LAB ASSIGNMENT 1

NORTHERN UNIVERSITY
BANGLADESH
Knowledge for Innovation and Change

SUBMITTED BY

NAME: *Sadia Islam*

ID: *41220300363*

DEPARTMENT: *CSE*

SECTION: *4B*

SUBJECT: *Algorithm design and analysis lab work*

SEMESTER: *Fall 2023*

SUBMITTED TO

Md. Mahadi Hasan

(Lecturer of CSE in NUB)

Submitted in 07 October 2023

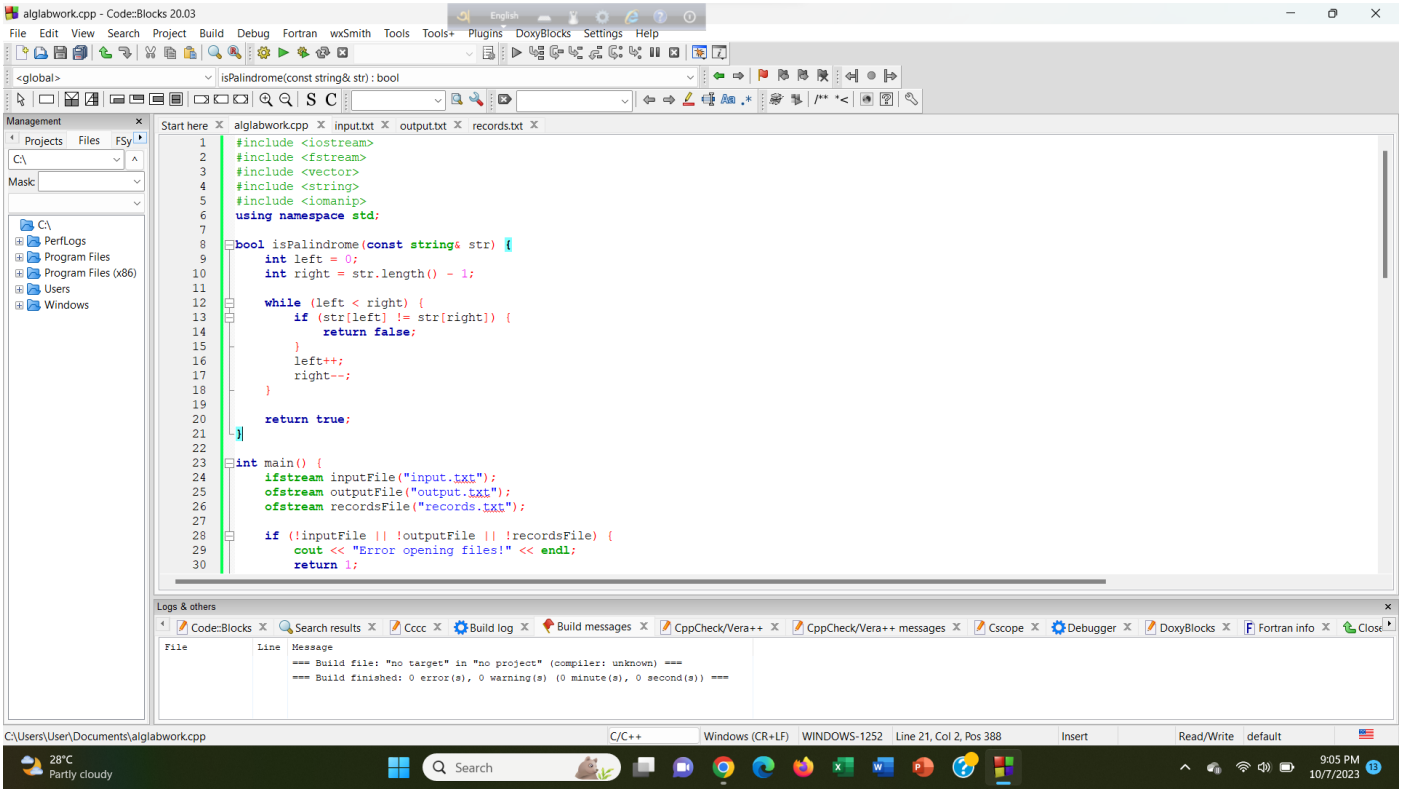
Problem 1:

