Assignment
Where's Wild Bill

At one time, millions of Bison in large herds roamed the North American great plains. Now the numbers are much less, resulting in preservation efforts. These efforts include tagging selected Bisons with tracking and status devices so that information regarding roaming range and health information statistics can be collected.

Wild Bill 1 in shorter grass

A bison selected for tagging, designated as Wild Bill 1, is in an area in the plains covered with extremely tall range grass. Wildlife Department of the Interior has asked Ranger Smith to tag Wild Bill 1. Ranger Smith will use a humane tranquilizer dart and then tag the sedated bison.

Ranger Smith will land in a helicopter in an area near where Wild Bill 1 is grazing. Ranger Smith will stealthily stalk by crawling on his hands and knees down wind through the tall grass so he can approach undetected.

Assignment
Where's Wild Bill

This is very challenging since Smith is having trouble spotting Wild Bill from this vantage point. The grass in front of Smith looks like a string of n (1 <= n <= 50,000) parentheses; for example:)((()())()) .

Smith knows that Wild Bill 1's hind legs look just like an adjacent pair of left parentheses ((, and that Wild Bill 1's front legs look exactly like a pair of adjacent right parentheses)).

Wild Bill 1's location can therefore be described by a pair of indices x < y such that ((is found at position x, and)) is found at position y.

Please compute the number of different such possible locations at which Wild Bill might be standing.

Calculate the number of patterns and the time to do the calculation for each of the input nbisonsearchin.txt files, where n = 01 to 10.

The test case files are located in :

~smd013000/courses/3377/assigns/01/testcases

on the UTD Linux systems.

Use these test case nbisonsearchin.txt input files to run your program.

Each nbisonsearchin.txt input file contains string of parentheses of length N (1 <= N <= 50,000).

Example:
file bisonsearchin.txt:
)((()())())

Check to make sure each file nbisonsearchin.txt opens successfully, where n is the file number.

Your program must produce a corresponding output nbisonfoundin.txt file for each nbisonsearchin.txt input file.

The output nbisonfoundin.txt file must contain the following:

- Time Elapsed
- Found Pattern Count
- Searched Pattern

Assignment
Where's Wild Bill

Sample contents of an output nbisonfoundin.txt file:

Time Elapsed (nano) :   500

Found Pattern Count : 4

Searched Pattern   :

) ((( ) ( )) ( ))

Timing

You must use `chrono` to time the algorithm in MONOTONIC nanoseconds.

```
#include <chrono>
#include <limits>
#include <locale>

#include "LineInfo.h"

using namespace std;
using namespace chrono;


:
:
auto timeStart = steady_clock::now();
what you want to time…
:
:
auto timeElapsed = duration_cast<nanoseconds> (steady_clock::now() – timeStart);
```

You must use C++ programming, for example streams and other C++ constructs.

-o is the output file name specifier

Zip up your source code .cpp file(s) and .h file(s) and the test case input files t this assignment on black board.