CMPT 120, Spring 2015, SFU Burnaby
Instructor: Diana Cukierman
Assignment #5: "The diamond treasure hunter"

This is an up to 3 people team work (you can also work in a team of 2 people or individually if you prefer so). While you may discuss generalities with colleagues from other teams about this exercise, you cannot develop the same code, nor share code among teams, nor obtain code from other sources.

Being a team exercise it places a big responsibility on each individual. You want to respect and be honest with your partner and with yourself: DO YOUR SHARE, and BE KNOWLEADGEABLE OF THE WHOLE ASSIGNMENT. Working on this assignment is also a way for you to get prepared for the final exam.

START WORKING ON THIS AS SOON AS POSSIBLE

Submission deadlines:

I) ~~*Friday April 10, 8:00 PM*~~: *Monday April 13, 8:00 PM First part: peerwise exercises associated to the assignment (and which may be used in the final code, identical or variations)*

II) ~~*Monday April 13, 9:00 AM*~~. *Thursday April 16, ~~8:00~~ 11:59 PM Final deadline*. ~~*This is 1/2 hour before the last class this semester*~~. *You certainly can submit before the deadline.*

General description -- READ THE WHOLE DESCRIPTION

1. The project consists of implementing a game which we will call "The diamond treasure hunter"

2. The game is played on a square board (square matrix) with integer numbers in each board cell, where the board represents a diamond mine with tunnels and special passages and the numbers represent the number of diamonds in each cell location. Each time a new game starts the user provides a name for a treasure hunter and values to create a board. Each game may allow a hunter do many 'treasure trips'. Several games (with different hunters each) may be played until the user does not want to play anymore.

3. One game consists on having one hunter travelling through the mine (board) (following various possible ways of traversing the board) . As the hunter visits the cells he collects diamonds (the hunter will be referred as a him, but it could clearly be a woman hunter). It is possible that the hunter does several trips (until the user decides not to have that hunter travel anymore) or it may be that the hunter gets trapped and cannot travel anymore. If the hunter is trapped that game with that hunter is stopped, but the user may later play with other hunters and boards.

4. In more detail, **one game consists of:**

   a. Asking the user for the name of the hunter

   b. Asking the user for values to create a board (more details below)

Asking the user for one travelling 'modality' . For example, a travelling modality can be "visit all the board cells in one row" or "visit randomly" (all the possible travelling modalities are explained below).

   c. The hunter then will travel according to the modality.

d. When the hunter reaches a position where there is a number (of diamonds), the hunter will collect those diamonds, although maybe not all. **The hunter will collect half of what is available in the cell if the number of diamonds is even or all the diamonds in the cell has an odd number of diamonds**. When the hunter visits a position where there is a 0, the hunter does not get to collect any diamonds/ points. **If the hunter reaches the cell with -1, the hunter gets trapped.**

e. If the hunter is trapped the game is interrupted (for that hunter) but the hunter keeps his points.

f. If the hunter is not trapped, the hunter can keep travelling, in possibly other modalities as determined by the user, collecting more diamonds/points (until the user does not want to keep playing with that hunter or until the hunter gets trapped.)

g. If the user does not want to play anymore with the same board and same hunter, the user indicates so, and the total points that the hunter got as well as a 'lucky number' associated to that game is shown to the user. (The calculation of the lucky number is explained below)

h. After one game is over and the lucky number was shown, if the user decides to keep on playing, for the next game there will be a new hunter and a new board (with diamonds generated again).

5. At the end of all the games (when the user does not want to play anymore with new boards and new hunters) the user must be informed several results and totals (see below).

Creation of the board and hunter for one game

6. The user is asked to provide a name for the hunter (any string is ok)

7. The user is asked to provide a number between 3 and 6 (including extremes)

8. Validate **that the value that the user provides is an integer and that the value is in such range**). This value will be used to create the board, the number will be the dimension of the board `dim`. The board will be a square matrix of size `dim x dim`.