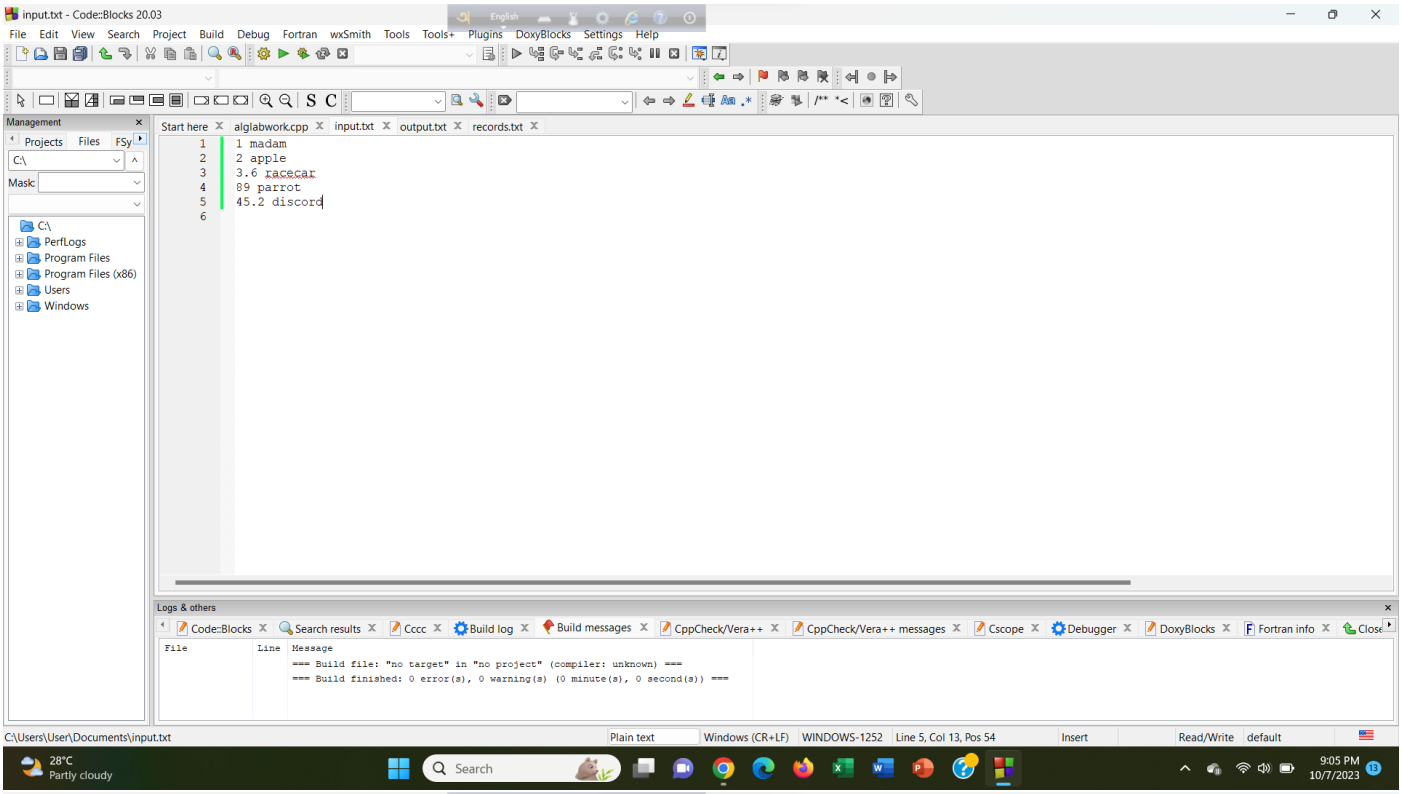
C++ code:

