物件導向程式設計 期中考

Object-Oriented Programming Midterm exam

1.1

請寫出一個程式讓使用者可以輸入兩個數字*N*、*M*，印出矩陣大小為*N**N*，且方向為*M*的旋轉矩陣。(M=0：順時針，*M=1*：逆時針)

Your program is required to take as input two numbers *N* and *M*. Please print a matrix of the size of *N**N* with the circular arrangement of numbers from *1* to *N**N*. The notation *M* determines the direction of circular arrangement. (*M=0*: clockwise，*M=1*: counter clockwise)

以下是範例

Two examples are shown below.

Input	**Output**

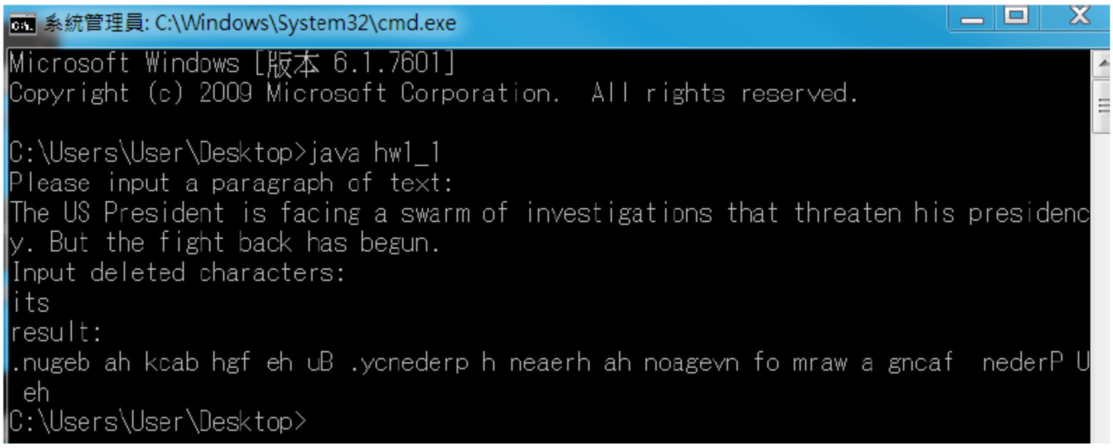
N=3, *M*=0	1 2 3
	8 9 4
	7 6 5

N=4, *M*=1	1 12 11 10
	2 13 16 9
	3 14 15 8
	4 5 6 7

1.2

請寫出程式讓使用者能夠輸入一段文字與要刪除的字元，此程式可以輸出反向字串並且不會顯示要刪除的字元。(大小寫視為同一種)

Please write a program that allows the user to input a paragraph of text and to delete characters. Your program is required to output a paragraph of text in a reverse direction, where the characters being deleted are replaced by empty characters. (The lower and upper cases are treated as the same.)



```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\User\Desktop>java hw1_1
Please input a paragraph of text:
The US President is facing a swarm of investigations that threaten his presidenc
y. But the fight back has begun.
Input deleted characters:
its
result:
.nugeb ah kcab hgf eh uB .ycnederp h neaerh ah noagevn fo mraw a gncaf  nederP U
 eh
C:\Users\User\Desktop>
```

Input (paragraph of text)	Input (deleted characters)	Output
TTttasdttTT	t	dsa
Qq. Ww. Ee. Rr. Tt. Yy. Uu.	E.r	uU yY tT wW qQ
1 2 3 4 5 6 7 8 9	4 6	9875321

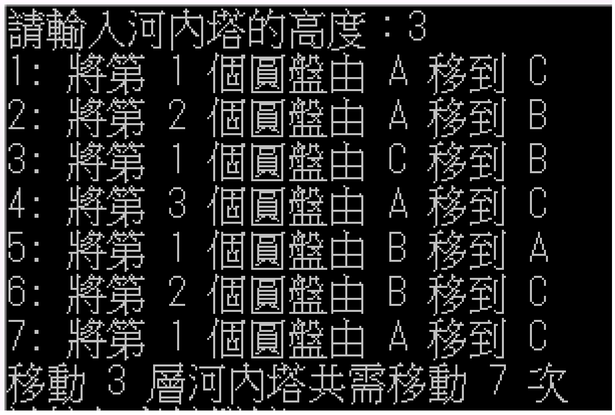
2.

