

DOMAIN WINTER WINNING CAMP 2024

ASSIGNMENT DAY 1

1) Write a program to calculate the area of different shapes using function overloading. Implement overloaded functions to compute the area of a circle, a rectangle, and a triangle.

Input Format

The program should accept:

1. Radius of the circle for the first function.
2. Length and breadth of the rectangle for the second function.
3. Base and height of the triangle for the third function.

Constraints

$1 \leq \text{radius, length, breadth, base, height} \leq 10^3$

Use 3.14159 for the value of π .

Output Format

Print the computed area of each shape in a new line.

Test Cases:

Example 1

Input:

Radius = 5

Length = 4, breadth = 6

Base = 3, height = 7

Output:

78.53975

24

10.5

Solution

```
#include <iostream>
```

```
#include <cmath>
```

```
using namespace std;
```

```
double calculateCircle(double radius) {  
    return M_PI * radius * radius;  
}
```

```
double calculateRect(double length, double width) {  
    return length * width;  
}
```

```
double calculateTri(double base, double height) {  
    return 0.5 * base * height;  
}
```

```
int main() {  
    cout << "Area of Circle: " << calculateCircle(5.0) << endl;  
    cout << "Area of Rectangle: " << calculateRect(4.0, 6.0) << endl;  
    cout << "Area of Triangle: " << calculateTri(3.0, 4.0) << endl;  
    return 0;  
}
```

OUTPUT:

```
Area of Circle: 78.5398
Area of Rectangle: 24
Area of Triangle: 6

=== Code Execution Successful ===
```

2) Print the multiplication table of a given number n . A multiplication table for a number n is a list of products of n with integers from 1 to 10. For example, the multiplication table for 3 is:
 $3 \times 1 = 3, 3 \times 2 = 6, \dots, 3 \times 10 = 30$.

Task

Given an integer n , print the multiplication table of n from $1 \times n$ to $10 \times n$.

Input Format

One integer n .

Constraints

- $1 \leq n \leq 100$

Output Format

For each integer i from 1 to 10, print the product $n \times i$ in the format:
 $n \times i = \text{product}$.

SOLUTION:

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n=5;
    for(int i=1;i<11;i++)
    {
```

```

        cout<<n<<"*"<<i<<"="<<n*i<<endl;
    }
    return 0;
}

```

OUTPUT:

```

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

```

3) Calculate the sum of all odd numbers from 1 to n. An odd number is an integer that is not divisible by 2. The sum of odd numbers, iterate through all the numbers from 1 to n, check if each number is odd, and accumulate the sum.

Task

Given an integer n, print the sum of all odd numbers from 1 to n.

Input Format

One integer n, the upper limit of the range.

Constraints

- $1 \leq n \leq 10^4$

Output Format

Print the sum of all odd numbers from 1 to n.

SOLUTION:

```

#include <iostream>
using namespace std;

```

```

int main()
{
    int sum=0,n=20;
    for(int i=0;i<=20;i++)
    {
        if (i%2==0)
        {
            sum+=i;
        }

    }
    cout<<sum;
    return 0;
}

```

OUTPUT:

110

=== Code Execution Successful ===

4)Count the total number of digits in a given number n. The number can be a positive integer. For example, for the number 12345, the count of digits is 5. For a number like 900000, the count of digits is 6.

Given an integer n, your task is to determine how many digits are present in n. This task will help you practice working with loops, number manipulation, and conditional logic.

Task

Given an integer n, print the total number of digits in n.

Input Format

One integer n.

Constraints

- $1 \leq n \leq 10^9$

Output Format

Print the number of digits in n.

SOLUTION:

```
#include <iostream>
using namespace std;
```

```
int main()
{
    int n=4352433;
    int count=0;
    while(n>10)
    {
        n/=10;
        count++;
    }
    cout<<count+1;
    return 0;
}
```

OUTPUT:

7

=== Code Execution Successful ===

5) Check if a given number n is a prime number. A prime number is a natural number greater than 1 that has no positive divisors other than 1 and itself.

To determine if a number is prime, iterate from 2 to \sqrt{n} and check if n is divisible by any number in this range. If it is divisible, it is not a prime number; otherwise, it is a prime.

Task

Given an integer n , print "Prime" if the number is prime, or "Not Prime" if it is not.

Input Format

One integer n .

Constraints

- $2 \leq n \leq 10^5$

Output Format

Print "Prime" if n is prime, otherwise print "Not Prime".

SOLUTION:

```
#include <iostream>
```

```
#include <math.h>
```

```
using namespace std;
```

```
bool prime(int n)
```

```
{
```

```
    int limit=sqrt(n);
```

```
    for(int i=1;i<=limit;i++) if(n%i==0) return false;
```

```
    return true;
```

```
}
```

```
int main()
```

```
{
```

```
    int n=79;
```

```
if(prime(n)) cout<<"Prime";  
else cout<<"Not Prime";  
return 0;  
}
```

OUTPUT:

```
Not Prime
```

```
=== Code Execution Successful ===
```