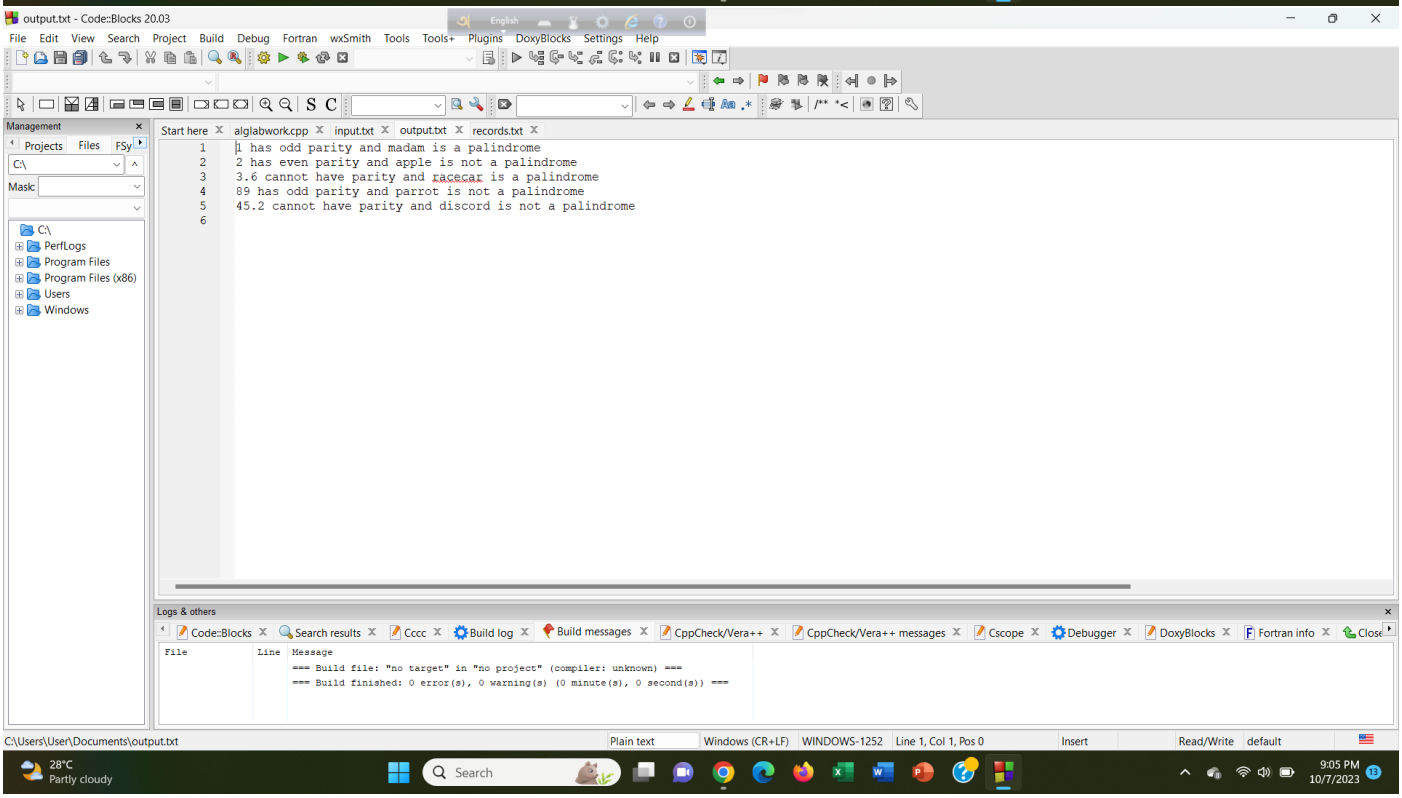
```
#include <iostream>
#include <fstream>
#include <vector>
#include <string>
#include <iomanip>
using namespace std;

bool isPalindrome(const string& str) {
    int left = 0;
    int right = str.length() - 1;
    while (left < right) {
        if (str[left] != str[right]) {
            return false;
        }
        left++;
        right--;
    }
  return true;
}
int main() {
    ifstream inputFile("input.txt");
    ofstream outputFile("output.txt");
    ofstream recordsFile("records.txt");

    if (!inputFile || !outputFile || !recordsFile) {
        cout << "Error opening files!" << endl;
        return 1;
    }
   vector<string> lines;
    string line;
   while (getline(inputFile, line)) {
```

