Lab 07 – Function Overloading and Recursion

Task 01:

Write a recursive function for the calculation of power.

Code:

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

int power(int base, int exp){
    if(exp == 0) return 1;
    return base*power(base, exp - 1);
}

int main(){
    int base = 0, exp = 0;
    cout << "Enter base: ";
    cin >> base;
    cout << "Enter power: ";
    cin >> exp;

    cout << base << " to the power " << exp << " is: " << power(base, exp) <<
endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task01.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter base: 3
Enter power: 4

3 to the power 4 is: 81
```

Task 02:

Number Comparison Utility

You are designing a program to assist users in analyzing a set of numbers. The application should allow users to enter between two to four numbers and then determine the smallest and largest numbers among them. Additionally, users should be able to compare different combinations of numbers, such as comparing two, three, or four numbers simultaneously. This flexibility enables users to make quick comparisons and analyze data effectively

Code:

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

int findSmall(int a, int b){
    if(a <= b) return a;
    else return b;
}

int findSmall(int a, int b, int c){
    if(a <= b && a <= c) return a;
    else if(b <= a && b <= c) return b;
    else return c;
}

int findSmall(int a, int b, int c, int d){
    if(a <= b && a <= c && a <= d) return a;
    else if(b <= a && b <= c && b <= d) return b;
    else if(c <= a && c <= b && c <= d) return c;
    else return d;
}

int findLarg(int a, int b){
    if(a >= b) return a;
    else return b;
}

int findLarg(int a, int b, int c){
    if(a >= b && a >= c) return a;
    else if(b >= a && b >= c) return b;
    else return c;
}

int findLarg(int a, int b, int c, int d){
    if(a >= b && a >= c && a >= d) return a;
```

```
    else if(b >= a && b >= c && b >= d) return b;
    else if(c >= a && c >= b && c >= d) return c;
    else return d;
}

int main(){
    int a[4];

    cout << "Enter 4 numbers: " << endl;
    for(int i = 0; i < 4; i++) cin >> a[i];

    cout << endl;
    cout << "**********************" << endl;
    cout << "Smallest among first two is: " << findSmall(a[0], a[1]) << endl;
    cout << "Largest among first two is: " << findLarg(a[0], a[1]) << endl;

    cout << "**********************" << endl;
    cout << "Smallest among first three is: " << findSmall(a[0], a[1], a[2]) <<
endl;
    cout << "Largest among first three is: " << findLarg(a[0],a[1], a[2]) <<
endl;

    cout << "**********************" << endl;
    cout << "Smallest among all is: " << findSmall(a[0], a[1], a[2], a[3]) <<
endl;
    cout << "Largest among all is: " << findLarg(a[0], a[1], a[2], a[3]) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task02.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter 4 numbers:
56
34
66
3

**********************
Smallest among first two is: 34
Largest among first two is: 56
**********************
Smallest among first three is: 34
Largest among first three is: 66
**********************
Smallest among all is: 3
Largest among all is: 66
```

Task 03:

Recursive Fibonacci number Calculation

Imagine you are developing a program to calculate Fibonacci numbers for a mathematics class. Your task is to create a C++ program that allows students to input a positive number and calculates the corresponding Fibonacci number using a recursive function. Here's how the program works:
The program prompts the user to enter a positive number.

- The user inputs a positive number, and the program stores it in a variable called N.
- The program calls a recursive function called fib() to calculate the Fibonacci number for the input value of N.

The fib() function calculates the Fibonacci number using the following rules:

- For n = 0, the Fibonacci number is 0.
- For n = 1, the Fibonacci number is 1.
- For n > 1, the Fibonacci number is calculated as the sum of the Fibonacci numbers of the previous:

two terms: fib(n) = fib(n-1) + fib(n-2)

Code:

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

int fib(int n){
    if(n == 0 || n == 1) return n;
    return fib(n-1) + fib(n-2);
}

int main(){
   int n = 0;
   cout << "Enter  a positive number: ";
   cin >> n;

   cout << "Fibonacci no. for " << n << " is: " << fib(n) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task03.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter a positive number: 5

Fibonacci no. for 5 is: 5
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Enter a positive number: 7

