ECE 2305

Introduction to Computer Programming

Programming Project 09

Cryptography

Write a C++ program that will accept a text message from a disc file and produce a coded or decoded message.

The code shall use a letter substitution cipher. The letters are identified by their numerical position of the letter (a → 0, b → 1, … z → 25). The $code\ number$ shall be calculated as follows

$code\ number = offset + spacing \times (letter\ number)$

Where $offset$ is an offset value and $spacing$ gives the spacing between letters. The $code\ number$ is to “wrap around” so the values will range from 0 to 25. The offset value $offset$ is an integer ranging from 1 to 25. The spacing value $spacing$ is a positive odd integer not greater than 11. These values are to be based on the number of characters in the message, including spaces, punctuation and other characters. Use the following formulas for these values:

```
int offset = length % 25 + 1; //returns a value from 1 to 25
int spacing = (length % 6) * 2 + 1;
//returns a positive odd value <= 11
```

where $length$ is the number of characters in the message.

Implement the code cipher by creating a lookup-table array. This same look-up table is to be used in decoding messages. Revise the input text to contain all lower-case letters. Apply the code to lower case letters only. Write the program to allow the user to choose to encode regular messages and to decode coded messages. Write the encoded and decoded messages to .txt disc files.

Document the program with the following:

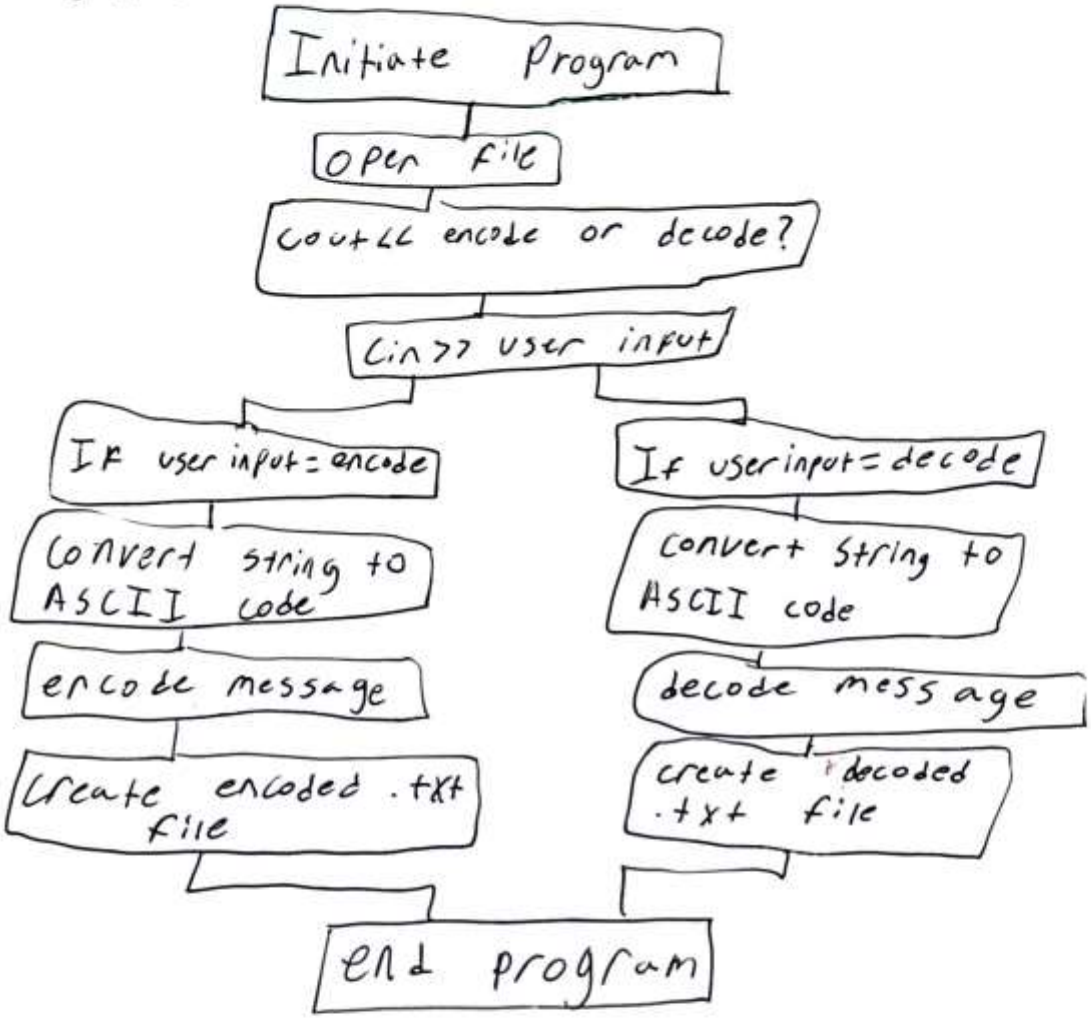
A. A written description of the program that describes the purpose of the program and

