Naafiul Hossain
ESE224
115107623
Tuesday 10-12:50am

Problem 1:

Main.cpp

```cpp
//Naafiul Hossain
// sbu id: 115107623
#include <iostream>
#include <fstream>

using namespace std;
void swap(double&, double&);
void fakeSwap(double, double);

int main() {
    double a, b;
    ifstream datain;
    datain.open("input.dat");
    if (datain.fail()) {
        cerr << "File canot open";
        exit(1);
    }
    ofstream dataout;
    dataout.open("output.dat");
    if (dataout.fail()) {
        cerr << "File canot open";
        exit(1);
    }
```

```
}
void swap(double& a, double& b) {
    double temp;
    temp = a;
    a = b;
    b = temp;
}
void fakeSwap(double a, double b) {
    double temp = a;
    a = b;
    b = temp;
}
```

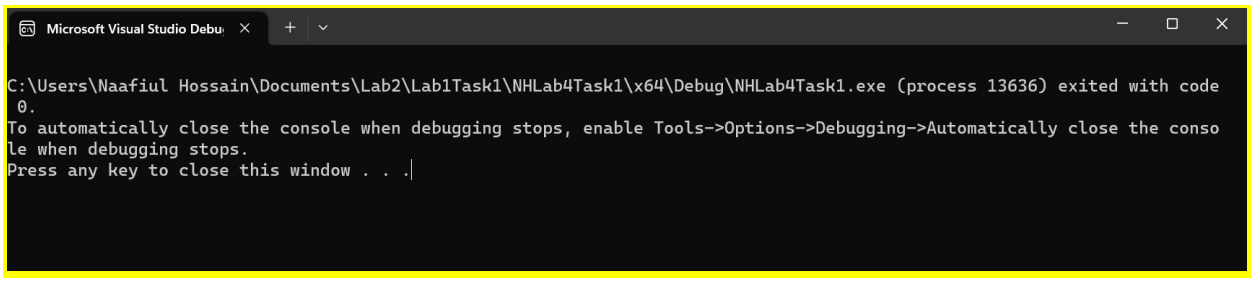
Input dat:

```
3 4
3 4
```

Output.dat:

```
3 4
4 3
3 4
4 3
4 3
3 4
```

Screenshot of the running program:



Problem 2

main.h

```cpp
//Naafiul Hossain
//SBU ID: 115107623

#include <iostream>
#include <random>

using namespace std;

int main() {
    int N;

    // Ask the user to input the value of N
    cout << "Enter the number of random double numbers to generate (N): ";
    cin >> N;

    random_device rd;
    mt19937 gen(rd());
    uniform_real_distribution<double> dis(1.0, 2.0);

    double sum = 0.0;
    double maximum = numeric_limits<double>::min();
    double minimum = numeric_limits<double>::max();
```

```
    cout << "Random numbers: ";

    for (int i = 0; i < N; ++i) {
        double random_number = dis(gen);
        sum += random_number;

        // Update maximum and minimum
        if (random_number > maximum)
            maximum = random_number;
        if (random_number < minimum)
            minimum = random_number;

        cout << random_number << " ";
    }

    double average = sum / N;

    cout << "\n\nAverage: " << average << endl;
    cout << "Maximum: " << maximum << endl;
    cout << "Minimum: " << minimum << endl;

    return 0;
```

Running of the Program:

```
Microsoft Visual Studio Debu
Enter the number of random double numbers to generate (N): 6
Random numbers: 1.28719 1.91672 1.81472 1.74241 1.41307 1.58323

Average: 1.62622
Maximum: 1.91672
Minimum: 1.28719

C:\Users\Naafiul Hossain\Documents\Lab2\Lab1Task1\NHLab4Task2\x64\Debug\NHLab4Task2.ex
 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugg
le when debugging stops.
Press any key to close this window . . .
```

