Stony Brook University
College of Engineering and Applied Science

ESE 224.L02

Lab 3

Ryan Lin

Professor: Xin Wang

Part 1:
Main.cpp

```cpp
int main(){
    ifstream fin("input.dat");
    ofstream fout("output.dat");
    string content;

    if(fin.fail()){
        cerr << "error opening input file" << endl;
        exit(1);
    }
    if (fout.fail()) {
        cerr << "error opening input file" << endl;
        exit(1);
    }
    cout << "Reading and writing " <<endl;

    fin >> content;
    fout << content;
    fin.close();
    fout.close();
    system("pause");
    return 0;
}
```

Output:

```
Reading and writing
sh: pause: command not found
```

Input.dat

```
1    Laboratory03
```

Output.dat

```
1    Laboratory03
```

Part 2:
Main.cpp

```cpp
int main(){
    ifstream fin("data1.dat");
    ofstream fout("data1report.dat");
    double width, height;
    int rectangleCount = 0, squareCount = 0;
    double maxArea = 0, minArea = MAX, avgArea = 0, squareArea = 0, avgSqareArea = 0, tot
    double minWidth, minHeight, maxWidth, maxHeight;

    if (fin.fail()){
        cerr << "error opening input file" << endl;
        exit(1);
    }
    if (fout.fail()){
        cerr << "error opening output file" << endl;
        exit(1);
    }

    while (fin >> width >> height){
        double area = width * height;
        rectangleCount++;
        totalArea += area;

    if(width == height){
        squareCount++;
        squareArea += area;
    }
    if(area < minArea){
        minArea = area;
        minWidth = width;
        minHeight = height;
    }
    if(area > maxArea){
        maxArea = area;
        maxWidth = width;
        maxHeight = height;
    }
    }
    if (rectangleCount > 0)
    {
        avgArea = totalArea / rectangleCount;
        avgSqareArea = squareArea / squareCount;

        fout << "Max Area: " << maxArea << " " << maxWidth << " " << maxHeight << endl;
        fout << "Min Area: " << minArea << " " << minWidth << " " << minHeight << endl;
        fout << "Average area of all rectangle: " << avgArea << endl;
        fout << "Average area of all square: " << avgSqareArea << endl;

    cout << "Report has been written to data1report.dat." << endl;
    }
    else {
    cout << "No data found in data1.dat." << endl;
    }
    fin.close();
    fout.close();
}
```

data1.dat

```
10 1
1.5 2.4
3 1.8
2 2
4 1
5 0.7
0.2 25
10 0.13
6 0.6
0.4 9
1.2 1.2
```

data1report.dat

```
Max Area: 10 10 1
Min Area: 1.3 10 0.13
Average area of all rectangle: 4.13091
Average area of all square: 2.72

```

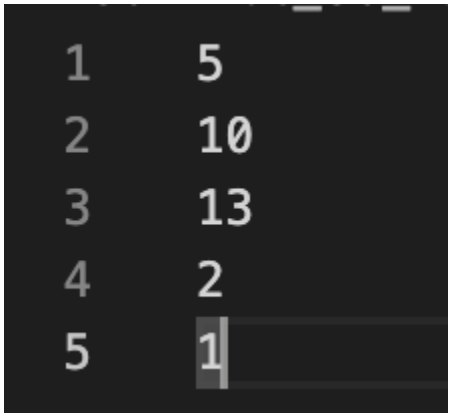
Part 3:
main.cpp

```cpp
int fibonacciSequence(int number)
{
    if (number <= 1) {
        return number;
    }
    return fibonacciSequence(number - 1) + fibonacciSequence(number - 2);
}
int main(){
    int n;
ifstream fin("data3.txt");
ofstream fout("output.txt");

if(fin.fail()){
    cerr << "Error opening the intput file";
    exit(1);
}
if(fout.fail()){
    cerr << "Error opening the output file";
    exit(1);
}
while(fin >> n){
    fout << "Fibonacci of " << n << " is " << fibonacciSequence(n) << endl;
}
cout << "Completed Fibonacci sqeuence";
fin.close();
fout.close();

}
```

data3.txt



output

```
1   Fibonacci of 5 is 5
2   Fibonacci of 10 is 55
3   Fibonacci of 13 is 233
4   Fibonacci of 2 is 1
5   Fibonacci of 1 is 1
6
```