請修改以下程式讓使用者可以輸入一個小於 31 的正整數 *N*，並根據程式碼需求在 main 中使用 class *Teacher* 的函數 *show()*輸出一個字串為第 ***N*** 次移動 (*N* 為使用者輸入的正整數參數，須設置給 class *Teacher* 內的 *number*) **的目的為 ”A/B/C 哪一個” 杆子**。

class Student 中的函數 hanoi(n, A, B, C) 此 n 初始值為 5 且並非為使用者輸入的正整數參數 *N*，其定義如下：
有三根杆子 A、B、C。A 杆上有 n 個穿孔圓盤，盤的尺寸由下到上依次變小。要求按下列兩條規則將所有圓盤移至 C 杆。請將第 ***N*** 次移動 **(*N* 為使用者輸入的正整數參數) 的目的地杆子**傳給 class *Teacher* 的字元成員。移動的規則如下：

- 每次只能移動一個圓盤。
- 大盤不能疊在小盤上面。

以下為 n=3 的 hanoi(n, A, B, C)範例。



```
請輸入河內塔的高度：3
1: 將第 1 個圓盤由 A 移到 C
2: 將第 2 個圓盤由 A 移到 B
3: 將第 1 個圓盤由 C 移到 B
4: 將第 3 個圓盤由 A 移到 C
5: 將第 1 個圓盤由 B 移到 A
6: 將第 2 個圓盤由 B 移到 C
7: 將第 1 個圓盤由 A 移到 C
移動 3 層河內塔共需移動 7 次
```

Input (N)	Output
5	第 5 次移動的目的地為 A 杆子
6	第 6 次移動的目的地為 C 杆子

Your program is required to take as input a positive integer *N* which is less than 31, and use the *show*() function to output a string **“the destination of the N^{th} move is “A or B or C”-rod**. *N* must be set to the *number* of the class *teacher*.

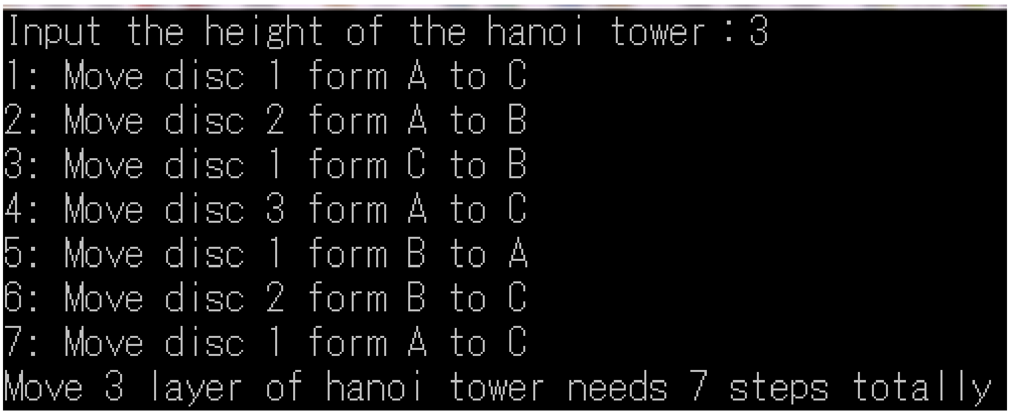
The definition of the hanoi(n,A,B,C) is shown below **(The initial value of n is set to 5 and it's not a positive integer parameter *N* entered by the user)** :

There are three rods A, B, and C. There are n perforated discs on the A-rod, and the size of the disc becomes smaller from the bottom to top. It is required to move all the discs from the A-rod to the C-rod according to the following rules.

- Only one disc can be moved at a time.
- A larger disc may not be placed on top of a smaller disc.

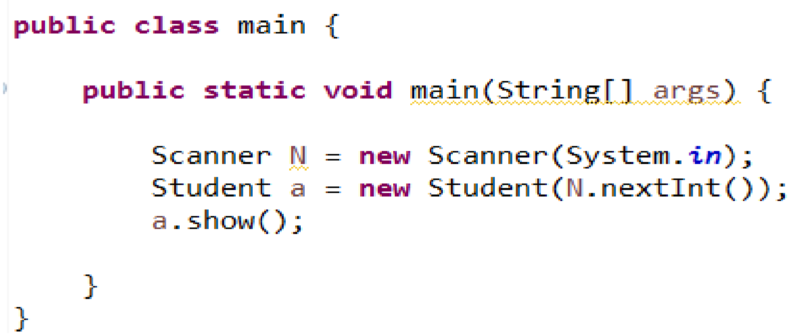
Please assign the destination rode of N^{th} move (*N* is the positive integer entered by the user) to the character member of the class *Teacher.*

The following is an example of hanoi(n,A,B,C) with n=3 .