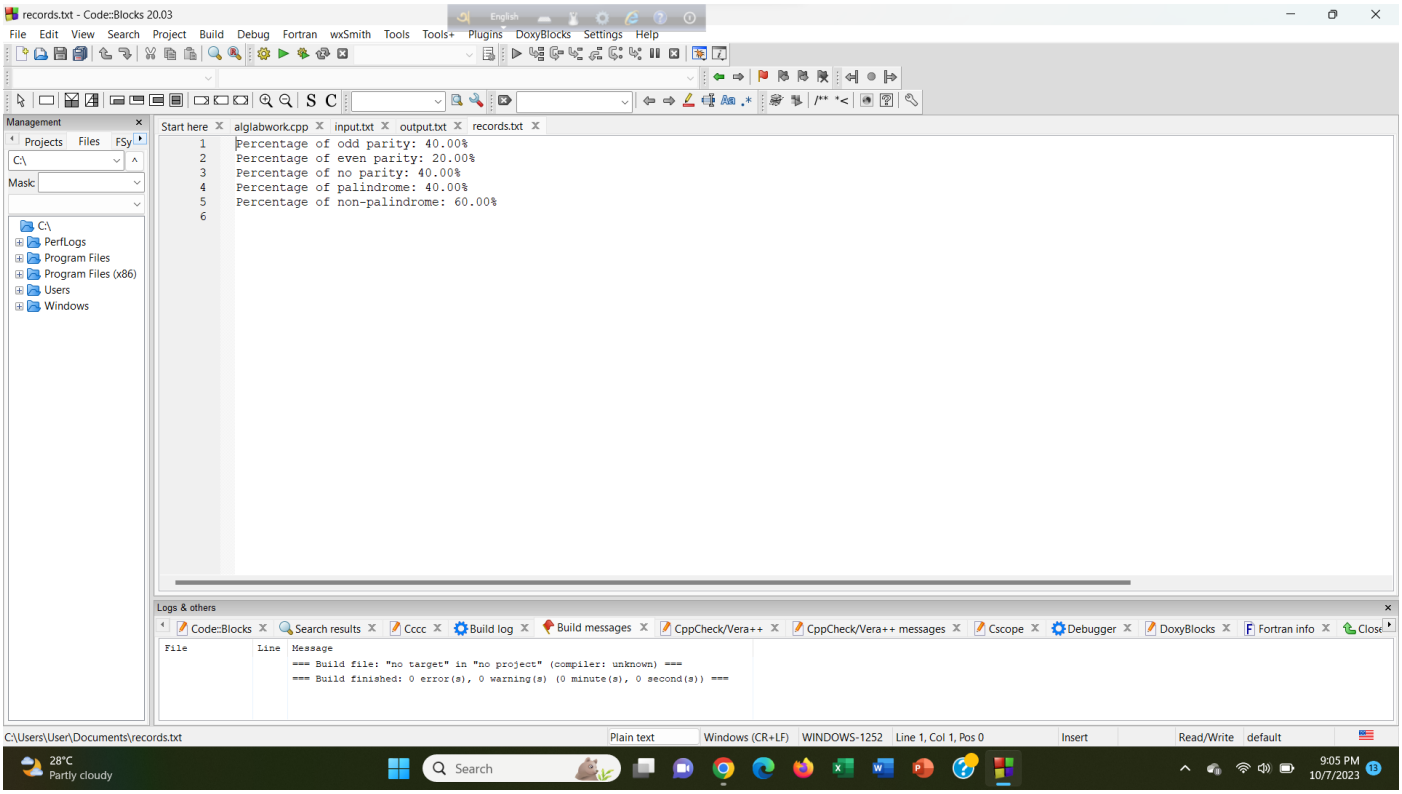
```
    lines.push_back(line);
}
int totalNumbers = 0;
int oddParityCount = 0;
int evenParityCount = 0;
int noParityCount = 0;
int palindromeCount = 0;
int nonPalindromeCount = 0;

for (const string& inputLine : lines) {
    istringstream iss(inputLine);
    double num;
    string word;
if ( iss>>num >> word) {
        totalNumbers++;
if (num == static_cast<int>(num)) {
            if (static_cast<int>(num) % 2 == 0) {
                outputFile << num << " has even parity";
                evenParityCount++;
            } else {
                outputFile << num << " has odd parity";
                oddParityCount++;
            }
        } else {
            outputFile << num << " cannot have parity";
            noParityCount++;
        }
     if (isPalindrome(word)) {
            outputFile << " and " << word << " is a palindrome" << endl;
            palindromeCount++;
        } else {
            outputFile << " and " << word << " is not a palindrome" << endl;
            nonPalindromeCount++;
        }
```

```
        }
    }
    double oddParityPercentage = (static_cast<double>(oddParityCount) / totalNumbers) * 100;
    double evenParityPercentage = (static_cast<double>(evenParityCount) / totalNumbers) * 100;
    double noParityPercentage = (static_cast<double>(noParityCount) / totalNumbers) * 100;
    double palindromePercentage = (static_cast<double>(palindromeCount) / totalNumbers) * 100;
    double nonPalindromePercentage = (static_cast<double>(nonPalindromeCount) / totalNumbers) * 100;
   recordsFile << "Percentage of odd parity: " << fixed << setprecision(2) << oddParityPercentage << "%"
<< endl;
    recordsFile << "Percentage of even parity: " << evenParityPercentage << "%" << endl;
    recordsFile << "Percentage of no parity: " << noParityPercentage << "%" << endl;
    recordsFile << "Percentage of palindrome: " << palindromePercentage << "%" << endl;
    recordsFile << "Percentage of non-palindrome: " << nonPalindromePercentage << "%" <<endl;
    inputFile.close();
    outputFile.close();
    recordsFile.close();
   return 0;
}
```




