CS2312 Problem Solving and Programming [2014-15 Semester B]
Quiz 1

Write your answers for Q1-3 in **pages 3-5**.
Pay attention to **code design, naming and code formatting**.
Program execution speed is not a major concern.

Duration: 40 minutes. Total mark: 40 marks +0.1 **for Q3** (partial mark up to 0.05 may be given for partial correctness / improper style ☺)

Q1. Searching in a 2D array [20 marks]

Complete the given method, `search`, which searches for a wanted value in a 10x10 2D array.

The search should be done by rows; ie. start from first row (ie row 0) , from column 0 to column 9; then row 1, row 2, .. etc..

If the value is not found, output "not found".

Otherwise, only output the first location where it is found, as shown in the sample rundown.

Sample rundown (See `main()` in P.3):
(underlined contents are input by the user)

```
   2    6   10   14   18   22    1    5    9   13
  15   20    0    5   10   15   20    0    5   10
   3    9   15   21    2    8   14   20    1    7
  16   23    5   12   19    1    8   15   22    4
   4   12   20    3   11   19    2   10   18    1
  17    1   10   19    3   12   21    5   14   23
   5   15    0   10   20    5   15    0   10   20
  18    4   15    1   12   23    9   20    6   17
   6   18    5   17    4   16    3   15    2   14
  19    7   20    8   21    9   22   10   23   11
Enter a value to search: 5
First location:  (row 0, col 7)
```

Q2. Array of Days [20 marks]

Add required code to complete the given program which lists all days in a month, as shown in the example:

Note/Hint:
Your solution should involve at least one method in the `Day` class.

Sample rundown (underlined contents are input by the user)

```
Enter year and month (eg. "2013 12"): 2012 2

There are 29 days in the month:
1 Feb 2012
2 Feb 2012
3 Feb 2012
4 Feb 2012
5 Feb 2012
6 Feb 2012
7 Feb 2012
8 Feb 2012
9 Feb 2012
10 Feb 2012
11 Feb 2012
12 Feb 2012
13 Feb 2012
14 Feb 2012
15 Feb 2012
16 Feb 2012
17 Feb 2012
18 Feb 2012
19 Feb 2012
20 Feb 2012
21 Feb 2012
22 Feb 2012
23 Feb 2012
24 Feb 2012
25 Feb 2012
26 Feb 2012
27 Feb 2012
28 Feb 2012
29 Feb 2012
```

Q3. *Given in page 5* [+0.1 mark]

[Blank page for your sketching]

[Q1]

```
                                                                Main.java
..

   public static void search(int [][] arr, int wanted)
   {
                                                               (20 Marks)













   }


   public static void main(String[] args)
   {
      int arr[][];
      .. //Set up the 10x10 2D array and its contents [You do not need to do this part]

      .. //Display the 2D array  [You do not need to do this part]

      //Search
      Scanner in = new Scanner(System.in);
      System.out.print("Enter a value to search: ");
      int x = in.nextInt();
      search(arr,x);
      in.close();
   }
}
```

[Q2]

Main.java

```
public static void main(String[] args)
{
   System.out.print("Enter year and month (eg. \"2013 12\"): ");
   Scanner scannerObj = new Scanner(System.in);
   int y=scannerObj.nextInt();
   int m=scannerObj.nextInt();

   Day[] dArray;
```

(2 Marks)

```
   System.out.printf("\nThere are %d days in the month:\n",dArray.length);
   for (int i=0;i<dArray.length;i++)
        System.out.println(dArray[i]);
   scannerObj.close();
}
```

Day.java

```
public class Day {

    private int year, month, day;

    public Day(int y, int m, int d) {..} // Constructor

    static public boolean valid(int y, int m, int d) {..} // check if y,m,d valid

    public String toString() {..} // Return a string for the day like dd MMM yyyy

    public Day next() {..} // create and return the "next day" of the current day object


}
```

(18 Marks)

[Q3] [0.1 mark; (partial mark up to 0.05 may be given for partial correctness / improper style ☺)]

Rewrite your method for Q1 such that if the wanted value appears 2 times or more, it clearly shows the first and last locations where the wanted value is found.

Reminder: If the value is not found, output "not found".

Sample rundown #1 (only one location)

```
…
Enter a value to search: 13
Found location:  (row 0, col 9)
```

Sample rundown #2 (2 locations)

```
…
Enter a value to search: 7
Found locations:  (row 2, col 9) and (row 9, col 1)
```

Sample rundown #3 (more than 2 locations)

```
…
Enter a value to search: 22
Found locations:  (row 0, col 5) .. (row 9, col 6)
```

Main.java

```
..

   public static void search(int [][] arr, int wanted)
   {














   }
..
```

Total: ____ / 40