the structure of the program.

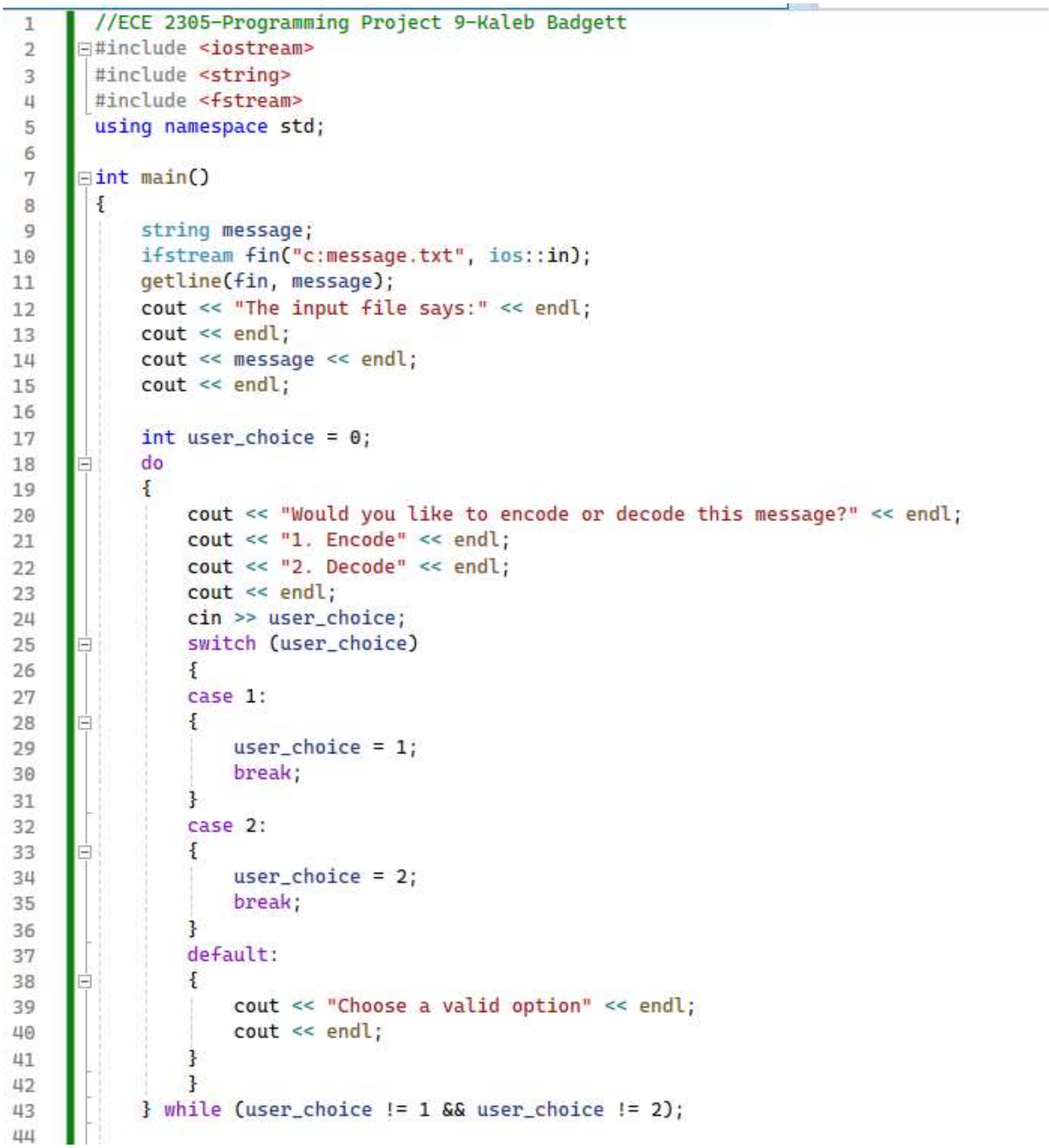
The purpose of this program is to encode or decode certain .txt files. It opens files, reads them, then uses a do while loop to ask the user whether to encode or decode the message. It then goes through a series of for loops to complete the assigned task.