input.txt - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSy
C:\
Mask:
C:\
PerfLogs
Program Files
Program Files (x86)
Users
Windows
Start here
alglabwork.cpp
input.txt
output.txt
records.txt
1 madam
2 apple
3.6 racecar
89 parrot
45.2 discord
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
Fortran info
Close
File
Line
Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
C:\Users\User\Documents\input.txt
Plain text
Windows (CR+LF)
WINDOWS-1252
Line 5, Col 13, Pos 54
Insert
Read/Write
default
28°C
Partly cloudy
Search
9:05 PM
10/7/2023


output.txt - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Management
Projects Files FSy
C:\
Mask:
C:\
PerfLogs
Program Files
Program Files (x86)
Users
Windows
Start here
alglabwork.cpp
input.txt
output.txt
records.txt
1 has odd parity and madam is a palindrome
2 has even parity and apple is not a palindrome
3.6 cannot have parity and racecar is a palindrome
89 has odd parity and parrot is not a palindrome
45.2 cannot have parity and discord is not a palindrome
Logs & others
Code::Blocks
Search results
Cccc
Build log
Build messages
CppCheck/Vera++
CppCheck/Vera++ messages
Cscope
Debugger
DoxyBlocks
Fortran info
Close
File
Line
Message
=== Build file: "no target" in "no project" (compiler: unknown) ===
=== Build finished: 0 error(s), 0 warning(s) (0 minute(s), 0 second(s)) ===
C:\Users\User\Documents\output.txt
Plain text
Windows (CR+LF)
WINDOWS-1252
Line 1, Col 1, Pos 0
Insert
Read/Write
default
28°C
Partly cloudy
Search
9:05 PM
10/7/2023



Problem 2:

Here is the implementation of finding the N-th Fibonacci number using both methods in C++ along with the complexity analysis:

For Loop method:

```
#include <iostream>

using namespace std;

int fibonacci(int n) {
    if (n <= 0) {
        return 0;
    } else if (n == 1) {
        return 1;
    } else {
        int a = 0, b = 1;
        for (int i = 2; i <= n; i++) {
            int temp = a + b;
            a = b;
            b = temp;
```

```
        }
        return b;
    }
}

int main() {
    int n;
    cout << "Enter the value of N: ";
    cin >> n;
    cout << "The " << n << "th Fibonacci number is: " << fibonacci(n) << endl;
    return 0;
}
```

The complexity of this method is O(N) since we iterate N times to calculate the N-th Fibonacci number.

