CS 213-M Assignment 3

Question 1

We are providing you with a file reading API. It supports the following operations:

1. **`int MFile::init(string filename, int linesize);`**
   This function loads the file and sets it up for reading. linesize is the number of characters in a line.
2. **`string MFile::readNext();`**
   This function gets the next line in sequence. If it could not read another line, it returns the empty string.
3. **`vector<string> MFile::readLines(vector<int> lines);`**
   This function reads the line numbers passed in the vector. This function takes linear time(in the file size) to execute. Keep calls to this function as low as possible.
4. **`int MFile::close();`**
   Tells the API that file operations are done, and it can release the file resources.

Your task, if you choose to accept it is the following: Count the number of unique lines in the entire file. Input will be two lines: The name of the file is the first line, and the number of characters in a line will be the second line. Print the number of unique lines as the output. Submit the file with a name **count_uniq.cpp.** It must contain a main function.

Important note: The number of duplicates will be approximately 2% to 5%. Use this to your advantage.

eg.

Input:
x.txt
10

Output:
3

where x.txt is

```
HelloDanny
ByeDaniele
WhereAreYu
HelloDanny
ByeDaniele
```

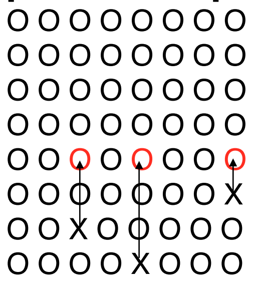
Question 2

We are going to solve the N-queens problem, and solve it really fast. To do this, we will be using a bit representation of the board, which speeds things up drastically. The basic idea for solving the problem is placing one queen at a time on the board. Suppose we are solving for N = 8. Lets say 3 queens have already been placed. Hence, the board will look like this:

```
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O X
O O X O O O O O
O O O O X O O O
```

We will be tracking three things for each row. The dangers due to vertical attacks, the dangers due to bottom left to top right diagonal attack (Diag1) and the dangers due to bottom right to top left diagonal attack (Diag2).

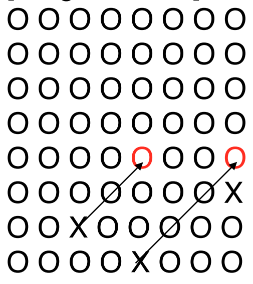
[Vertical attacks]



```
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O X
O O X O O O O O
O O O O X O O O
```

This will be represented as 41 in a C++ signed integer. The bit representation (From MSB to LSB) is 000…000101001

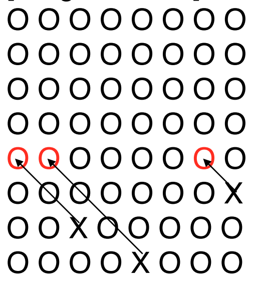
[Diag1 attacks]



```
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O X
O O X O O O O O
O O O O X O O O
```

This will be represented as 9 in a C++ signed integer. The bit representation (From MSB to LSB) is 000…000001001

[Diag2 attacks]



```
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O O
O O O O O O O X
O O X O O O O O
O O O O X O O O
```

This will be represented as 194 in a C++ signed integer. The bit representation (From MSB to LSB) is 000…011000010

You have been provided with a super detailed skeleton code. **DO NOT CHANGE queen.cpp.** Modify queen_defs.h and only submit queen_defs.h. There is no cpp file required. Please go through the comments in the code, because that is where the secret lies :)