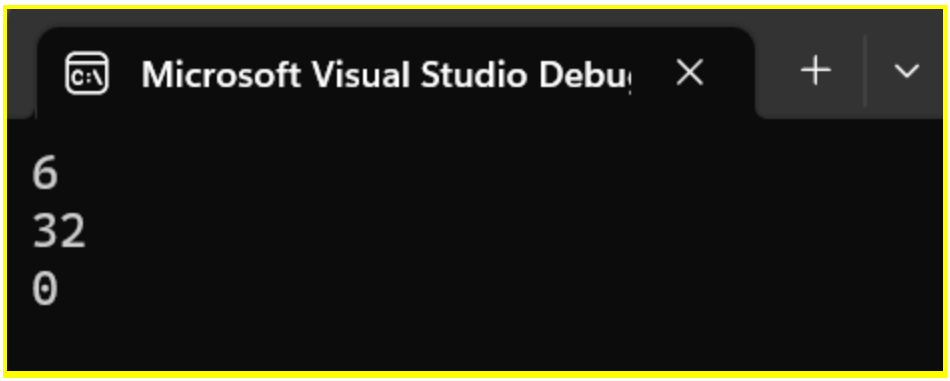
Problem 3

Main.cpp

```cpp
//Naafiul Hossain
//SBU ID: 115107623

#include <iostream>

//Euclidean Algorithm
int g_c_d(int x, int y) {
    if (x < 0 || y < 0) {
        return 0;
    }
    while (x && y) { //if both x and y are non zero
        x %= y;
        std::swap(x, y);

    }
    return x + y;

}
```

```cpp
int main() {
    using std::cout;
    using std::endl;
    cout << g_c_d(42, 6) << endl;
    cout << g_c_d(0, 32) << endl;
    cout << g_c_d(10, -6) << endl;
    return 0;
}
```

Screenshot of the running program:



Problem 4

Player.h

```
//Naafiul Hossain
//SBU ID: 115107623
#pragma once
class Player {
public:
    Player();
    void playround(int dice);
    int getCoins() const;
    void setCoins(int coins);
    int getChoice() const;
    void setChoice(int choice);

private:
    int coins;
    int choice;
};
```

Player.cpp

```cpp
#include <iostream>
#include "Player.h"  // Include the header file to access class declarations

// Constructor
Player::Player() : coins(200), choice(0) {}

// Function to play a round
void Player::playround(int dice) {
    if (choice == dice) {
        coins += 100;  // Double the coins if choice matches dice
    }
    else {
        coins -= 50;   // Lose 50 coins if choice doesn't match dice
    }
}
```

```cpp
// Getter for coins
int Player::getCoins() const {
    return coins;
}

// Setter for coins
void Player::setCoins(int coins) {
    this->coins = coins;
}

// Getter for choice
int Player::getChoice() const {
    return choice;
}

// Setter for choice
void Player::setChoice(int choice) {
    this->choice = choice;
}
```

Main.cpp

```cpp
#include <iostream>
#include <cstdlib>
#include <ctime>
#include "Player.h"  // Include the Player class header file

int rollDice() {
    return rand() % 6 + 1;  // Simulating a dice roll (1-6)
}

int main() {
    // Seed the random number generator
    std::srand(static_cast<unsigned int>(std::time(0)));

    // Instantiate two player objects
    Player player1;
    Player player2;
```

```cpp
    for (int round = 1; round <= 3; ++round) {
        std::cout << "\nRound " << round << ":\n";

        // Get player choices
        std::cout << "Player 1, enter your choice (1-6): ";
        int choice1;
        std::cin >> choice1;
        player1.setChoice(choice1);

        std::cout << "Player 2, enter your choice (1-6): ";
        int choice2;
        std::cin >> choice2;
        player2.setChoice(choice2);

        // Roll the dice
        int diceValue = rollDice();
        std::cout << "Dice value: " << diceValue << "\n";

        // Play the round for each player
        player1.playround(diceValue);
        player2.playround(diceValue);
```

```
    // Display current scores
    std::cout << "Player 1 coins: " << player1.getCoins() << "\n";
    std::cout << "Player 2 coins: " << player2.getCoins() << "\n";
}

// Determine the winner
if (player1.getCoins() > player2.getCoins()) {
    std::cout << "\nPlayer 1 wins!\n";
}
else if (player1.getCoins() < player2.getCoins()) {
    std::cout << "\nPlayer 2 wins!\n";
}
else {
    std::cout << "\nIt's a draw!\n";
}

return 0;
```

Screenshot of the Program running:

```
Round 1:
Player 1, enter your choice (1-6): 4
Player 2, enter your choice (1-6): 3
Dice value: 4
Player 1 coins: 300
Player 2 coins: 150

Round 2:
Player 1, enter your choice (1-6): 2
Player 2, enter your choice (1-6): 3
Dice value: 2
Player 1 coins: 400
Player 2 coins: 100

Round 3:
Player 1, enter your choice (1-6): 6
Player 2, enter your choice (1-6): 2
Dice value: 2
Player 1 coins: 350
Player 2 coins: 200

Player 1 wins!

C:\Users\Naafiul Hossain\Documents\Lab2\Lab1Task1\NHLab4Task
 0.
```