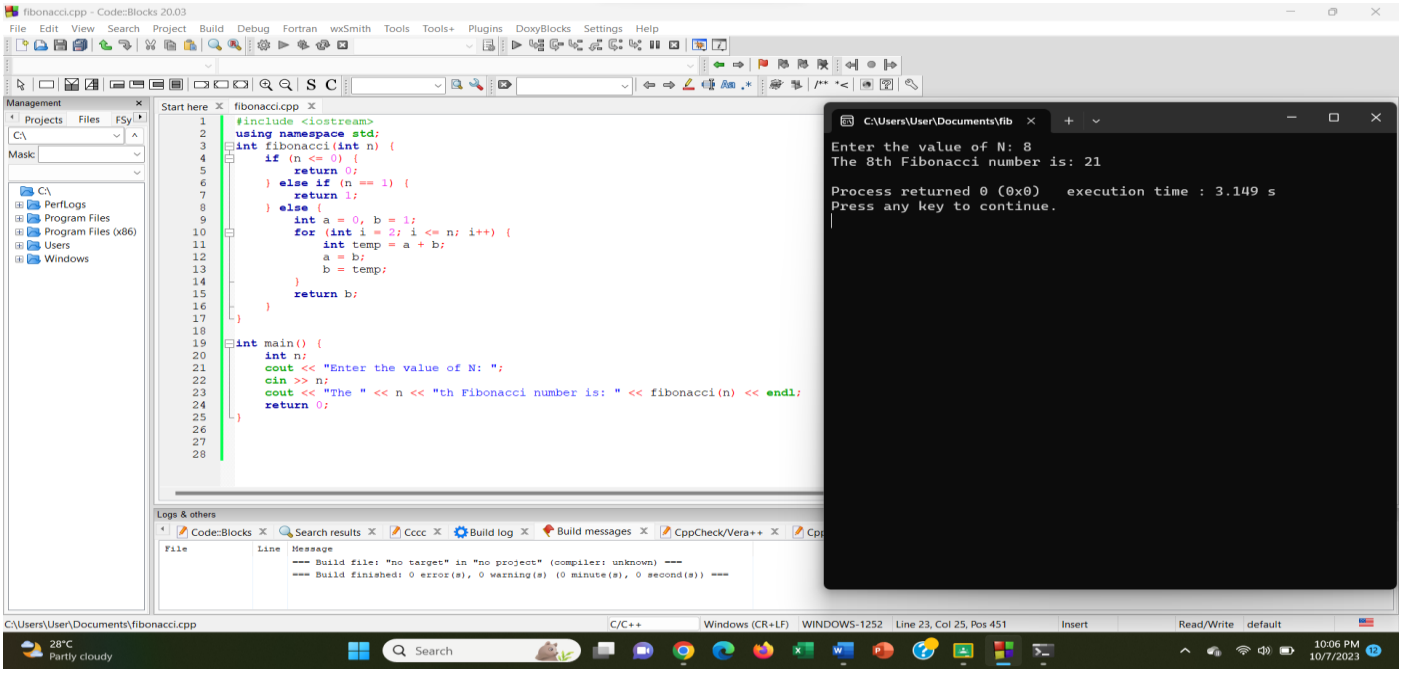
Recursion method:

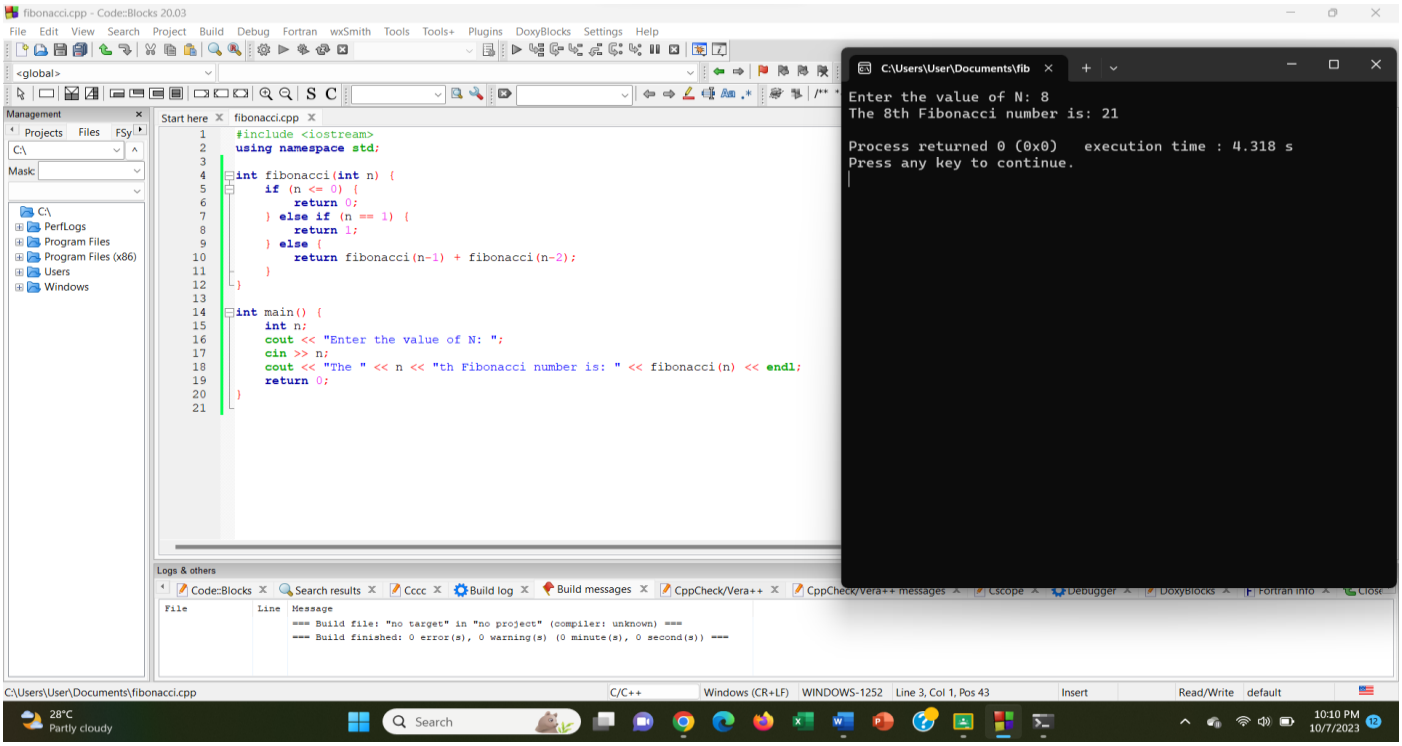
```
#include <iostream>
using namespace std;
int fibonacci(int n) {
    if (n <= 0) {
        return 0;
    } else if (n == 1) {
        return 1;
    } else {
        return fibonacci(n-1) + fibonacci(n-2);
    }
}

int main() {
    int n;
    cout << "Enter the value of N: ";
    cin >> n;
```

```
    cout << "The " << n << "th Fibonacci number is: " << fibonacci(n) << endl;
    return 0;
}
```

