

Nested While Loop

1. Write a C++ program that lets the user guess the number you have in mind. The number is between 1 and 100. If the user guess is too low, print the message "your guess is too low", if the user guess is too high, print the message "your guess is too high". It continues until the user guess the number. Print "Congratulations! You guessed it".
2. Modify the program written above so that the game continues after the number is guessed correctly by the user. Ask the user "Would you like to try another guess?". If the user answers 'y' or 'Y', repeat the game with a new number. Otherwise, terminate the program.

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

int main()
{
    int    theNumber, playerGuess;
    char   answer;
    bool   guessed = false;

    cout << "Do you like to play a guessing game?";
    cout << " ('y' or 'Y' for yes)" << endl;
    cin >> answer;

    while (answer == 'y' || answer == 'Y') {
        cout << "I am thinking of an integer number between 1 and 100 ... \n";
        theNumber = rand()%100 + 1;    // rand() function is defined in
        <cstdlib>

        cout << "What is your guess?" << endl;
        cin >> playerGuess;

        while ( !guessed) {
            if ( playerGuess > theNumber)
                cout << "Your guess is too high" << endl;
            else if (playerGuess < theNumber)
                cout << "Your guess is too low" << endl;
            else
                guessed = true;

            if (!guessed) {
                cout << "What is your guess?" << endl;
                cin >> playerGuess;
            }
        }

        cout << "Congratulations, you guessed my number. " << endl;

        guessed = false;
        cout << "Do you want to guess another number ?" << endl;
        cin >> answer;
    }
}
```

```

    }

    cout << "Thank you, hope you had fun!" << endl;

    return 0;
}

```

3. Read a text file and count the number of lines in the file:

```

char    nextChar;
int     lineCount;
int     charCount;
ifstream myIn;

myIn.open("StarWar");

charCount = 0;
lineCount = 0;
myIn.get(nextChar);
while (myIn)
{
    while (myIn && (nextChar != '\n'))
    {
        if (nextChar != ' ')
            charCount++;

        myIn.get(nextChar);
    }

    lineCount++;
    myIn.get(nextChar);
}

```

4. Read a text file and count how many **non-blank** lines there are in the file

```

char    nextChar;
int     lineCount;
int     charCount;
bool    blankLine; // flag
ifstream myIn;

myIn.open("StarWar");

charCount = 0;
lineCount = 0;
myIn.get(nextChar);
while (myIn)
{
    blankLine = true;
    while (myIn && (nextChar != '\n'))
    {

```

```
        if (nextChar != ' '){
            charCount++;
            // assume the non-blank
            //lines only have blanks
            blankLine = false;
        }

        myIn.get(nextChar);
    }

    if ( ! blankLine)
        lineCount++;

    myIn.get(nextChar);
}
```