Algorithm

Assignment #1

Due: Mar.-24 (Wed.) (before 11:59pm)

Instruction.

a. You have 2 problems in this assignment.

b. Submit your source codes in a single compressed file "ID_A1.zip" to iCampus (Do not include executable files).

c. G++ compiler will be used to compile your submitted codes.

Compile Command: "`g++ YOUR_SOURCE_CODE.cpp -O2 -std=c++11`"

d. Your program will be tested on Ubuntu 16.04.

e. Your program will get two arguments, input and output file names, from the command line as follows:

```
>> Your_Executable   INPUT_FILE_NAME   OUTPUT_FILE_NAME
```

For instance:

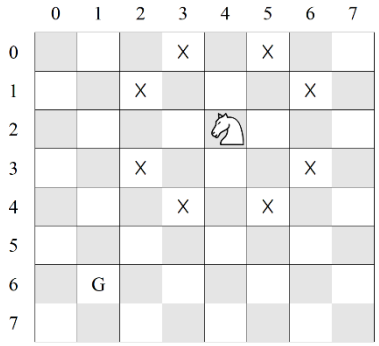
```
>> a.out  input_1.txt  output_5.txt
```

f. You are not allowed to declare any global variable in your codes.

g. Any work that you turn in should be your own. Copy detection will be seriously performed.

Problem #1. (Programming) Knight (35 pts)

Given a squared chess board, find the minimum number of steps taken by a Knight to reach desired destination from its given source position. As illustrated in the right figure, Knight can move to 8 different locations by a single step.



For instance, we need at least 3 steps to move the Knight at (2,4) to the goal position (6,1). Specifically, the Knight can move the following path: (2,4) → (3,2) → (4,0) → (6,1). Note that, locations are represented by (vertical, horizontal) coordinates and left-upper corner is (0,0) similar to the array in C/C++.

Sample input	Sample output
8 2 4 6 1	3

The first value in the input file indicates the chess board size N ($4 \leq N \leq 20$). The second and third lines are the source and destination locations, repsectively. The output file only contains the minimum number of steps from the source to the destination. You can output "-1", if the destination is unreachable. Your program has to produce an answer within 5 seconds.

You are asked to name your source code file "ID_ Knight.cpp", and you have to use `std::queue` to solve the problem.

Problem #2. (Programming) Calculator (65 pts)

Calculate a given expression consisting of natural numbers less than 999 (1, 2, 3, ... , 999), binary operators (+, -, *, /), and parentheses. Provide the result to the second decimal place by rounding off (e.g., 5/3 = 1.67).

Refer the following example:

Sample input	Sample output
52+3*219	709.00

Another example:

Sample input	Sample output
((51+3)*2)+(255-500/3)	196.33

Note that, the length of input expression does not exceed 100, and your program has to produce an answer within 5 seconds.

You are asked to name your source code file "ID_Calc.cpp", and you have to use `std::stack` to solve the problem.