    for (year = 1; year <= 10; year++)
11    {
12        population = population + population * 0.1 ;
13        printf("Year %d: %.0f\n", year, population);
14    }
15
16    return 0;
17 }
```

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output> & .\exp4.exe
Population growth in the last decade :
Year 1: 110000
Year 2: 121000
Year 3: 133100
Year 4: 146410
Year 5: 161051
Year 6: 177156
Year 7: 194872
Year 8: 214359
Year 9: 235795
Year 10: 259374
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output>
```

The status bar at the bottom indicates the current line and column (Ln 12, Col 52), character encoding (UTF-8), and file type (C). The bottom navigation bar includes icons for file operations like New, Open, Save, and Close, as well as Debug, Compile, and Run.

Exp - 3.2.5

→

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

→

include < stdio.h >

int main () {

int a, b, c, d, L ;

int sum1, sum2 ;

printf("Enter the limit (L) : ");

scanf ("%d", &L);

printf (" Ramanujan numbers up to %d ^ 3 are :\nIn ", L);

for (a=1 ; a <= L ; a++)

for (b=1 ; b <= L ; b++)

Teacher's Signature _____

{

$$\text{sum1} = a * a * a + b * b * b;$$

```
for (c=1; c <= L; c++)
```

{

```
    for (d=1; d <= L; d++) {
```

$$\text{sum2} = c * c * c + d * d * d;$$

```
    if (sum1 == sum2 && (a != c || b != d))
```

{

```
        printf ("%d - %d = %d - %d ^ 3 + %d - %d ^ 3\n",
```

```
            sum1, a, b, c, d);
```

}

}

}

}

}

```
return 0;
```

3

The screenshot shows a C programming environment in Visual Studio Code. The current file is `exp5.c`, which contains a program to find Ramanujan numbers up to L^3 . The code uses nested loops to iterate through all combinations of a, b, c, d from 1 to L . It calculates two sums, $sum1 = a^3 + b^3$ and $sum2 = c^3 + d^3$, and prints them if they are equal and $a \neq c$ and $b \neq d$.

```
1 #include <stdio.h>
2
3 int main() {
4     int a, b, c, d, L;
5     int sum1, sum2;
6
7     printf("Enter the limit (L): ");
8     scanf("%d", &L);
9
10    printf("Ramanujan numbers up to %d^3 are:\n", L);
11
12    for (a = 1; a <= L; a++) {
13        for (b = a; b <= L; b++) {
14            sum1 = a*a*a + b*b*b;
15
16            for (c = a; c <= L; c++) {
17                for (d = c; d <= L; d++) {
18                    sum2 = c*c*c + d*d*d;
19
20                    if (sum1 == sum2 && !(a == c && b == d)) {
21                        printf("%d = %d^3 + %d^3 = %d^3 + %d^3\n",
22                               sum1, a, b, c, d);
23                    }
24                }
25            }
26        }
27    }
28
29    return 0;
30 }
31
```

The terminal window shows the output of the program when run with $L = 12$. It finds one Ramanujan number, 1729, which is equal to $1^3 + 12^3 = 9^3 + 10^3$.

```
PS C:\Users\sanch\OneDrive\Desktop\c programming> cd 'c:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output'
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output> & .\exp5.exe
Enter the limit (L): 12
Ramanujan numbers up to 12^3 are:
1729 = 1^3 + 12^3 = 9^3 + 10^3
PS C:\Users\sanch\OneDrive\Desktop\c programming\exp_3.2.c\output>
```