Computer Science Exam
SCAN 1st - Décembre 2020

Toal duration: *1h*
Documents: *None*
Smartphones are not allowed.

- The graduation can still change.
- The assignment is on 7 pages.
- All the questions are independent. If it is necessary for A to be solved to solve B, then you can do as if you had solved A (and clearly indicate your assumption) to solve B.

Thought of the day:
« Always code as if the guy who ends up maintaining your code will be a violent psychopath who knows where you live. » *Martin Golding*

Indicative grading scale :

- Exercise 1 : 3 points
- Exercise 2 : 3 points
- Exercise 3 : 4 points
- Exercise 4 : 3 points
- Exercise 5 : 7 points

1 Code correction

```
public class Ex1 {
    public static void main (String[] args) {
        // Initialisation
        double size = 40;
        int iteration = 15
        boolean[] cell = new boolean[size];
        for(int i = 0; i < cell.length; i++)
            cell[i] = false;
        }
        cell[size/2] = 1;

        // Display
        Display.displayWorld(cell);

        for(int j = 1; j < iteration; j++){
            // Creation of a new array
            boolean r = new boolean[cell.length];
            for(int i = 0; i < cell.length-1; i++){
                if(cell[i-1] == false && cell[i] == false && cell[i+1] == false) {
                    r[i] = false;
                } else if (cell[i-1] == true && cell[i] == true && cell[i+1] == true) {
```

```java
                    r[i] = false;
                } else {
                    r[i] = true;
                }
            }
            // Modification
            cell = r;
            // Display
            Display.displayWorld(cell);
        }
    }
}
```

(Q1.1) Provide a correction for each of those errors and briefly justify your choice.

Answer :

```java
public class Ex1Correction {
    public static void main (String[] args) {
        // Initialisation
        int size = 40; // err1 : size of an array therefore int
        int iteration = 15; // err2 : missing ;
        boolean[] cell = new boolean[size];
        for(int i = 0; i < cell.length; i++){ // err3 : missing
            cell[i] = false;
        }
        cell[size/2] = true;// err4 : conversion not possible

        // Display
        Display.displayWorld(cell);

        for(int j = 1; j < iteration; j++){
            boolean[] r = new boolean[cell.length]; // err5 : incorrect array declaration
            for(int i = 1; i < cell.length-1; i++){ // err6 : i should start at 1 otherwise pbm with i-1
                if(cell[i-1] == false && cell[i] == false && cell[i+1] == false) {
                    r[i] = false;
                } else if (cell[i-1] == true && cell[i] == true && cell[i+1] == true) {
                    r[i] = false;
                } else {
                    r[i] = true;
                }
            }
            // Modification
            cell = r;
            // Affichage
            Display.displayWorld(cell);
        }
    }
}
```

2 Code understanding 1

(Q2.1) What is displayed on the terminal when executing the following class `Ex2`?

```java
public class Ex2 {
    public static void main(String[] args) {
        int n = 20;
        int[] tab = new int[n];
        int r = (int)(Math.sqrt(n));
        int p;
        for (int i = 0; i < n; i++){
            tab[i]=i;
        }
        for (int i = 2; i <= r; i++) {
            p=tab[i];
            if (p>0){
                for (int j = i + p; j < tab.length; j += p) {
                    tab[j] = 0;
                }
            }
        }
        for (int i = 2; i < tab.length; i++) {
            if (tab[i]>0) {
                System.out.println(tab[i]);
            }
        }
    }
}
```

Answer : Prime numbers from 2 to 19

3 Code understanding 2

(Q3.1) What is displayed on the terminal when executing the following class Ex3 and replacing the ..... by 5, then 164, and 2442 ?

```java
public class Ex3 {
   public static void main(String[] args) {
      int n = ..... ;
      int s = 0;
      int temp = n;
      int r;
      while(temp > 0) {
         r = temp % 10;
         s = (s * 10) + r;
         temp = temp / 10;
      }
      if(n == s) {
         System.out.println("True");
      } else {
         System.out.println("False");
      }
   }
}
```

Answer : True, False, True

(Q3.2) What is the class Ex3 doing ?

Answer : Testing if a number is a palindrome and displaying the answer

4 Where is the rabbit ?

You will program a small game in which the player needs to find a rabbit hidden in a field. The field is represented by a square grid of size 10 by 10 in which the rabbit is hidden at coordinates (x_L, y_L) selected randomly at the beginning of the game. At each turn, the player selects a cell by providing two integers (x, y) designating the cell. The game answers by printing the euclidean distance between this cell and the cell of the rabbit. The game ends when the player finds the rabbit.

