DOMAIN WINTER CAMP

Name: - Priti
UID:- 22BCS12729
Section:- 22KPIT-901-B

DAY-1

Q.1. Find the largest digit in each number n. For example, for the number 2734, the largest digit is 7. You need to extract each digit from the number and determine the largest one. The task will involve using loops and modulus operations to isolate the digits.

Program Code: -

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

int main(){

    int n;
    cin>>n;
    int largest=0;
    while(n>0){
        if(n % 10 > largest){
            largest=n%10;
        }
        n=n/10;
    }
    cout<<largest;
    return 0;
}
```

Output:-

```
8546351
8
PS D:\Coding\dsa>
```

Q.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:
3×1=3,3×2=6,…,3×10=30.

Program Code:-

```cpp
#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;
}
```

Output:-

```
7
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70
PS D:\Coding\dsa>
```

Q.3. Write a program that demonstrates encapsulation by creating a class Employee. The class should have private attributes to store:
Employee ID.
Employee Name.
Employee Salary.
Provide public methods to set and get these attributes, and a method to display all details of the employee.
Program Code:-

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

class Employee {
private:
    int employeeID;
    string employeeName;
    double employeeSalary;

public:
    void setEmployeeID(int id) {
        employeeID = id;
    }
    int getEmployeeID() {
        return employeeID;
    }
    void setEmployeeName(const string &name) {
        employeeName = name;
    }
    string getEmployeeName() {
        return employeeName;
    }
    void setEmployeeSalary(double salary) {
        employeeSalary = (salary >= 0) ? salary : 0;
    }
    double getEmployeeSalary() {
        return employeeSalary;
    }
    void displayEmployeeDetails() {
        cout << "Employee Details:\n";
        cout << "ID: " << employeeID << "\n";
        cout << "Name: " << employeeName << "\n";
        cout << "Salary: " << employeeSalary << "\n";
    }
};
```

```
int main() {
    Employee emp;
    emp.setEmployeeID(101);
    emp.setEmployeeName("John Doe");
    emp.setEmployeeSalary(50000.50);
    emp.displayEmployeeDetails();
    cout << "\nAccessing Individual Attributes:\n";
    cout << "Employee ID: " << emp.getEmployeeID() << "\n";
    cout << "Employee Name: " << emp.getEmployeeName() << "\n";
    cout << "Employee Salary: " << emp.getEmployeeSalary() << "\n";
    return 0;
}
```

Output:-

```
Employee Details:
ID: 101
Name: John Doe
Salary: 50000.5

Accessing Individual Attributes:
Employee ID: 101
Employee Name: John Doe
Employee Salary: 50000.5
PS D:\Coding\dsa>
```

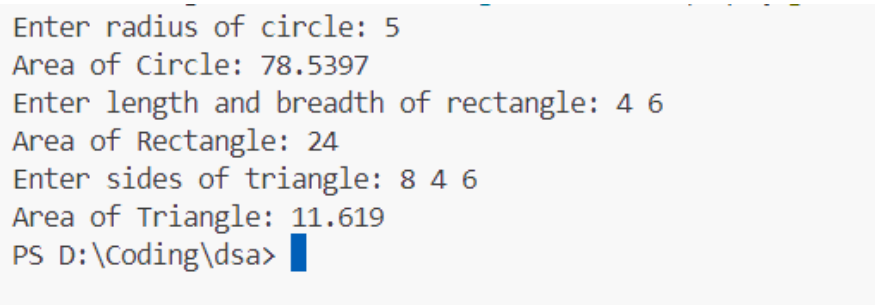
Q.4. Write a program to calculate the area of different shapes using function overloading. Implement overloaded functions to compute the area of a circle, a rectangle, and a triangle.

Program Code:-

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


double calculateArea(double radius) {
   return 3.14159 * radius * radius;
}

double calculateArea(double length, double breadth) {
   return length * breadth;
}

double calculateArea(double a, double b, double c) {
   double s = (a + b + c) / 2.0;
   double area = s * (s - a) * (s - b) * (s - c);
   double result = 1;
   for (int i = 0; i < 10; ++i) {

      result = 0.5 * (result + area / result);

   }
```

```
    return result;

}

int main() {

    double radius, length, breadth, a, b, c;
    cout << "Enter radius of circle: ";
    cin >> radius;

    cout << "Area of Circle: " << calculateArea(radius) << endl;
    cout << "Enter length and breadth of rectangle: ";
    cin >> length >> breadth;

    cout << "Area of Rectangle: " << calculateArea(length, breadth) << endl;
    cout << "Enter sides of triangle: ";
    cin >> a >> b >> c;

    cout << "Area of Triangle: " << calculateArea(a, b, c) << endl;
    return 0;
}
```

Output:-



Q.5. Design a C++ program using polymorphism to calculate the area of different shapes:
A Rectangle (Area = Length × Breadth).
A Circle (Area = $\pi \times Radius^2$).
A Triangle (Area = ½ × Base × Height).
Create a base class Shape with a pure virtual function getArea(). Use derived classes Rectangle, Circle, and Triangle to override this function.

Program Code:-

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

class Shape {
public:
    virtual double getArea() const = 0;
    virtual ~Shape() {}
};

class Rectangle : public Shape {
    double length, breadth;
public:
    Rectangle(double l, double b) : length(l), breadth(b) {}
    double getArea() const override { return length * breadth; }
};

class Circle : public Shape {
    double radius;
public:
    Circle(double r) : radius(r) {}
    double getArea() const override { return M_PI * radius * radius; }
};

class Triangle : public Shape {
    double base, height;
public:
    Triangle(double b, double h) : base(b), height(h) {}
    double getArea() const override { return 0.5 * base * height; }
};

int main() {
    Shape* shapes[] = {
        new Rectangle(10, 5),
        new Circle(7),
        new Triangle(6, 4)
    };
    for (Shape* shape : shapes) {
        cout << "Area: " << shape->getArea() << "\n";
        delete shape;}
    return 0;
}
```

Output:-

```
Area: 50
Area: 153.938
Area: 12
PS D:\Coding\dsa>
```