9. There are **two modalities to fill the board with numbers**.

   a. **Fixed**: A board may be provided by the user in one input only, with a list of lists, (each row being a list of integer numbers and possibly including a -1) . No validation is required, just allow the user to type the whole list representing the whole matrix.See sample run.

   b. ADDITIONAL **- A board may be generated randomly**. Additionally to the dim number, the user is asked for another integer number, `maxdiam`, between 1 and 10. This is the maximum diamonds to be included in each cell. Initially, the contents of each cell in that matrix will be a random number between 0 and `maxdiam`, extremes included . After the board is initially populated with these numbers, one particular cell may be given the value -1 (it may be the case that there is no cell with value -1) . You should generate -1 in one cell with 80% chance.

CMPT 120, Spring 2015, SFU Burnaby
Instructor: Diana Cukierman
Assignment #5: "The diamond treasure hunter"

10. **<u>As the game unfolds and the hunter collects diamonds</u>**, **<u>the number of diamonds in each cell will change.</u>**

<u>Modalities for hunter trips</u>

11. Once the board is defined it is shown to the user.

12. Then the user is asked to choose one 'hunter travelling modality'. (Each time that the user is asked a travelling modality the board should be shown again to the user) . The following travelling modalities are possible:
    a. Visit all the cells in a row (the row number has to be provided by the user) , where the first row is 0
    b. Visit all the cells in a column (the column number has to be provided by the user), where the first column is 0
    c. Visit all the cells in the main diagonal
    d. Visit all the cells in the secondary diagonal
    e. ADDITIONAL Visit cells randomly (not more than `dim x dim`) (while the hunter travels there is <u>NO need to check</u> if a cell is visited more than once)

1	2	3
4	5	6
7	8	9

If a board has the values as in this matrix, the travelling modalities would encounter the values as follows:

All cells in row 0: 1,2,3
All cells in column: 0: 1,4,7
Cells in main diagonal: 1,5,9
Cells in secondary diagonal: 3,5,7

13. After a travelling modality is given by the user, the hunter starts travelling and collecting diamonds as defined above (and of course there will be less or no diamonds remaining in that cell after he visited).

14. After the hunter visited all cells the program should show the positions that was just visited (this is useful for debugging also)

15. If the hunter visits the cell with the value -1 then the hunter informs that it got trapped and the game is over (for that hunter) . The user can choose to keep playing with a new hunter.

16. (For better testing use different names for the different treasure hunters)

17. When the hunter finishes visiting the board according to the modality given, a new travelling modality can be provided for the same board and same hunter (unless the hunter is trapped). (at this stage the board should be shown to the user again)

Example:
Assume that the user indicates that the hunter name is "Jones" and `dim` is 4
Assume that the following board is generated (or input by the user).

CMPT 120, Spring 2015, SFU Burnaby
Instructor: Diana Cukierman
Assignment #5: "The diamond treasure hunter"

```
5   4   0   2
0   8   1   6
0   0   0  -1
3   3   4   7
```

The user then chooses the following travelling modalities:

a. For example, if the first travelling modality is "all cells in row 1", then Jones will travel as follows:
   [1,0] [1,1] [1,2] [1,3] accumulating 0 + 4 + 1 + 3 = 8 diamonds/points. (values 8 and 6 are even, so Jones collects half the diamonds in each of those cells as described above, value1 is odd, he collects the 1 diamond in that cell)

The board now will look like: (the numbers that changed are underlined just for the sake of the example)

```
5   4   0   2
0   4   0   3
0   0   0  -1
3   3   4   7
```

b. Assume the next travelling modality is " secondary diagonal", then Jones will travel as follows:
   [0.3] [1.2] [2.1] 3,0] and the total points collected during this trip will be 1 + 0 + 0 + 3 = 4 points, and the board will become:

```
5   4   0   1
0   4   0   3
0   0   0  -1
0   3   4   7
```

c. Assume the next travelling modality is "all cells in column 3", then Jones will travel as follows:
   [0,3] [1,3] [2,3] -- but because Jones gets trapped he does not keep travelling, and the total points for this trip is 1 + 3 = 4, and the last state of the board is

```
5   4   0   0
0   4   0   0
0   0   0  -1
0   3   4   7
```

d. Now Jones is trapped and cannot travel anymore; the total points/diamond that Jones collected is 8+4+ 4=16 during his three trips.

e. Before allowing the user to continue playing with another board and hunter, the lucky number associated to the last state of the board is calculated and shown to the user (see below)

f. Now the user can choose to play another game (with another board and another hunter) or finish playing.