B. A flowchart to graphically illustrate the structure of the program.

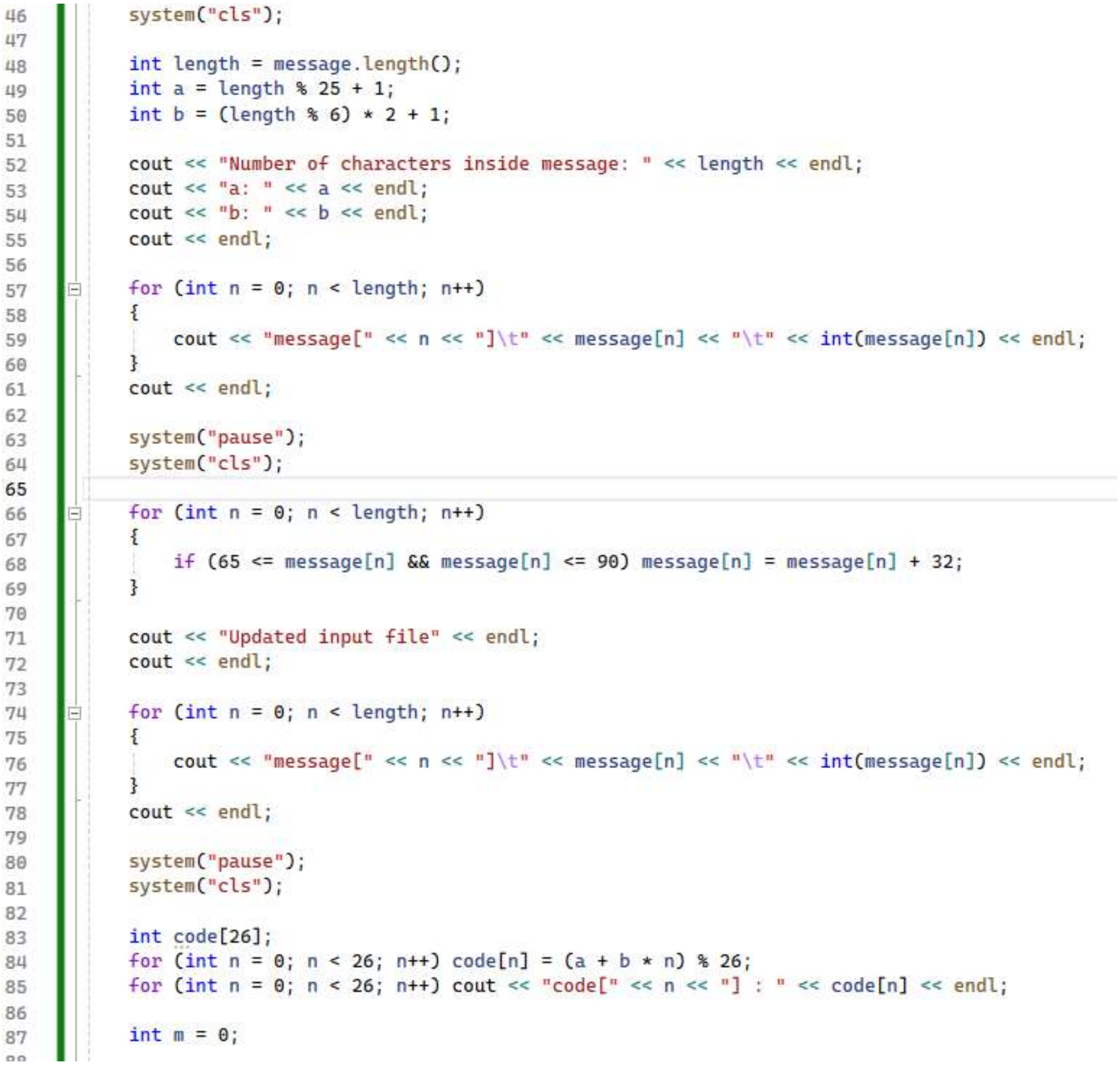
Programming Project 9 Flow Chart 3/27/2024