Part 4:
main.cpp

```cpp
int main(){
    ofstream foutCos("cosine.txt");
    ofstream foutSine("sine.txt");
    ifstream finCos("cosine.txt");
    ifstream finSine("sine.txt");
    ofstream foutResult("result.txt");

    int count = 0;
    double rmse = 0.0;
    if(foutCos.fail() || finCos.fail()){
        cerr << "Error cosine.txt could not be opened" << endl;
        exit(1);
    }
    if(foutSine.fail() || finSine.fail()){
        cerr << "Error sine.txt could not be opened" << endl;
        exit(1);
    }
    if(foutResult.fail()){
        cerr << "Error result.txt could not be opened" << endl;
        exit(1);
    }
    int sample;
    cout << "Enter sample size: \n";
    cin >> sample;
    double step = (2 * PI) / sample;
    if (sample < 50)
    {
        cout << "Sample size too small\n";
        exit(1);
    }
    for(int i = 0; i < sample  ; i++){
    double angle;

    angle = i * step;
    foutCos << fixed << setprecision(2) << angle << "\t" << cos(angle) << endl;
    foutSine << fixed << setprecision(2) << angle << "\t" << sin(angle) << endl;
    }
    double read_angle;
    double read_cos, identity, read_sine;
    while(finCos >> read_angle >> read_cos && finSine >> read_angle >> read_sine){
        identity = pow(read_sine,2) + pow(read_cos, 2);
        // cout << identity << endl;
        foutResult << identity << endl;
        count++;

    //calculate RMSE
    double expectedValue = 1.0;
    double error = identity - expectedValue;
        rmse += error * error;
    }

    rmse = sqrt(rmse / count);

    cout << "Root Mean Squared Error (RMSE): " << rmse << endl;

    foutCos.close();
    foutSine.close();
    foutResult.close();
    finCos.close();
    finSine.close();
}
```

Output

```
Enter sample size:
60
Root Mean Squared Error (RMSE): 0.00531865
```

Part 5:

factorial.cpp

```cpp
unsigned long long factorial(int n) {
    unsigned long long result = 1;
    for (int i = 2; i <= n; i++) {
        result *= i;
    }
    return result;
}

int trailZero(unsigned long long number) {
    string numStr = to_string(number);
    int count = 0;
    for (int i = numStr.length() - 1; i >= 0 && numStr[i] == '0'; i--) {
        count++;
    }
    return count;
}

int displayMenu(){
    cout << "Menu:\n1. Calculate the number of trailing zeros\n2. Change the negative number to positive\n3. Terminate\n";
    return 0;
}
```

factorial.h

```h
#ifndef MENU_H
#define MENU_H

unsigned long long factorial(int n);
int trailZero(unsigned long long number);
int displayMenu();

#endif
```

main.cpp

```cpp
int main(){
    ifstream fin("input.txt");
    ofstream foutTemp("temp.txt");
    ifstream finTemp("temp.txt");
    ofstream fout("output.txt");

    int num;
    if(fin.fail()){
        cerr << "Error input.txt cannot be opened";
        exit(1);
    }
    if(fout.fail()){
        cerr << "Error output.txt cannot be opened";
        exit(1);
    }
    if(foutTemp.fail()){
        cerr << "error temp.txt cannot be opened";
        exit(1);
    }
    if(finTemp.fail()){
        cerr << "error temp.txt cannot be opened";
        exit(1);
    }

```

continued…

```
    do{
        fin >> num;
        //using unsigned long long b/c the factorial is positive
        //and will be a big number if the factorial is big
        unsigned long long fact = factorial(num);
        cout << "Number read from the file is " << num << endl;
        int oper = displayMenu();
        cout << "Enter your choice (1/2/3): ";
        cin >> oper;
        switch(oper)
        {
            case 1:
                if (num > 0){
                    int zeroCount = trailZero(fact);
                    fout << zeroCount;
                    cout << "\nResult has been written to output.txt\n";
                }
                break;
            case 2:
                if (num < 0){
                    int posNum = abs(num);
                    cout << "Positive Number: " << posNum << endl;
                    foutTemp << posNum << endl;
                    finTemp >> num;
                if (remove("input.txt") != 0) {
                    cerr << "Error: Unable to delete input file." << endl;
                    return 1;
                }

                if (rename("temp.txt", "input.txt") != 0) {
                    cerr << "Error: Unable to rename temporary file." << endl;
                    return 1;
                }
                }
                break;
            case 3:
                cout << "Program is terminating...\n";
                terminate = 1;
                system("pause");
                break;
            default:
                cout << "Invalid choice. Please try again\n";
        }
    }while (terminate != 1);

    fin.close();
    fout.close();
}
```

Output

```
p && ./factorial
Number read from the file is 10
Menu:
1. Calculate the number of trailing zeros
2. Change the negative number to positive
3. Terminate
Enter your choice (1/2/3): 1

Result has been written to output.txt
Number read from the file is 10
Menu:
1. Calculate the number of trailing zeros
2. Change the negative number to positive
3. Terminate
Enter your choice (1/2/3): 3
Program is terminating...
sh: pause: command not found
```

Output.txt

```
1    2
```