Fibonacci no. for 7 is: 13
```

Task 04:

Growth Simulator

Write a C++ program that uses recursion to calculate the population size of rabbits after a specified number of months.
The program should prompt the user to enter the initial number of rabbit pairs and the number of months for which they want to simulate the growth.
Implement a recursive function to calculate the population size based on the following rules:

- Each pair of rabbits reproduces every month.
- Each pair produces exactly one new pair of offspring.
- Rabbits mature in one month and start reproducing.
- Assume no rabbits die during the simulation.
- After computing the population size, display the result to the user.
- Ensure that the program handles valid inputs and gracefully handles errors.

Code:

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

int growthSim(int month){
   if(month == 1 || month == 2) return 1;
   return growthSim(month - 1) + growthSim(month - 2);
}

int main(){
   int initialPairs = 0, month = 0;
   cout << "Welcome to the Rabbit popullation Growth Simulator" << endl << endl;

   cout << "Enter the initial number of rabbit pairs: ";
   cin >> initialPairs;
   cout << "Enter the number of months for simulation: ";
   cin >> month;

   cout << "After " << month << " months, Rabbits population will be: " <<
initialPairs*growthSim(month) << endl;
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task04.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Welcome to the Rabbit popullation Growth Simulator

Enter the initial number of rabbit pairs: 3
Enter the number of months for simulation: 4

After 4 months, Rabbits population will be: 9
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe
Welcome to the Rabbit popullation Growth Simulator

Enter the initial number of rabbit pairs: 2
Enter the number of months for simulation: 6

After 6 months, Rabbits population will be: 16
```

Task 05:

Finding the Maximum

Imagine you are developing a program to assist engineers in analyzing sensor data from three different sources. The program needs to identify the highest recorded values from each sensor to determine potential anomalies in the data. For this purpose, you decide to create a function that finds the maximum of three floating-point numbers, as well as find the maximum of two integer point numbers.

Code:

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

float findMax(float a, float b, float c){
    if(a >= b && a >= c) return a;
    else if(b >= a && b >= c) return b;
    else return c;
}

int findMax(int a, int b){
    if(a >= b) return a;
    else return b;
}

int main(){
    int choice = 0;
    float fVal[3];
    int intVal[2];

    do{
        cout << "1. Compare 3 float values." << endl;
        cout << "2. Compare 2 integer values." << endl;
        cout << "3. Exit."  << endl;
        cout << "Enter choice: ";
        cin >> choice;

        if(choice == 1){
            cout << "Enter values: ";
            cin >> fVal[0] >> fVal[1] >> fVal[2];
            cout << "Largest among them is: " << findMax(fVal[0], fVal[1],
fVal[2]) << endl;
        }
        else if(choice == 2){
            cout << "Enter value: ";
```

```
            cin >> intVal[0] >> intVal[1];
            cout << "Largest among them is: " << findMax(intVal[0], intVal[1]) <<
endl;
        }
        else if(choice == 3) cout << "Closong program..." << endl;
        else cout << "Invalid choice!" << endl;

    }while(choice != 3);
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task05.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.
Enter choice: 1

Enter values: 45.5
45.6
32

Largest among them is: 45

2. Compare 2 integer values.
3. Exit.
Enter choice: 3

Closong program...
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task05.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.
Enter choice: 1

Enter values: 45.5
45.6
34

Largest among them is: 45.6
```

```
Largest among them is: 45.6

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.
Enter choice: 2

Enter value: 3
78

Largest among them is: 78

1. Compare 3 float values.
2. Compare 2 integer values.
3. Exit.
Enter choice: 3

Closong program...
```

Task 06:

Calculating Area with Different Shapes

Picture yourself developing a geometry calculator program using function overloading. This application allows users to calculate the area of various shapes such as circles, rectangles, and triangles. Users can input the required dimensions for each shape, and the program intelligently selects the appropriate function to calculate and display the area.

Code:

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

float cirArea(float r){
    return (3.14*(r*r));
}

float recArea(float l, float w){
    return l*w;
}

float triArea(float b, float h){
    return (1/2)*b*h;
}

int main(){
    int choice = 0;

    do{
        cout << endl;
        cout << "1. Calculate area of circle." << endl;
        cout << "2. Calculate area of rectangle." << endl;
        cout << "3. Calculate area of triangle." << endl;
        cout << "4. Exit." << endl;
        cout << "Enter choiec: ";
        cin >> choice;

        switch(choice){
            case 1:{
                cout << endl;
                float radius = 0.00;
                cout << "Enter radius: ";
                cin >> radius;
                cout << "Area of circle is: " << cirArea(radius) << endl;
                break;
```

```
            }

            case 2:{
                cout << endl;
                float length = 0.00, width = 0.00;
                cout << "Enter length: ";
                cin >> length;
                cout << "Enter width: ";
                cin >> width;
                cout << "Area is: " << recArea(length, width) << endl;
                break;
            }

            case 3:{
                cout << endl;
                float base = 0.00, height = 0.00;
                cout << "Enter base: ";
                cin >> base;
                cout << "Enter height: ";
                cin >> height;
                cout << "Area is: " << triArea(base, height) << endl;
                break;
            }

            case 4: cout << "\nExiting program." << endl;
            break;

            default: cout << "\nInvalid choice!" << endl;
            break;
        }
    }while(choice != 4);
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task06.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe

1. Calculate area of circle.
2. Calculate area of rectangle.
3. Calculate area of triangle.
4. Exit.
Enter choiec: 1

Enter radius: 45
Area of circle is: 6358.5

1. Calculate area of circle.
2. Calculate area of rectangle.
3. Calculate area of triangle.
4. Exit.
Enter choiec: 3

Enter base: 5
Enter height: 6
Area is: 0

1. Calculate area of circle.
2. Calculate area of rectangle.
3. Calculate area of triangle.
4. Exit.
Enter choiec: 4

Exiting program.
```