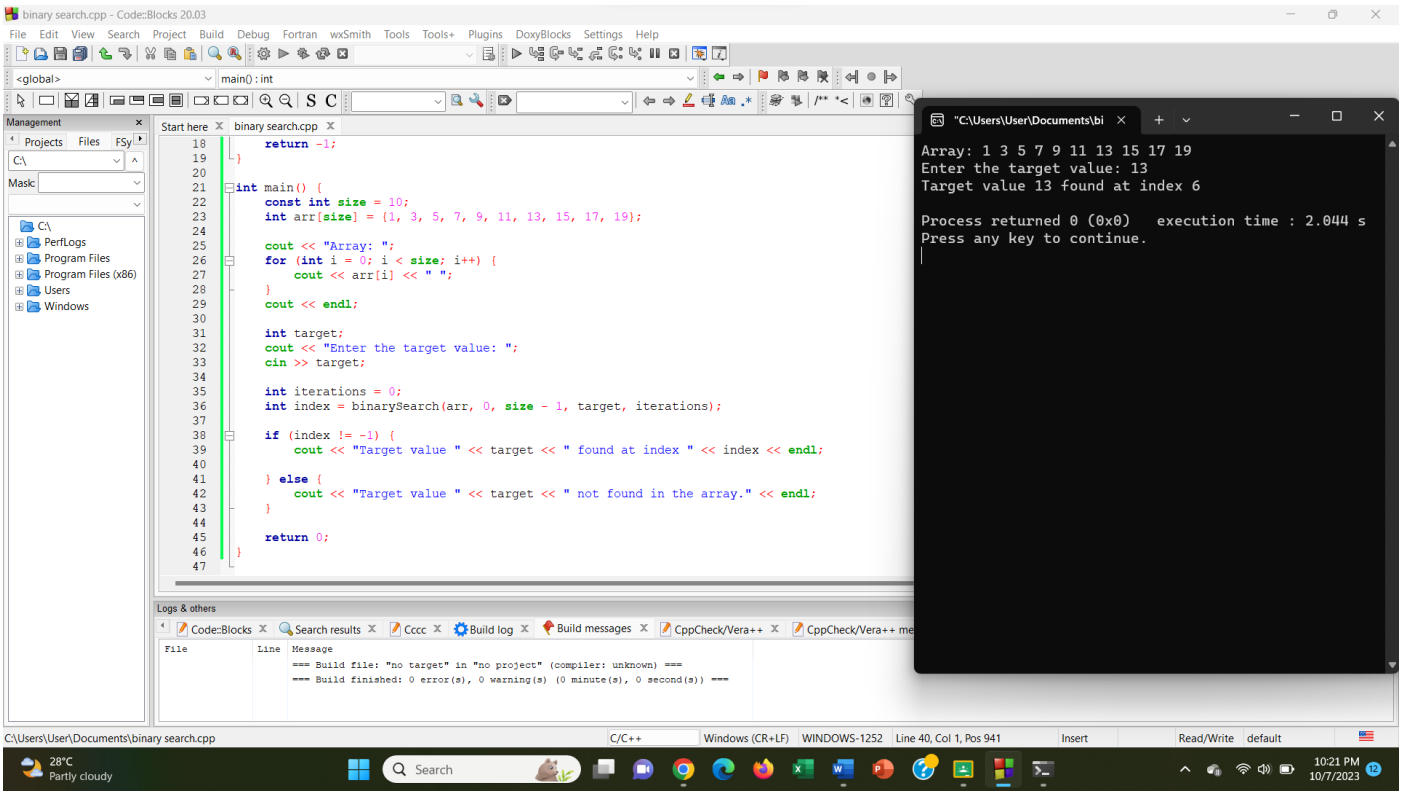
The complexity of this method is exponential, O(2^N), since we make two recursive calls for each recursive call until we reach the base case. Therefore, the recursion method is less efficient compared to the for loop method.

In conclusion, the for loop method has a linear complexity of O(N) while the recursion method has an exponential complexity of O(2^N).





Problem 3:




```
#include <iostream>
using namespace std;
int binarySearch(int arr[], int low, int high, int target, int& iterations) {
    while (low <= high) {
        int mid = low + (high - low) / 2;

        if (arr[mid] == target) {
            return mid;
        } else if (arr[mid] < target) {
            low = mid + 1;
        } else {
            high = mid - 1;
        }

        iterations++;
    }
```

```
    return -1;
}

int main() {
    const int size = 10;
    int arr[size] = {1, 3, 5, 7, 9, 11, 13, 15, 17, 19};

    cout << "Array: ";
    for (int i = 0; i < size; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;

    int target;
    cout << "Enter the target value: ";
    cin >> target;

    int iterations = 0;
    int index = binarySearch(arr, 0, size - 1, target, iterations);

    if (index != -1) {
        cout << "Target value " << target << " found at index " << index << endl;

    } else {
        cout << "Target value " << target << " not found in the array." << endl;
    }

    return 0;
}
```