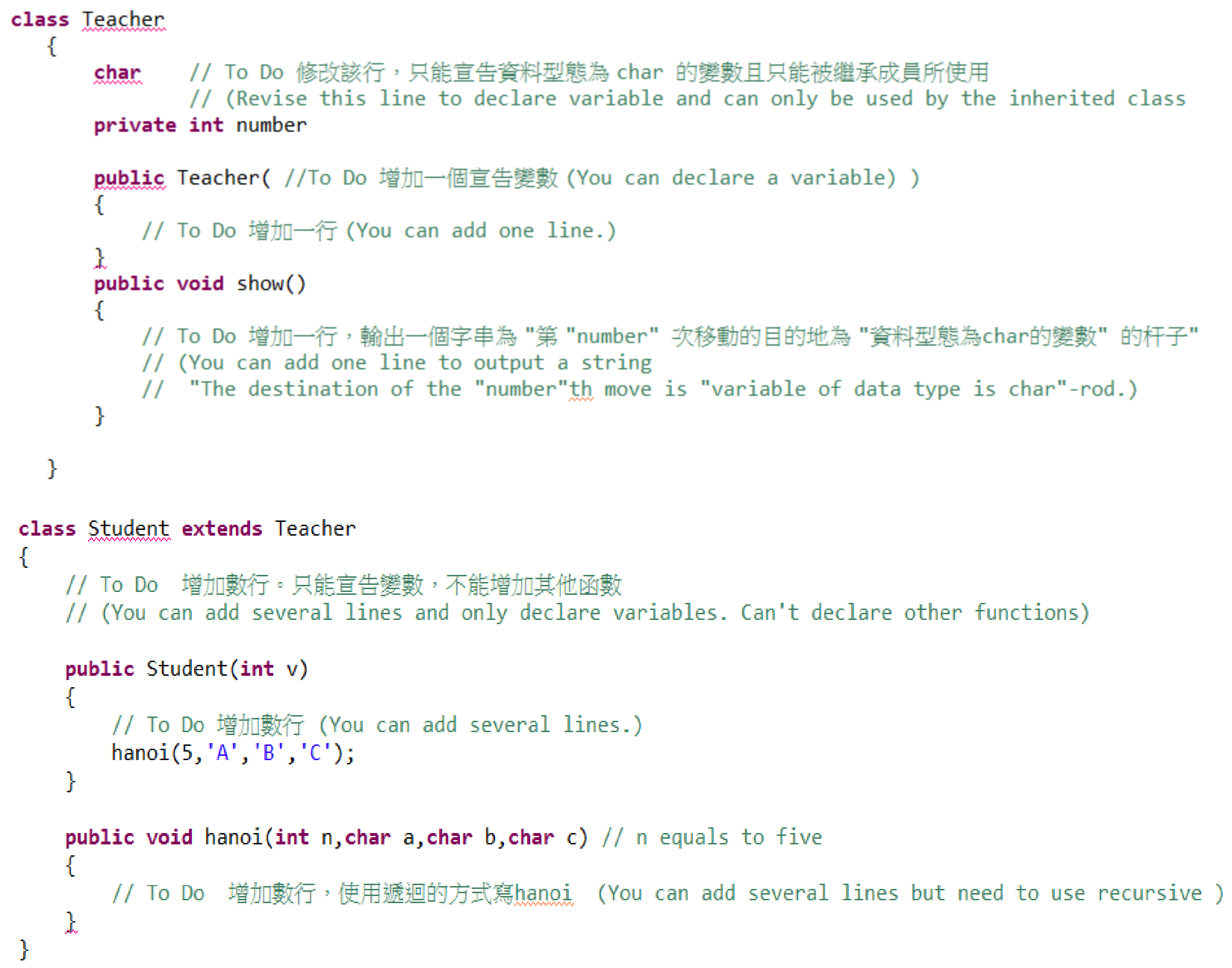


```
Input the height of the hanoi tower : 3
1: Move disc 1 form A to C
2: Move disc 2 form A to B
3: Move disc 1 form C to B
4: Move disc 3 form A to C
5: Move disc 1 form B to A
6: Move disc 2 form B to C
7: Move disc 1 form A to C
Move 3 layer of hanoi tower needs 7 steps totally
```

Input (N)	**Output**
5	The destination of the 5th move is A-rod.
6	The destination of the 6th move is C-rod.



```java
public class main {

    public static void main(String[] args) {

        Scanner N = new Scanner(System.in);
        Student a = new Student(N.nextInt());
        a.show();

    }
}
```



