DAY 1

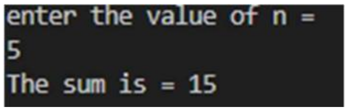
Student Name: Akarsh **UID: 22BCS15248**
Branch: BE-CSE **Section/Group: 620 - A**
Date of Performance:19/12/24

Problem 1

1. **Aim: Sum of Natural Numbers up to N**
2. **Problem Statement:** Calculate the sum of all natural numbers from 1 to n, where n is a positive integer. Use the formula: Sum=n×(n+1)/2 .
   Take n as input and output the sum of natural numbers from 1 to n .
3. **Task:** Given an integer n, print the sum of all natural numbers from 1 to n.
4. **Code:**

```
#include<iostream>
using namespace std;
int main()
{
   int n,sum=0;
   cout<<"enter the value of n = "<<endl;
   cin>>n;
   sum = n*(n+1)/2;
   cout<<"The sum is = "<<sum;
   return 0;
}
```

5. **Output:**



```
enter the value of n =
5
The sum is = 15
```

Problem 2

1. **Aim: Print Multiplication Table of a Number**

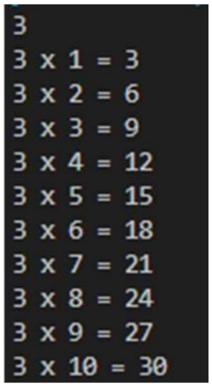
2. **Problem Statement:** Print the multiplication table of a given number n. A multiplication table for a number n is a list of products of n with integers from 1 to 10. For example, the multiplication table for 3 is:
   3×1=3,3×2=6,…,3×10=30.

3. **Task:** Given an integer n, print the multiplication table of n from 1×n to 10×n.
4. **Code:**

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

int main()
{
    int n;
    cin>>n;
    for (int i = 1; i <= 10; i++)
    {
      cout<<n<<" x "<<i<<" = "<<n*i<<endl;
    }
    return 0;
```

5. **Output:**



```
3
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30
```

Problem 3

1. **Aim: Count Digits in a Number**

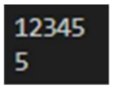
2. **Problem Statement:** Count the total number of digits in a given number n. The number can be a positive integer. For example, for the number 12345, the count of digits is 5. For a number like 900000, the count of digits is 6.
   Given an integer n, your task is to determine how many digits are present in n. This task will help you practice working with loops, number manipulation, and conditional logic.
3. **Task:** Given an integer n, print the multiplication table of n from 1×n to 10×n.
4. **Code:**

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

int main()
{
   int n, count = 0 ;
   cin>> n;
   if (n<0)
```

```
    {
        n = 0-n;
    }
    while(n>0)
    {
        n=n/10;
        count++;
    }
    cout<<count;
    return 0;
}
```

5. **Output:**



```
12345
5
```

Problem 4

1. **Aim: Check if a Number is Prime**

2. **Problem Statement:** 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 √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

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

4. Code:

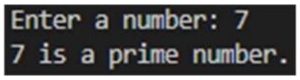
```
 #include<iostream>
 using namespace std;

bool isPrime(int number) {
    if (number <= 1) {
        return false;
    }
    for (int i = 2; i * i <= number; i++) {
        if (number % i == 0) {
            return false;
        }
    }
    return true;
}
int main() {
    int num;
    cout << "Enter a number: ";
    cin >> num;

    if (isPrime(num)) {
        cout << num << " is a prime number." << endl;
    } else {
        cout << num << " is not a prime number." << endl;
    }

    return 0;
}
```

5. }Output:



```
Enter a number: 7
7 is a prime number.
```

Problem 5

1. **Aim: Print Odd Numbers up to N**

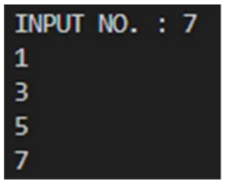
2. **Problem Statement:** Print all odd numbers between 1 and n, inclusive. Odd numbers are integers that are not divisible by 2. These numbers should be printed in ascending order, separated by spaces.
   This problem is a simple introduction to loops and conditional checks. The goal is to use a loop to iterate over the numbers and check if they are odd using the condition i%2≠0.

3. **Task:** Given an integer n, print all odd numbers from 1 to n, inclusive.
4. **Code:**

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

int main()
{
   int n;
   cout<<"INPUT NO. : ";
   cin>>n;
   for(int i = 1; i<=n; i =i+2)
   {
      cout<<i<<"\n";
   }
   return 0;
}
```

5. **Output:**



```
INPUT NO. : 7
1
3
5
7
```

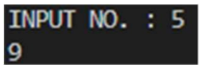
Problem 6

1. **Aim: Sum of Odd Numbers up to N**
2. **Problem Statement:** 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.
3. **Task:** Given an integer n, print the sum of all odd numbers from 1 to n.
4. **Code:**

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

int main()
{
   int n, SUM;
   cout<<"INPUT NO. : ";
   cin>>n;
   for(int i = 1; i<=n; i =i+2)
   {
      SUM += i;
   }
   cout<<SUM;
   return 0;
}
```

5. **Output:**



```
INPUT NO. : 5
9
```

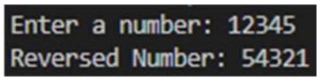
Problem 7

1. **Aim: Reverse a Number**
2. **Problem Statement:** Reverse the digits of a given number n. For example, if the input number is 12345, the output should be 54321. The task involves using loops and modulus operators to extract the digits and construct the reversed number
3. **Task:** Given an integer n, print the number with its digits in reverse order.
4. **Code:**

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

int main() {
    int num, rev_num = 0;
    cout << "Enter a number: ";
    cin >> num;
    while (num != 0) {
        rev_num = rev_num * 10 + num % 10;
        num /= 10;
    }
    cout << "Reversed Number: " << rev_num << endl;
    return 0;
}
```

5. **Output**



```
Enter a number: 12345
Reversed Number: 54321
```