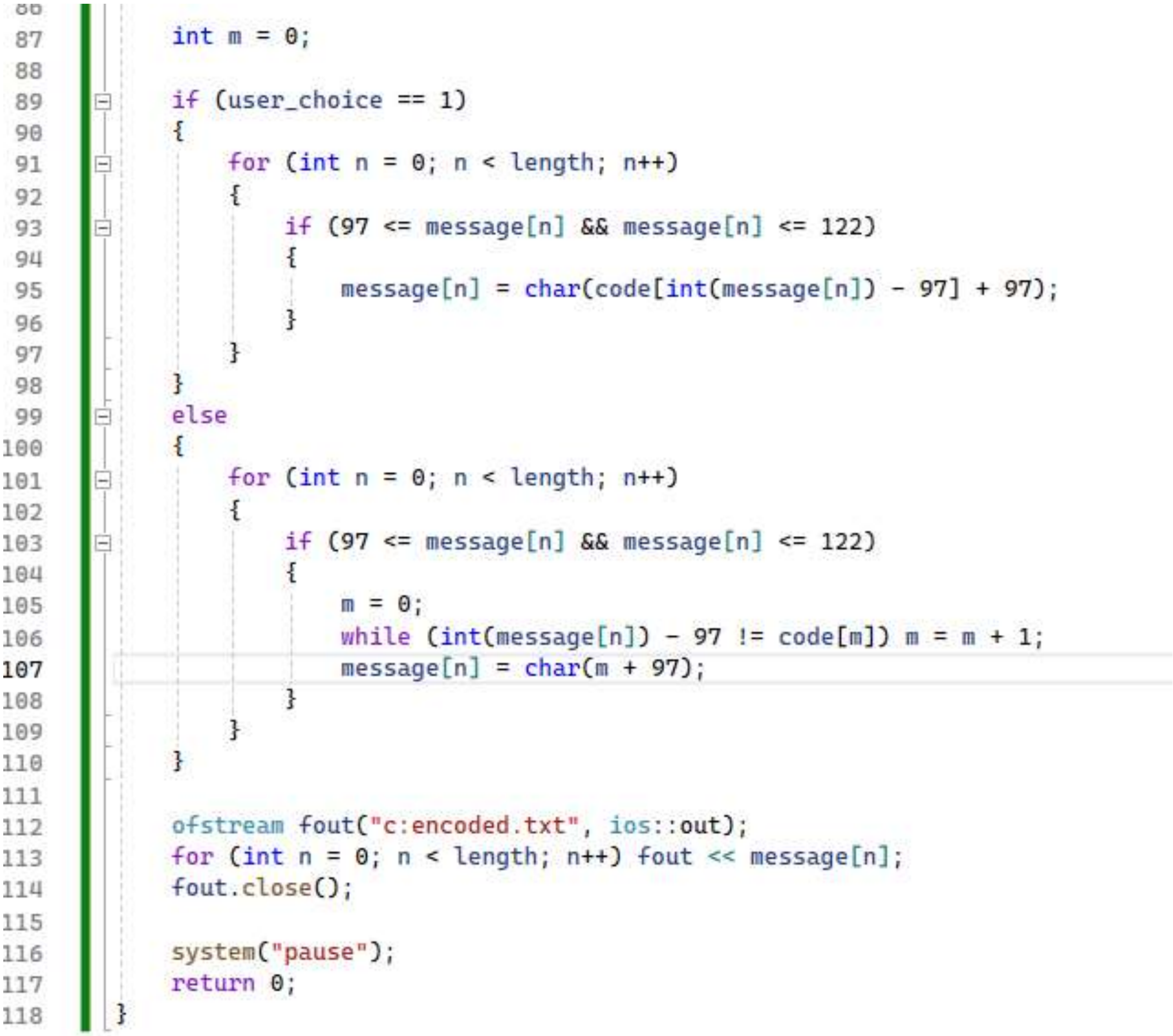
C. The code listing.



```
//ECE 2305-Programming Project 9-Kaleb Badgett
#include <iostream>
#include <string>
#include <fstream>
using namespace std;

int main()
{
    string message;
    ifstream fin("c:message.txt", ios::in);
    getline(fin, message);
    cout << "The input file says:" << endl;
    cout << endl;
    cout << message << endl;
    cout << endl;

    int user_choice = 0;
    do
    {
        cout << "Would you like to encode or decode this message?" << endl;
        cout << "1. Encode" << endl;
        cout << "2. Decode" << endl;
        cout << endl;
        cin >> user_choice;
        switch (user_choice)
        {
        case 1:
        {
            user_choice = 1;
            break;
        }
        case 2:
        {
            user_choice = 2;
            break;
        }
        default:
        {
            cout << "Choose a valid option" << endl;
            cout << endl;
        }
        }
    } while (user_choice != 1 && user_choice != 2);