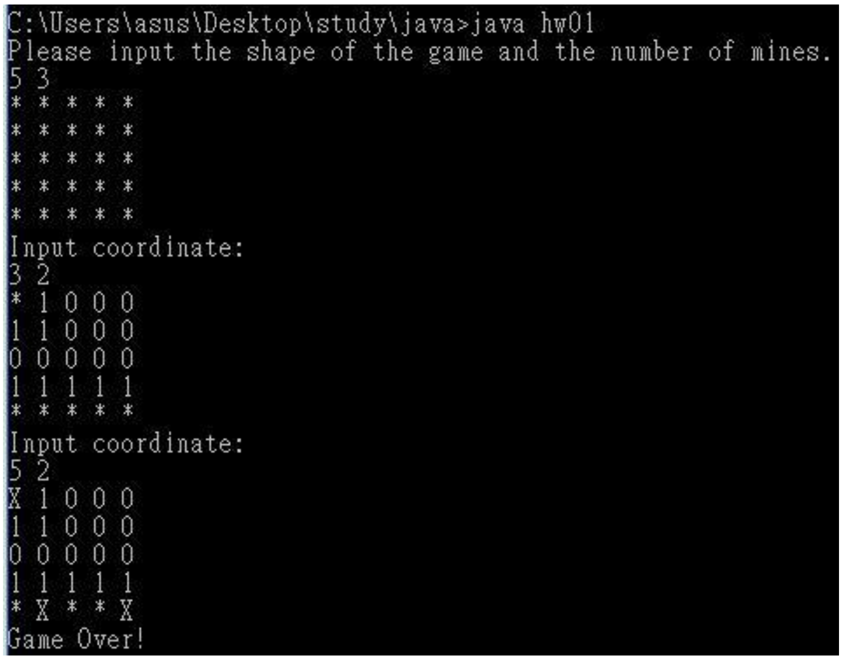
```
class Teacher
   {
      char    // To Do 修改該行，只能宣告資料型態為 char 的變數且只能被繼承成員所使用
              // (Revise this line to declare variable and can only be used by the inherited class
      private int number

      public Teacher( //To Do 增加一個宣告變數 (You can declare a variable) )
      {
          // To Do 增加一行 (You can add one line.)
      }
      public void show()
      {
          // To Do 增加一行，輸出一個字串為 "第 "number" 次移動的目的地為 "資料型態為char的變數" 的杆子"
          // (You can add one line to output a string
          //  "The destination of the "number"th move is "variable of data type is char"-rod.)
      }

   }

 class Student extends Teacher
 {
     // To Do  增加數行。只能宣告變數，不能增加其他函數
     // (You can add several lines and only declare variables. Can't declare other functions)

     public Student(int v)
     {
         // To Do 增加數行 (You can add several lines.)
         hanoi(5,'A','B','C');
     }

     public void hanoi(int n,char a,char b,char c) // n equals to five
     {
         // To Do  增加數行，使用遞迴的方式寫hanoi  (You can add several lines but need to use recursive )
     }
 }
```

3.

