



UPES

C PROGRAMMING LAB FILE

(1st Semester)

Name:- Sparsh Srivastava

SAP ID :- 590022251

Batch 33

Experiment 1: Installation, Environment Setup and starting with C language

1. Write a C program to print “ Hello World”

The screenshot shows the OnlineGDB interface. The code editor contains the following C program:

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

The terminal window below shows the output:

```
Hello World
...Program finished with exit code 0
Press ENTER to exit console.
```

2. Write a C Program to print the address in multiple lines (new line)

The screenshot shows the OnlineGDB interface. The code editor contains the following C program:

```
#include <stdio.h>
int main() {
    printf("Sparsh Srivastava\n");
    printf("First year\n");
    printf("UPES DEHRADUN\n");
    printf("from Faridabad, Haryana\n");
    return 0;
}
```

The terminal window below shows the output:

```
Sparsh Srivastava
First year
UPES DEHRADUN
from Faridabad, Haryana
...Program finished with exit code 0
Press ENTER to exit console.
```

3. Write a program that prompts the user to enter their name and age.

The screenshot shows the OnlineGDB interface. The code editor contains the following C program:

```
#include <stdio.h>
int main() {
    char name[50];
    int age;
    printf("Enter your name: ");
    scanf("%s", name);
    printf("Enter your age: ");
    scanf("%d", &age);
    printf("Hi %s, you are %d years old.\n", name, age);
    return 0;
}
```

The terminal window below shows the output:

```
Enter your name: Sparsh
Enter your age: 18
Hi Sparsh, you are 18 years old.
...Program finished with exit code 0
Press ENTER to exit console.
```

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

The screenshot shows the OnlineGDB interface. The code in the editor is:

```
main.c
1 #include <stdio.h>
2 int main() {
3     int num1, num2, sum;
4     printf("Enter first number: ");
5     scanf("%d", &num1);
6     printf("Enter second number: ");
7     scanf("%d", &num2);
8     sum = num1 + num2;
9     printf("Sum = %d\n", sum);
10 }
11
12 }
```

The terminal window shows the output of running the program:

```
input
Enter first number: 13
Enter second number: 13
Sum = 26

...Program finished with exit code 0
Press ENTER to exit console.
```

At the bottom, there is a footer with links: About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2025 GDB Online

Experiment 2: Operators:

1. WAP a C program to calculate the area and perimeter of a rectangle based on its length and width.

The screenshot shows the OnlineGDB interface. The code in the editor is:

```
main.c
1 #include <stdio.h>
2 int main() {
3     float length, width, area, perimeter;
4     printf("Enter length of the rectangle: ");
5     scanf("%f", &length);
6     printf("Enter width of the rectangle: ");
7     scanf("%f", &width);
8     area = length * width;
9     perimeter = 2 * (length + width);
10    printf("Area = %.2f\n", area);
11    printf("Perimeter = %.2f\n", perimeter);
12 }
13
14
15 }
```

The terminal window shows the output of running the program:

```
input
Enter length of the rectangle: 4
Enter width of the rectangle: 2
Area = 8.00
Perimeter = 12.00

...Program finished with exit code 0
Press ENTER to exit console.
```

At the bottom, there is a footer with links: About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2025 GDB Online

2. WAP a C program to Convert temperature from Celsius to Fahrenheit using the formula: $F = (C * 9/5) + 32$.

The screenshot shows the OnlineGDB interface. The code in the editor is:

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

The terminal window shows the output of running the program:

```
input
Enter temperature in Celsius: 35
Temperature in Fahrenheit = 95.00

...Program finished with exit code 0
Press ENTER to exit console.
```

At the bottom, there is a footer with links: About • FAQ • Blog • Terms of Use • Contact Us • GDB Tutorial • Credits • Privacy © 2016 - 2025 GDB Online

Experiment 3.1: 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.

The screenshot shows the OnlineGDB interface with a C program named 'main.c'. The code prompts the user to enter three sides of a triangle and then checks if they form a valid triangle. If valid, it further classifies it as Equilateral, Isosceles, Right-angled, or Scalene. The output window shows the program's execution and classification of a triangle with sides 2, 3, and 4 as a Scalene triangle.

```
#include <stdio.h>
int main() {
    float a, b, c;
    printf("Enter three sides of triangle: ");
    scanf("%f %f %f", &a, &b, &c);
    if (a + b > c && b + c > a && c + a > b) {
        printf("Valid triangle\n");
        if (a == b && b == c)
            printf("Equilateral triangle\n");
        else if (a == b || b == c || c == a)
            printf("Isosceles triangle\n");
        else
            printf("Scalene triangle\n");
        if ((a*a + b*b == c*c) || (b*b + c*c == a*a) || (c*c + a*a == b*b))
            printf("Right-angled triangle\n");
    } else {
        printf("Not a valid triangle\n");
    }
    return 0;
}
```

Input: Enter three sides of triangle: 2 3 4
Output: Valid triangle
Scalene triangle
...Program finished with exit code 0
Press ENTER to exit console.

