```
#include <iostream>
#include <thread>
#include <chrono>
#include <iostream>
#include <fstream>
#include <string>


using std::cout;
using std::endl;
using std::string;
using std::ifstream;

class MSCcode
{
public:

	MSCcode();  //constructor
	void display(void);  //display options
	bool checkcode(void);

};

void checkcode()
{
	bool std = false; // this is used for checking if the program has good security
practices in place.

	int opens = 0;
	// creates a variable called opens used for checking how many opening loops
there are
	int closes = 0;
	// creates a variable called closes used for checking how many closing loops
there are
	// these two variables can later be compared to check if loops are opening and
closing correctly.

	int lines = 0;
	//  lines is used to see the total number of lines in the program
	int endlines = 0;
	// endlines is used to store the interger of all ; used to end lines

	int blanklines = 0;
	// a variables called blanklines to work out how much white lines are in the
program

	int hashtags = 0;
	// the above interger will be used to count the number of included headerfiles
in the program.

	string myText;
	// creates a variables called mytext to store and output the code being read in
for testing

	// name of the testcode file ready for future testing.
	ifstream MyReadFile("testcode.txt");

	// Use a while loop together with the getline() function to read the file line
by line
	while (getline(MyReadFile, myText)) {
```

```
            std::string s(myText);

            for (char& c : s)
            {
                    // std::cout << "One character: " << c << "\n";
                    // the above code was for displaying the charicters during
testing
                    if (c == 59)
                    {
                            endlines++; // counts the number of end lines
                    }

                    // 123 is the charicter code for {
                    else if (c == 123)
                    {
                            opens++; // counts how many code brackets were opened
                    }

                    // 125 is the charicter code for }
                    else if (c == 125)
                    {
                            closes++; // counts how many code brackets were closed
                    }

                    // counts how many includes were used in the project
                    else if (c == 35)
                    {
                            hashtags++;
                    }
                    if (myText == "using namespace std;")
                    {
                            std = true;
                    }



                    c = '*';
            }

            // Output the text from the file and end each line as appropriate to
make sure the output is correct
            cout << myText << endl;


            lines++; // counts the numnber of total lines in the code


        }
        cout << "number of code brackets opened " << opens << endl;
        cout << "number of code brackets closed " << closes << endl;
        blanklines = lines - endlines - closes - opens - hashtags;
        cout << "total number of headers in tested code " << hashtags << endl;
        cout << "total number of blanklines in tested code " << blanklines << endl;
        cout << "number of end lines " << endlines << endl;
        cout << "total number of lines in tested code " << lines << endl;


        if (opens == closes)
        {
                cout << "all loops are working correctly " << endl;
        }
        else if (closes > opens)
```

```
	{
		cout << "error found in loop opening  " << endl;
	}
	else
	{
		cout << "error found in loop closing " << endl;
	}

	if (std == true)
	{
		cout << endl;
		cout << "please do not use using namespace std;" << endl;
		cout << "it is bad practice from a security standpoint due to all the
packages it contains" << endl;
		cout << "try using std::cout; instead this means other types will" <<
endl;
		cout << "not be used by hackers in weakspots later as they will not be
available" << endl;

	}

	if (hashtags >= 8)
	{
		cout << endl;
		cout << "please delete some headers if possible;" << endl;
		cout << "as the number increased so does the security risk" << endl;


	}


	// Close the file
	MyReadFile.close();
	std::this_thread::sleep_for(std::chrono::milliseconds(200000));
}

void display()
//draw puzzle as it is now
{
	system("cls"); // blanks the screen its clear screen
	cout << "the program will now start looking for errors please wait" << endl;
	//cout << "please press 2 for code optimization" << endl;
	//cout << "please press 3 to exit the program" << endl;
	//cout << endl;
	//cout << "please choose a valid option " << endl;
	cout << endl;
	// this will display the options available for the main menu
}

int main(void)
{


	//main menu for code security error finder
	while (true) //go forever
	{
		display(); //display menu options
		checkcode(); // checks if puzzle is solved


	}
	return 0;
```

```
}//the end
```