請實現一個簡單的踩地雷遊戲. 請輸入遊戲格數與地雷數量，其中地雷位置為隨機產生，此程式可以玩遊戲直到結束，如下圖所示。

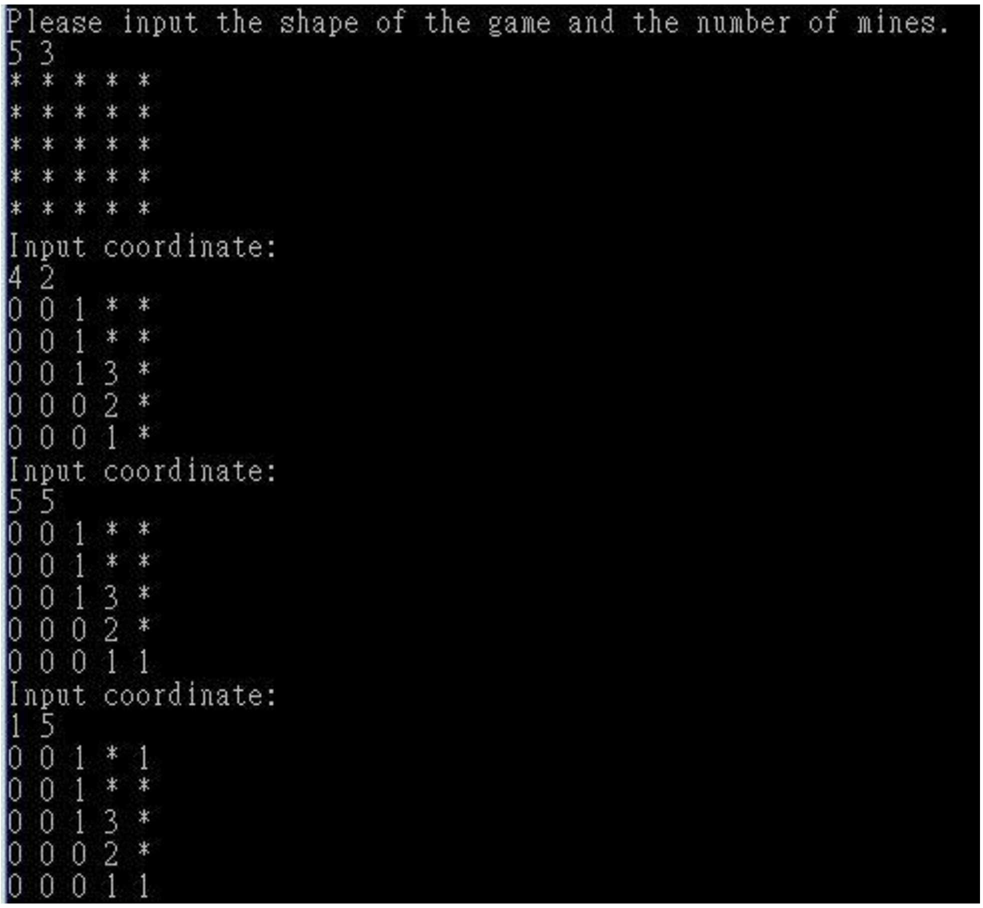
You are required to write a simple Minesweeper game. Please input the shape and size of the chessboard and the number of mines. The locations of the mines are randomly generated. Your program is required to be able to play the game until it's over. As shown in below.

(Example: Game Over)

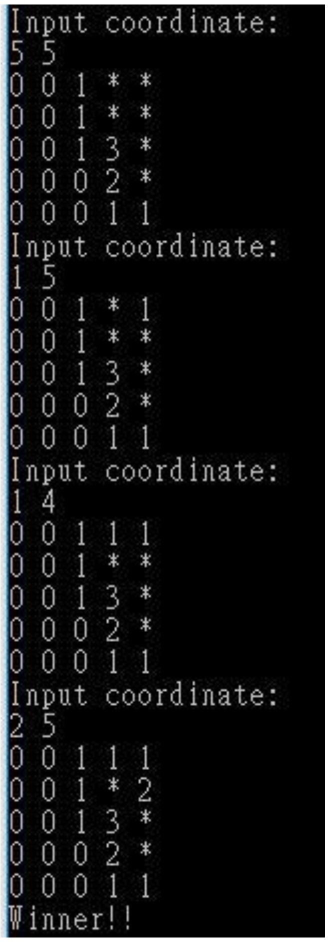


```
C:\Users\asus\Desktop\study\java>java hw01
Please input the shape of the game and the number of mines.
5 3
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Input coordinate:
3 2
* 1 0 0 0
1 1 0 0 0
0 0 0 0 0
1 1 1 1 1
* * * * *
Input coordinate:
5 2
X 1 0 0 0
1 1 0 0 0
0 0 0 0 0
1 1 1 1 1
* X * * X
Game Over!
```

(Example: Win)



```
Please input the shape of the game and the number of mines.
5 3
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
Input coordinate:
4 2
0 0 1 * *
0 0 1 * *
0 0 1 3 *
0 0 0 2 *
0 0 0 1 *
Input coordinate:
5 5
0 0 1 * *
0 0 1 * *
0 0 1 3 *
0 0 0 2 *
0 0 0 1 1
Input coordinate:
1 5
0 0 1 * 1
0 0 1 * *
0 0 1 3 *
0 0 0 2 *
0 0 0 1 1
```



```
Input coordinate:
5 5
0 0 1 * *
0 0 1 * *
0 0 1 3 *
0 0 0 2 *
0 0 0 1 1
Input coordinate:
1 5
0 0 1 * 1
0 0 1 * *
0 0 1 3 *
0 0 0 2 *
0 0 0 1 1
Input coordinate:
1 4
0 0 1 1 1
0 0 1 * *
0 0 1 3 *
0 0 0 2 *
0 0 0 1 1
Input coordinate:
2 5
0 0 1 1 1
0 0 1 * 2
0 0 1 3 *
0 0 0 2 *
0 0 0 1 1
Winner!!
```