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
 $BMI = \frac{\text{weight(kgs)}}{\text{Height(Mts)}^2}$.

The screenshot shows the OnlineGDB interface with a C program named 'main.c'. The code prompts the user to enter weight in kilograms and height in meters, then calculates BMI and prints its category. The output window shows the program's execution and classification of a person with weight 74 kg and height 1.78 m as Ideal.

```
#include <stdio.h>
int main() {
    float weight, height, bmi;
    printf("Enter weight (kg): ");
    scanf("%f", &weight);
    printf("Enter height (m): ");
    scanf("%f", &height);
    bmi = weight / (height * height);
    printf("BMI = %.2f\n", bmi);
    if (bmi < 15)
        printf("Starvation\n");
    else if (bmi <= 17.5)
        printf("Anorexic\n");
    else if (bmi <= 18.5)
        printf("Underweight\n");
    else if (bmi <= 24.9)
        printf("Ideal\n");
    else if (bmi <= 25.9)
        printf("Overweight\n");
    else if (bmi <= 39.9)
        printf("Obese\n");
    else
        printf("Morbidly Obese\n");
    return 0;
}
```

Input: Enter weight (kg): 74
Enter height (m): 1.78
Output: BMI = 23.36
Ideal
...Program finished with exit code 0
Press ENTER to exit console.

3. WAP to check if three points (x_1, y_1) , (x_2, y_2) and (x_3, y_3) are collinear or not.

The screenshot shows the OnlineGDB interface with a C program named 'main.c'. The code prompts the user to enter coordinates of three points and calculates the area of the triangle formed by them. If the area is zero, the points are collinear; otherwise, they are not. The output window shows the program's execution and confirmation that points (1, 2), (3, 4), and (5, 6) are collinear.

```
#include <stdio.h>
int main() {
    float x1, y1, x2, y2, x3, y3;
    printf("Enter coordinates of three points (x1 y1 x2 y2 x3 y3): ");
    scanf("%f %f %f %f %f %f", &x1, &y1, &x2, &y2, &x3, &y3);
    float area = 0.5 * ((x1*(y2 - y3)) + (x2*(y3 - y1)) + (x3*(y1 - y2)));
    if (area == 0)
        printf("Points are collinear.\n");
    else
        printf("Points are NOT collinear.\n");
    return 0;
}
```

Input: Enter coordinates of three points (x1 y1 x2 y2 x3 y3): 1 2 3 4 5 6
Output: Points are collinear.
...Program finished with exit code 0
Press ENTER to exit console.

- 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.**

The screenshot shows the OnlineGDB interface. The code in the editor is:

```

main.c
1 #include <stdio.h>
2 int main() {
3     int year, day;
4     printf("Enter year: ");
5     scanf("%d", &year);
6     day = (year + (year - 1)/4 - (year - 1)/100 + (year - 1)/400) % 7;
7     char *days[] = {"Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", "Sunday"};
8     printf("1st January %d is a %s\n", year, days[day]);
9     return 0;
10 }
11

```

The terminal window shows the output of running the program with the input '2029':

```

input
Enter year: 2029
1st January 2029 is a Tuesday

...Program finished with exit code 0
Press ENTER to exit console.

```

- 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.**

The screenshot shows the OnlineGDB interface. The code in the editor is:

```

main.c
1 #include <stdio.h>
2 int main() {
3     int l1, b1, l2, b2, l3, b3;
4     int p1, p2, p3, max;
5     printf("Enter length and breadth of rectangle 1: ");
6     scanf("%d %d", &l1, &b1);
7     printf("Enter length and breadth of rectangle 2: ");
8     scanf("%d %d", &l2, &b2);
9     printf("Enter length and breadth of rectangle 3: ");
10    scanf("%d %d", &l3, &b3);
11    p1 = 2 * (l1 + b1);
12    p2 = 2 * (l2 + b2);
13    p3 = 2 * (l3 + b3);
14    max = (p1 > p2 && p1 > p3) ? p1 : (p2 > p3 ? p2 : p3);
15    printf("Maximum perimeter is: %d\n", max);
16    return 0;
17 }
18

```

The terminal window shows the output of running the program with inputs 1 2, 2 3, and 4 5:

```

input
Enter length and breadth of rectangle 1: 1 2
Enter length and breadth of rectangle 2: 2 3
Enter length and breadth of rectangle 3: 4 5
Maximum perimeter is: 18

...Program finished with exit code 0
Press ENTER to exit console.