Task 07:

Fuel Efficiency Calculator

You're building a fuel efficiency calculator for a vehicle management system. The goal is to let users check fuel efficiency for cars, trucks, and airplanes using function overloading.

What You Have to Do:
Use function overloading to create separate fuel efficiency functions for: Cars, Trucks & Airplanes

Parameters by Vehicle Type:

Cars:
- int: engine displacement
- double: fuel consumption rate
- double: vehicle weight

Trucks:
- int: cargo weight
- float: engine power
- double: aerodynamic drag coefficient

Airplanes:
- string: aircraft type
- double: cruising altitude
- float: airspeed

User Interaction:
- Ask the user which vehicle they're calculating for
- Get inputs for the required parameters
- Calculate and display the fuel efficiency with correct units

Code:

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

float calEfficiency(int disp, double consump, double weight){
    return ((disp / 1000.0) * (100.0 / consump) / (weight / 1000.0));
}
float calEfficiency(int cargoWeight, float pow, double aeroDynamicDrag){
    return (pow * 0.8) / ((cargoWeight / 1000.0) * aeroDynamicDrag + 1.0);
}
float calEfficiency(string type, double altitude, float speed){
    return (speed * 0.1) / (altitude / 10000.0 + 1.0);
}
```

```
int main(){
    int choice = 0;
    do{
        cout << endl;
        cout << "1. Calculate efficiency of car." << endl;
        cout << "2. Calculate efficiency of truck." << endl;
        cout << "3. Calculate efficiency of Airplane." << endl;
        cout << "4. Exit." << endl;
        cout << "Enter choice: ";
        cin >> choice;
        if(choice == 1){
            int displacement = 0;
            double fuelCon = 0.00, weight = 0.00;
            cout << "\nEnter displacement: ";
            cin >> displacement;
            cout << "Enter fuel consumption: ";
            cin >> fuelCon;
            cout << "Enter vehcile weight: ";
            cin >> weight;
            cout << "\nFuel Efficiency is: " << calEfficiency(displacement,
fuelCon, weight) << endl;
        }
        else if(choice == 2){
            int cargoWeight = 0;
            float enginePow = 0.00;
            double dragCoefficient = 0.00;
            cout << "\nEnter Cargo weight: ";
            cin >> cargoWeight;
            cout << "Enter engine power: ";
            cin >> enginePow;
            cout << "Enter aerodynamic drag coefficient: ";
            cin >> dragCoefficient;
            cout << "\nFuel Efficiency is: " << calEfficiency(cargoWeight,
enginePow, dragCoefficient) << endl;
        }
        else if(choice == 3){
            string type;
            double altitude = 0.00;
            float speed = 0.00;

            cin.ignore();
            cout << "\nEnter aircraft type: ";
            getline(cin, type);
            cout << "Enter cruising altitude: ";
            cin >> altitude;
```

```
            cout << "Enter airspeed: ";
            cin >> speed;

            cout << "\nFuel Efficiency is: " << calEfficiency(type, altitude,
speed) << endl;
        }

        else if(choice == 4) cout << "\nExting program." << endl;
        else cout << "\nInvalid choice!" << endl;
    }while(choice != 4);
}
```

Output:

```
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> g++ task07.cpp
PS D:\Hasan\C++\00. University\Lab 07 - Recursion> ./a.exe

1. Calculate efficiency of car.
2. Calculate efficiency of truck.
3. Calculate efficiency of Airplane.
4. Exit.
Enter choice: 2

Enter Cargo weight: 45
Enter engine power: 887
Enter aerodynamic drag coefficient: 54

Fuel Efficiency is: 206.88

1. Calculate efficiency of car.
2. Calculate efficiency of truck.
3. Calculate efficiency of Airplane.
4. Exit.
Enter choice: 3

Enter aircraft type: Cargo
Enter cruising altitude: 6
Enter airspeed: 89

Fuel Efficiency is: 8.89466

1. Calculate efficiency of car.
2. Calculate efficiency of truck.
3. Calculate efficiency of Airplane.
4. Exit.
Enter choice: 4

Exting program.
```