```



```cpp
    system("cls");

    int length = message.length();
    int a = length % 25 + 1;
    int b = (length % 6) * 2 + 1;

    cout << "Number of characters inside message: " << length << endl;
    cout << "a: " << a << endl;
    cout << "b: " << b << endl;
    cout << endl;

    for (int n = 0; n < length; n++)
    {
        cout << "message[" << n << "]\t" << message[n] << "\t" << int(message[n]) << endl;
    }
    cout << endl;

    system("pause");
    system("cls");

    for (int n = 0; n < length; n++)
    {
        if (65 <= message[n] && message[n] <= 90) message[n] = message[n] + 32;
    }

    cout << "Updated input file" << endl;
    cout << endl;

    for (int n = 0; n < length; n++)
    {
        cout << "message[" << n << "]\t" << message[n] << "\t" << int(message[n]) << endl;
    }
    cout << endl;

    system("pause");
    system("cls");

    int code[26];
    for (int n = 0; n < 26; n++) code[n] = (a + b * n) % 26;
    for (int n = 0; n < 26; n++) cout << "code[" << n << "] : " << code[n] << endl;

    int m = 0;
```



```cpp
    int m = 0;

    if (user_choice == 1)
    {
        for (int n = 0; n < length; n++)
        {
            if (97 <= message[n] && message[n] <= 122)
            {
                message[n] = char(code[int(message[n]) - 97] + 97);
            }
        }
    }
    else
    {
        for (int n = 0; n < length; n++)
        {
            if (97 <= message[n] && message[n] <= 122)
            {
                m = 0;
                while (int(message[n]) - 97 != code[m]) m = m + 1;
                message[n] = char(m + 97);
            }
        }
    }

    ofstream fout("c:encoded.txt", ios::out);
    for (int n = 0; n < length; n++) fout << message[n];
    fout.close();

    system("pause");
    return 0;
}
```

These items shall be written in a PDF document.

D. Text files of an example input message and the corresponding coded message.

E. Decode the two coded messages and save the decoded massages as a text file.

Upload the PDF document and all of the text files to Blackboard.