```

Experiment 3.2: Loops

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

The screenshot shows the OnlineGDB interface. The code in the editor is:

```
main.c
1 #include <stdio.h>
2 int main() {
3     int num, pos = 0, neg = 0, zero = 0;
4     char choice;
5     do {
6         printf("Enter a number: ");
7         scanf("%d", &num);
8         if (num > 0) pos++;
9         else if (num < 0) neg++;
10        else zero++;
11        printf("Do you want to enter another number? (y/n): ");
12        scanf(" %c", &choice);
13    } while (choice == 'y' || choice == 'Y');
14    printf("Positive: %d\nNegative: %d\nZeroes: %d\n", pos, neg, zero);
15    return 0;
16 }
```

The output window shows the following interaction:

```
input
Enter a number: -1
Do you want to enter another number? (y/n): y
Enter a number: 10
Do you want to enter another number? (y/n):
y
Enter a number: 0
Do you want to enter another number? (y/n): n
Positive: 1
Negative: 1
Zeroes: 1
```

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

The screenshot shows the OnlineGDB interface. The code in the editor is:

```
main.c
1 #include <stdio.h>
2 int main() {
3     int num;
4     printf("Enter a number: ");
5     scanf("%d", &num);
6     for (int i = 1; i <= 10; i++) {
7         printf("%d * %d = %d\n", num, i, num * i);
8     }
9     return 0;
10 }
```

The output window shows the multiplication table for the number 12:

```
input
12 * 1 = 12
12 * 2 = 24
12 * 3 = 36
12 * 4 = 48
12 * 5 = 60
12 * 6 = 72
12 * 7 = 84
12 * 8 = 96
12 * 9 = 108
12 * 10 = 120

...Program finished with exit code 0
Press ENTER to exit console.
```

3. WAP to generate the following set of output.

a. 1

2 3

4 5 6

The screenshot shows the OnlineGDB interface with a C program named 'main.c' in the editor. The code prints a triangle of numbers with increasing rows and consistent spacing. The output window shows the printed triangle:

```

1
2   3
4   5   6
...Program finished with exit code 0
Press ENTER to exit console.

```

b. 1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

The screenshot shows the OnlineGDB interface with a C program named 'main.c' in the editor. The code prints a symmetric Pascal's triangle of 5 rows. The output window shows the triangle:

```

1
1   1
1   2   1
1   3   3   1
1   4   6   4   1
...Program finished with exit code 0
Press ENTER to exit console.

```

4. 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.

The screenshot shows the OnlineGDB interface. The code in the editor is:

```

1 #include <stdio.h>
2 int main() {
3     float population = 100000;
4     for (int year = 1; year <= 10; year++) {
5         population += population * 0.10;
6         printf("Year %d: Population = %.0f\n", year, population);
7     }
8     return 0;
9 }
10

```

The output window shows the population for each year from 1 to 10:

```

Year 1: Population = 110000
Year 2: Population = 121000
Year 3: Population = 133100
Year 4: Population = 146410
Year 5: Population = 161051
Year 6: Population = 177156
Year 7: Population = 194872
Year 8: Population = 214359
Year 9: Population = 235795
Year 10: Population = 259375

```

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. Example of Ramanujan number: $1729 = 1^3 + 12^3$ and $1729 = 9^3 + 10^3$. for a number L=20(that is limit)

The screenshot shows the OnlineGDB interface. The code in the editor is:

```

1 #include <stdio.h>
2 int main() {
3     int L = 20;
4     for (int a = 1; a <= L; a++) {
5         for (int b = a + 1; b <= L; b++) {
6             int sum1 = a*a*a + b*b*b;
7
8             for (int c = a + 1; c <= L; c++) {
9                 for (int d = c + 1; d <= L; d++) {
10                    int sum2 = c*c*c + d*d*d;
11
12                    if (sum1 == sum2 && (a != c && b != d)) {
13                        printf("Ramanujan Number: %d = %d^3 + %d^3 = %d^3 + %d^3\n",
14                               sum1, a, b, c, d);
15                    }
16                }
17            }
18        }
19    }
20    return 0;
21

```

The output window shows the Ramanujan numbers found:

```

Ramanujan Number: 1729 = 1^3 + 12^3 = 9^3 + 10^3
Ramanujan Number: 4104 = 2^3 + 16^3 = 9^3 + 15^3

...Program finished with exit code 0
Press ENTER to exit console.

```

EXPERIMENT 4

- 1) <https://onlinegdb.com/xBUVJyJTbp>
- 2) <https://onlinegdb.com/3wvQzVgZX>

Experiment 5

- 1) <https://onlinegdb.com/nMUYTKpmE>
- 2) https://onlinegdb.com/Gd6YkpEFI_chatgpt
- 3) https://onlinegdb.com/l6Gi-0y6N_chatgpt
- 4) https://onlinegdb.com/_GajliAub_chatgpt

Experiment 7.1

- 1) https://onlinegdb.com/_lbtdf1yt
- 3) <https://onlinegdb.com/RAa4CyZe1X>
- 4) https://onlinegdb.com/w_StGlCew