CMPT 120, Spring 2015, SFU Burnaby
Instructor: Diana Cukierman
Assignment #5: "The diamond treasure hunter"

Lucky number associated to the board

18. After the game is over (i.e. all the travelling with one board and one hunter are over) a lucky number associated to the board must be calculated and shown to the user. To do this calculation each row of the board is considered to generate a binary number as follows: if the number is odd, it stands for a 1 otherwise it stands for a 0. Then, each row (binary number) should be converted to base 10 (if there is a -1 in some position consider it to be 1 for this calculation). Then the resulting values for each row are added, providing the lucky number associated to the board .

    For example, the board at the end of the previous game produces the following binary numbers per row: 8,0,1,5, (e.g. the first row, 5 4 0 0 results in the binary number 1 0 0 0 , which converted to base 10 is 8: $1000_{(2} = 8_{(10}$). Recall that the -1 is interpreted as a 1. Then adding all rows, the lucky number is 8 + 0 + 1 + 5 = 14:

5	4	0	0		1	0	0	0	→ 8
0	4	0	0		0	0	0	0	→ 0
0	0	0	−1		0	0	0	1	→ 1
0	3	4	7		0	1	0	1	→ 5

Totals and results at the end of all the games

19. When all the games are over (the user does not want to play any more) the program should inform :
    a. how many hunters played.

    b. What the total points of all the hunters together were (regardless of whether they were trapped or not)

    c. Which hunter (the name) got the maximum diamonds/points among all the hunters and how many points those were (even if the hunter ended up trapped).

    d. ADDITIONAL The program must also inform if the hunter who got the maximum points had became trapped.

    e. ADDITIONAL The program must also show the minimum lucky number among all the boards.

REQUIREMENTS

A. Your solution should be modular. That is, your code should include a top level calling functions and functions may be calling other functions (or just return to the top level), including passage of parameters, returning of values, etc.

B. You should have at least 5 functions.

C. You should have at least one function receiving parameters

D. You should have at least one function returning a Boolean value

E. You should have at least one function returning some value (other than Boolean)

F. You should have at least one non-productive function

G. Represent the board as a list of lists, where each row is a list and each element in a row is an integer between 0 and `10 inclusive`. See sample runs.

H. The board has to be printed with one row per line, similar to what is shown in the sample run

HINTS!!! Incorporating features to the program gradually

I. Make sure that you understand the game well. Follow the sample runs.

J. Arrange with your team mates how you will distribute tour work. Make sure that you explain to each other what you learn

K. You can use variables, lists, strings, and any methods associated to them. You can use for loops, while loops, functions as indicated or more.

.

L. For the general top level you may want to get inspired in a flowchart done in class with two nested while loops, event controlled.

M. Creating some intermediate lists may be useful to you, for example you may want to create some intermediate list with the positions the hunter will be visiting . Since the matrix has a row and a column you may want to do this with two lists, a list with row numbers of the positions visited and a list with column numbers .

N. Develop your code gradually: Some ideas to develop gradually:

   a. Implement only one game first, and after one game works implement many games, and keep track of the totals or of special variables to hold the maximum values, etc.
   b. Develop some concrete functions that can be developed independently (for example, given a list with 0's and 1's treat it as a binary number, with position 0 being the most significant bit, and convert to base 10) (This function in particular is a Peerwise question.)
   c. First ask values to the user without any validation and then you may add some validation gradually.
   d. First implement only one travel modality, and as you debug your program then add more travel modalities.
   e. First do not include a -1 in your board, and once the code works incorporate it
   f. Leave the requirements indicated as ADDITIONAL to implement later. This will have bonus points instead of regular points.

CMPT 120, Spring 2015, SFU Burnaby
Instructor: Diana Cukierman
Assignment #5: "The diamond treasure hunter"

What to submit (all compressed)

1. Code – with authors names at the top, number of hours, good naming conventions
2. Brief description of work distribution in the team
3. If needed, Brief description of any additional assumption

Sample runs are provided. Your program does not have to have the exact look of the sample runs, but all the information present in the sample runs should be present in your solution.

If there are additional posts with clarifications, these will be announced. If you have any questions, please contact the teaching staff.

End of description of "Assignment #5".