Remark : The euclidean distance d between two points $\text{A}(x_A, y_A)$ and $\text{B}(x_B, y_B)$ is given by :

$$d = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

Remark : the input of the two coordinates can be done using the following instruction : `int x = sc.nextInt();`.

(Q4.1) Write a program implementing this game and counting the number of turns that have been necessary to find the rabbit.

Some useful java instructions :

- `double a = Math.sqrt(b)` : computes the square root of `b`
- `double a = Math.pow(b,c)` : computes b^c
- `double a = Math.random()` : generates a random number in the interval $[0, 1[$

Answer :

```
import java.util.Scanner;
public class Lapin {
   public static void main (String[] args) {
       // INITIALISATION
      Scanner sc = new Scanner (System.in);
      int i = 1;
      int size = 10;
      int xL  = (int)(Math.random()*size + 1);
      int yL  = (int)(Math.random()*taille + 1);
      boolean encore = true;

      while (encore){
          // Player Input
         System.out.println("Turn "+i+", your move:");
         System.out.print("X = ");
         int xA  = sc.nextInt();
         System.out.print("Y = ");
         int yA  = sc.nextInt();

         // Computing the distance
         double d = Math.sqrt((xL-xA)*(xL-xA)+(yL-yA)*(yL-yA));

         // Should we continue ?
         if(d == 0){
            System.out.println("You won !");
            encore = false ;
         } else {
            System.out.println("d = "+d);
            i++;
         }
      }
   }
}
```

5 Secrets of the lottery

Let us consider the following mysterious sequence *lotocode* composed of 50 integer values :

index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	29	21	22	23	24
value	23	23	18	28	1	10	5	1	11	16	6	19	40	34	30	42	25	27	18	34	5	3	41	6	2
index	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49
value	10	33	39	48	32	42	0	15	19	0	5	9	3	7	48	47	46	8	16	46	44	2	3	13	49

(Q5.1) Write the line of code that declares and initialises an array containing these values (no need to put all the values ; e.g. 23 23 .... 13 49)

Answer :

```
   int[] lotocode = { 23,23,18, .. };
```

The sequence `lotocode` can be decoded using a key. This key is an integer representing the starting position in the sequence. Positions are represented starting from 0 as in Java arrays. At the position indicated by the key, we find a value which represents the position of the next value in the sequence, and so on. This sequence ends with the value '0'.
For instance, if the key is 39, the sequence is decoded by looking at position 39 where we find 48, then by looking at position 48 where we find 13, then at position 13 where we find 34, then at position 34 where we find 0. The decoded sequence is thus (48,13,34).

(Q5.2) The results of tomorrow's lottery are encoded in that sequence with the key 14. What are the numbers of tomorrow's lottery ?

Réponse : 30,42,8,11,19,3

(Q5.3) Write a program that decodes the mysterious sequence given a key, and displays the result **on one line** with the values separated by a ','.

Answer :

```java
int cle=14;

while (lotocode[cle]>0) {
    cle=lotocode[cle];
    System.out.print(cle+",");
}
System.out.println();
```

(Q5.4) You are worried that others could get access to the lottery results. You want to alter the sequence by changing some of its values. Write a program that modifies the sequence `lotocode` in such a way that decoding with key 14 will give the following values.

```java
int[] fakeResults = { 31, 43, 9, 12, 20, 35 };
```

Answer :

```java
cle=14;
for (int i=0; i<fauxresultats.length; i++) {
    lotocode[cle] = fauxresultats[i];
    cle = fauxresultats[i];
}
lotocode[cle]=0;
```

There is a weakness in this mysterious encoding. If we search the zeros, we can backtrack the process and find the values in reverse order. For instance, at position 34 we can find a 0. We then search for the value 34 that we find at position 13. We continue by searching the value 13, etc. The value zero is thus an issue for the secrecy of the scheme. We need to find a way to get rid of those zeros.

(Q5.5) Suggest a solution to avoid using a stop value such as 0. Write the corresponding modification(s) that would have to be made in answers to the two previous questions.

Answer : by looping on the first value and detecting this loop.

```java
int cle=14;
int cleinitiale=cle;
```

```
while (lotocode[cle]!=cleinitiale) {
    cle=lotocode[cle];
    System.out.print(cle+",");
}
System.out.println();
```

```
cle=14;
cleinitiale=cle;
for (int i=0; i<fauxresultats.length; i++) {
    lotocode[cle] = fauxresultats[i];
    cle=solution[i];
}
lotocode[cle]=cleinitiale;
```