Department of Computer Science and Engineering

Name:- Vikash Ranjan Kumar UID:- 22BCS11322 Section:- 22KPIT-901-B

DAY-1

Q.1. Calculate the sum of all natural numbers from 1 to n, where n is a positive integer.

Program Code:-

```
#include <iostream>

using namespace std;

int main() {

    int n;

    cin >> n;

    int sum = n * (n + 1) / 2;

    cout << sum << endl;

    return 0;

}
```

Output:-

Sample Input

5

Your Output

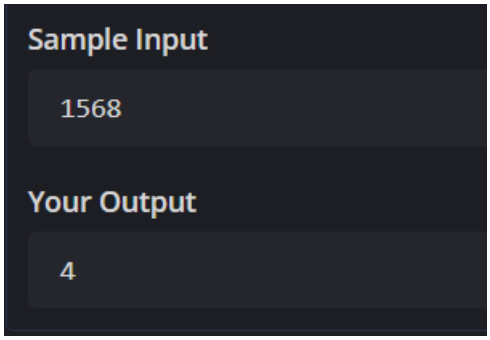
15

Q.2. 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.

Program Code:-

```
#include <iostream>

using namespace std;


int main() {

    int n, count = 0;

    cin >> n;

    while (n != 0) {

        n /= 10;

        count++;

    }

    cout << count << endl;

    return 0;

}
```

Output:-



Q.3. 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.

Program Code:-

```
#include <iostream>

using namespace std;

const double PI = 3.14159;

double calculateArea(double radius) {

    return PI * radius * radius;

}

int calculateArea(int length, int breadth) {

    return length * breadth;

}

double calculateArea(double base, double height) {

    return 0.5 * base * height;
```

```
}
int main() {
    double radius, base, height;
    int length, breadth;
    cin >> radius;
    cin >> length >> breadth;
    cin >> base >> height;
    cout << calculateArea(radius) << endl;
    cout << calculateArea(length, breadth) << endl;
    cout << calculateArea(base, height) << endl;
    return 0;
}
```

Output:-

Sample Input

```
5
10
5
12
2
```

Your Output

```
78.5397
50
12
```

Q.4. Write a program to demonstrate runtime polymorphism in C++ using a base class Shape and derived classes Circle, Rectangle, and Triangle. The program should use virtual functions to calculate and print the area of each shape based on user input.

Program Code:-

```
 #include <iostream>

#include <cmath>

using namespace std;

class Shape {

public:

    virtual void calculateArea() = 0; // Pure virtual function

    virtual ~Shape() {}              // Virtual destructor

};

class Circle : public Shape {

private:

    double radius;

public:

    Circle(double r) : radius(r) {}

    void calculateArea() override {

        double area = M_PI * radius * radius;

        cout << "Area of Circle: " << area << endl;
```

```
    }
};
class Rectangle : public Shape {
private:
    double length, width;
public:
    Rectangle(double l, double w) : length(l), width(w) {}
    void calculateArea() override {
        double area = length * width;
        cout << "Area of Rectangle: " << area << endl;
    }
};
class Triangle : public Shape {
private:
    double base, height;
public:
    Triangle(double b, double h) : base(b), height(h) {}
    void calculateArea() override {
        double area = 0.5 * base * height;
```

```
        cout << "Area of Triangle: " << area << endl;
    }
};

int main() {
    Shape* shape = nullptr;
    int choice;
    cout << "Choose a shape to calculate the area:\n";
    cout << "1. Circle\n";
    cout << "2. Rectangle\n";
    cout << "3. Triangle\n";
    cout << "Enter your choice: ";
    cin >> choice;
    switch (choice) {
        case 1: {
            double radius;
            cout << "Enter the radius of the circle: ";
            cin >> radius;
            shape = new Circle(radius);
```

```
    break;
}
case 2: {
    double length, width;
    cout << "Enter the length and width of the rectangle: ";
    cin >> length >> width;
    shape = new Rectangle(length, width);
    break;
}
case 3: {
    double base, height;
    cout << "Enter the base and height of the triangle: ";
    cin >> base >> height;
    shape = new Triangle(base, height);
    break;
}
default:
     cout << "Invalid choice!" << endl;
    return 1;
```

```
    }
    if (shape) {
        shape->calculateArea();
        delete shape; // Free allocated memory
    }
    return 0;
}
```

Output:-

Sample Input

```
1
5
```

Your Output

```
Choose a shape to calculate the area:
1. Circle
2. Rectangle
3. Triangle
Enter your choice: Enter the radius of the circle: Area of Circle: 78.5398
```

Q.5. Design a C++ program to simulate a banking system using polymorphism. Create a base class Account with a virtual method calculateInterest(). Use the derived classes SavingsAccount and CurrentAccount to implement specific interest calculation logic:

- **SavingsAccount**: Interest = Balance × Rate × Time.
- **CurrentAccount**: No interest, but includes a maintenance fee deduction.

```
#include <iostream>
using namespace std;
class Account {
public:
virtual void calculateInterest() = 0;
};
class SavingsAccount : public Account {
int balance, rate, time;
public:
SavingsAccount(int b, int r, int t) : balance(b), rate(r), time(t) {}
void calculateInterest() override {
cout << "Savings Account Interest: " << (balance * rate * time) / 100 << endl;
}
};
class CurrentAccount : public Account {
int balance, fee;
public:
CurrentAccount(int b, int f) : balance(b), fee(f) {}
void calculateInterest() override {
cout << "Balance after fee deduction: " << balance - fee << endl;
}
};
int main() {
```

```
int type;
cin >> type;
if (type == 1) {
int balance, rate, time;
cin >> balance >> rate >> time;
SavingsAccount sa(balance, rate, time);
sa.calculateInterest();
} else if (type == 2) {
int balance, fee;
cin >> balance >> fee;
CurrentAccount ca(balance, fee);
ca.calculateInterest();
} else {
cout << "Invalid account type." << endl;
}
return 0;
}
```

Output:-

Sample Input

```
1
5000
4
2
```

Your Output

```
Savings Account Interest: 400
```