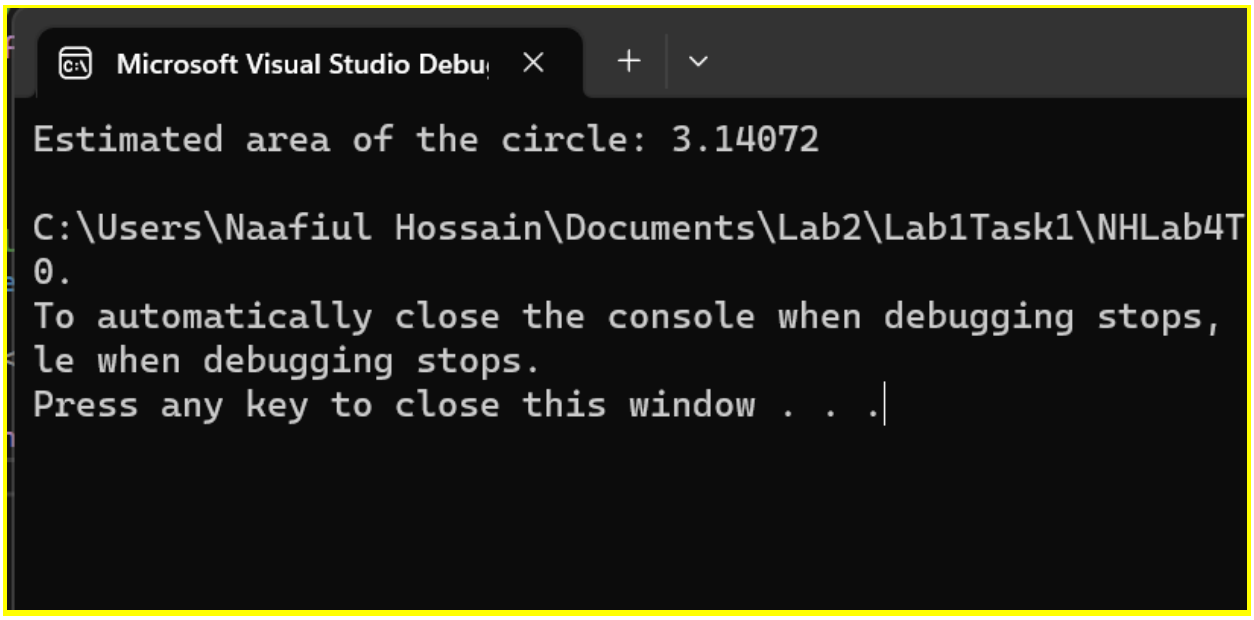
Problem 5

Main.cpp

```cpp
//Naafiul Hossain
//SBU ID: 115107623
#include <iostream>
#include <cstdlib>
#include <ctime>

using namespace std;

int main() {
    srand(time(0)); // Seed the random number generator with the current time

    int totalPoints = 1000000; // Total number of random points to generate
    int pointsInsideCircle = 0; // Counter for points inside the circle

    double circleRadius = 1.0;

    for (int i = 0; i < totalPoints; i++) {
        // Generate random coordinates within the square [-1, 1] x [-1, 1]
        double x = (double)rand() / RAND_MAX * 2.0 - 1.0;
        double y = (double)rand() / RAND_MAX * 2.0 - 1.0;
```

```cpp
        // Check if the point is inside the circle
        if (x * x + y * y <= circleRadius * circleRadius) {
            pointsInsideCircle++;
        }
    }

    // Calculate the area of the circle using the ratio of points inside the circle to total points
    double estimatedArea = 4.0 * pointsInsideCircle / totalPoints;

    cout << "Estimated area of the circle: " << estimatedArea << endl;

    return 0;
```

Screenshot of the running program:



Problem 6

Main.cpp

```cpp
//Naafiul Hossain
// 115107623
#include <algorithm>
#include <iostream>


using namespace std;

int longestPalSubstr(string str)
{
    // get length of input string
    int n = str.size();

    // All substrings of length 1
    // are palindromes
    int maxLength = 1, start = 0;
```

```cpp
    // are palindromes
    int maxLength = 1, start = 0;

    // Nested loop to mark start and end index
    for (int i = 0; i < str.length(); i++) {
        for (int j = i; j < str.length(); j++) {
            int flag = 1;

            // Check palindrome
            for (int k = 0; k < (j - i + 1) / 2; k++)
                if (str[i + k] != str[j - k])
                    flag = 0;

            // Palindrome
            if (flag && (j - i + 1) > maxLength) {
                start = i;
                maxLength = j - i + 1;
            }
        }
    }
```

```
    // return length of LPS
    return maxLength;
}

int main() {
    string s = "Hello";
    cout << longestPalSubstr(s) << endl;
    return 0;
}
```

ScreenShot of the Program running: