Computer Science Exam
SCAN 1st - January 2021

INSTITUT NATIONAL DES SCIENCES APPLIQUÉES LYON

Toal duration: *1.5h*
Documents: *None*
Smartphones are not allowed.

- — The graduation can still change.
- — The assignment is on 8 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 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*

1 Finding the errors (3pts)

Let us consider the class `Errors` that contains 6 errors.

```
public class Errors{
   public static void main(String args[]){
      int n = 10;
      double [] t = createArray(4);
      display(t);
   }
   public static double power(int x,int y){
      double p = 1.0;
      for(int i=1;i<=y;i++){
         p=p*x;
      }
      return p;
   }
   public static display(int [] a){
      for(int i=0;i<=a.length;i++){
         System.out.print(a(i));
      }
   }
   public static double [] createArray(int x){
      double [] t = new [x];
      if(t.length%2==0){
         t[0]= power(x,2);
         return t;
      }
   }
}
```

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

Answer :

```java
public class Errors{
   public static void main(String args[]){
      double [] t = createArray(4);
      display(t);
   }
   public static double power(int x,int y){
      double p = 1.0;
      for(int i=1;i<=y;i++){
         p=p*x;
      }
      return p;
   }
   public static void display(double [] a){ // missing void //incompatible parameter
       type
      for(int i=0;i<a.length;i++){ // i<a.length
         System.out.print(a[i]); // [] instead of ()
      }
   }
   public static double [] createArray(int x){
      int n = 10;
      double [] t = new double [n]; // missing type new int[n]
      if(t.length%2==0){
         t[0]= power(x,2);
          // return needs to be placed outside the if
      }
      return t;
   }
}
```

2 Code understanding (5pts)

(Q2.1) What is displayed on the terminal when executing the following class Exercise21 ? (2pts)

```java
public class Exercise21{
   public static void main (String[] args) {
      int a =0;
      boolean b=true;
      int x = 5;
      char y = 'y';
      int res=x;
      x = foxtrot(x,b);
      System.out.print("main : "+a+" "+ b+" "+ x +" "+ y+" " +res);
      System.out.println();
      if (res < 0) {
         System.out.println("bravo ; the end");
      }
   }
   public static int foxtrot (int x, boolean b) {
      int res = 0;
      if (b = true) {
            res=x-3;
            b=false;
        }
        System.out.println("foxtrot 1: " + x + " " +b+ " "+res);
        x= sierra (x, res, b);
        System.out.println("foxtrot 2: " + x + " " +b+ " "+res);
        return res;
    }
    public static int sierra (int a, int x, boolean b) {
        if (! b) {
            x = x%(a+1);
        } else {
            x = x/(a+1);
        }
        System.out.println("sierra : "+a+ " " +x+ " " + b);
        return x;
    }
}
```

Answer : foxtrot 1 : 5 false 2
sierra : 5 2 false
foxtrot 2 : 2 false 2
main : 0 true 2 y 5

(Q2.2) What is displayed on the terminal when executing the following class `Exercise22`? (3pts)

```java
public class Exercise22 {
   public static void main (String[] args) {
      int [ ] t = {1, 7, 12, -4 };
      int [ ] tab = {2, 6, 5, 3};
      int n=2;
      tab=tango(t);
      alpha(t);
      charlie (t,n);
      alpha(t);
      tab=whiskey(t);
      alpha(tab);
   }
   public static void alpha(int []t) {
      for (int i=0;i<t.length;i++){
         System.out.print (t[i] +" ");
      }
      System.out.println( );
   }
   public static void charlie(int [ ] t1, int m){
      while (m>=0) {
         if (m%2==0) {
            t1[m]=t1[m]-1;
         }
         m = m-1;
      }
   }
   public static int [] tango (int []t1){
      int [ ] res=new int [t1.length];
      for (int i=0;i<t1.length;i++) {
         res[i]=t1[i]+i;
      }
      return res;
   }
   public static int [] whiskey (int [ ] t1){
      int [ ] res=new int [t1.length];
      res = t1;
      for (int i=0;i<t1.length;i++) {
         res[i]=res[i]*2;
      }
      return res;
   }
}
```

Answer : 1 7 12 -4
0 7 11 -4
0 14 22 -8

3 Merging arrays (3pts)

When merging two arrays a_1 and a_2 we get a third array a_3 that contains all the values of a_1 and a_2. For instance, merging $\{2, 8, 6\}$ with $\{9, 4, 2\}$ will result in $\{2, 8, 6, 9, 4, 2\}$.

(Q3.1) Write the method `mergeArrays` that takes as input two arrays of integers and return an array of integers containing the values of the two first arrays. (1pt)

In some cases, the arrays are sorted by ascending order, and we want to ensure that the array resulting from the merging is also sorted. For instance , merging $\{2, 8, 6\}$ with $\{2, 4, 9\}$ will result in $\{2, 2, 4, 6, 8, 9\}$ (and not $\{2, 8, 6, 9, 4, 2\}$ as in the previous example).

(Q3.2) Write the method `mergeSortedArrays` that takes as input two arrays of integers sorted by ascending order and return an array of integers, also sorted by ascending order and containing the values of the two first arrays.(2pt)

Answer :

```java
public class MergeArrays{
   public static void main(String args[]){
      int a[] = {2,8,6};
      int b[] = {2,4,9};
      int c[] = mergeSortedArrays(a,b);
      for(int i=0;i<c.length;i++){
         System.out.print(c[i] + " ");
      }
      System.out.println();
   }
   public static int [] mergeArrays(int [] a, int []b){
      int [] c = new int [a.length+b.length];
      int k=0; // index in c
      for(int i=0;i<a.length;i++){
         c[k]=a[i];
         k++;
      }
      for(int i=0;i<a.length;i++){
         c[k]=b[i];
         k++;
      }
      return c;
   }
   public static int [] mergeSortedArrays(int [] a, int []b){
      int [] c = new int [a.length+b.length];
      int k=0; // index in c
      int i=0; // index in a
      int j=0; // index in b
      while(i<a.length && j<b.length){ // take the element one by one
         if(a[i]<b[j]){ // select which element should be taken
            c[k]= a[i];
            i++;
         }else{
            c[k]=b[j];
            j++;
         }
         k++;
```

```
        }
        while(i<a.length){ // flush the rest of the array a
            c[k]= a[i];
            i++;
            k++;
        }
        while(j<b.length){ // flush the rest of the array b
            c[k]= b[j];
            j++;
            k++;
        }
        return c;
    }
}
```

4 Oscillator and trigonometry (9pts)

In this exercise, we want to simulate a physical oscillator (a pendulum) using a computer program. We assume that this oscillator can be modeled using the following equation :

$$\Theta(t) = \Theta_0 \, cos\left(\sqrt{\frac{g}{\ell}} * t\right)$$

where $\Theta(t)$ is the angle at time t, Θ_0 is the initial angle, g is the gravitational constant, ℓ is the length of the pendulum.

To compute the cosine, we will use the following approximation :

$$cos(x) = \sum_{k=0}^{N} \frac{(-1)^k x^{2k}}{(2k)!}$$

where N is the number of terms.

(Q4.1) **Write the method `factorial` that takes as input an integer `n` and returns $n! = 1 \times 2 \times \cdots \times (n-1) \times n$ (1pt)**

In the following we assume that we have access to the following methods of the `Math` library :

— `double a = Math.sqrt(b)` : computes the square root of `b`

— `double a = Math.pow(b,c)` : computes b^c

Other methods such as `Math.cos` **should not be used !**

(Q4.2) **Write the method `cosine` that takes as input a double `x` and an integer `n`, and return an estimation of its cosine, $cos(x)$, computed with n terms. This method should call the method `factorial`. (2pts)**

We have the necessary mathematical tools, we can begin writing the simulation of our oscillator.

(Q4.3) **Write the method `computeAngle` that takes as input, a time `t`, an initial angle `theta0` and a length ℓ, and returns $\Theta(t)$ the angle of the pendulum at time `t`. We will take $g = 6.67$. (1pt)**

We would like to visualize the evolution of our oscillator. To do so, we need to compute the sequence of states taken by the oscillator across a certain period. Given a sequence of dates $\{t_1, t_2, \ldots, t_n\}$, we want to compute the corresponding states $\{\Theta(t_1), \Theta(t_2), \ldots, \Theta(t_n)\}$.

The n dates t_i will be uniformly distributed between two dates t_a and t_b (with $t_a < t_b$), such that the distance between t_i and $t_i + 1$ is $\Delta = (t_b - t_a)/(n - 1)$

(Q4.4) Write the method `computeMultipleAngles` that takes as input two double `tA` and `tB` (resp. start and end of the time interval), an integer `n` representing the number of dates, an initial angle `theta0` and a length ℓ. This method returns an array containing the corresponding values $\Theta(t)$.(2pts)

The visualisation of the angles will be done one by one using a number of "*" to represent the value of the angle. An angle Θ will be represented by the printing of m "*" on a single line. The number of "*" will be computed as $m = \Theta/0.1 + 10$
The following figure represents the expected representation of a sequence of angles taken by an oscillator.

```
********************
*******************
*****************
**************
**********
********
****
**
*
*
**
*****
********
***********
**************
*****************
*******************
*******************
*******************
****************
*************
**********
*******
```

(Q4.5) Write the method `displayAngles` that takes as input an array of double representing a sequence of angles, and that displays those angles using the approach presented above. (1.5pts)

(Q4.6) Write a `main` method that performs the following operations (1.5pts)

— Compute and display the angle for $t = 0$.

— Compute the n angles taken by the oscillator between $t = 0$ and $t = 5$.

— Display the corresponding angles using the method `displayAngles`.

We will use the following parameters

— $n = 50$ (number of dates)

— $\ell = 0.50$ (length of the pendulum)

— $\Theta_0 = 1$

Answer :

```java
public class PendulumTrigonometry {
  public static void main (String[] args) {
    // Compute position of pendulum as a function of t
    // Store results inside an array
    System.out.println(factorial(5));
    double [] angles = computeMultipleAngles(0,5,50,1,0.5);
    displayAngles(angles);
  }
  public static double computePi(int n){
    double x = 0;
    for(int i=0;i<n;i++){
        x=x+(Math.pow(-1,i)/(2*i+1));
    }
    return 4*x;
    //return 3.14;
  }
  public static double cosinus(double x, int n){
    double cos=0;

    for(int i=0;i<n;i++){
        cos+= (Math.pow(-1,n)*Math.pow(x,2*n))/factorial(2*n);
    }
    //return cos;
    return Math.cos(x);
  }
  public static int factorial(int n){
    int res =1;
    for(int i=2;i<=n;i++){
        res = res *i;
    }
    return res;
  }
  public static double computeAngle(double t, double theta0, double l){
    final double G=6.67;
    return theta0 * cosinus(Math.sqrt(G/l)*t,10);
  }
  public static double []computeMultipleAngles(double t0, double t1, int n, double
       theta0, double l){
    double [] angles = new double[n];
    double delta = (t1-t0)/(n-1);
    for(int i=0;i<n;i++){
        angles[i]=computeAngle(t0+i*delta,theta0,l);
        System.out.println(i + " : "+angles[i]);
    }
    return angles;
  }
  public static void displayAngles(double [] angles){
    for(int i=0;i<angles.length;i++){
        int n= (int)(angles[i]/0.1)+10;
        for(int j=0;j<n;j++){
            System.out.print("*");
        }
        System.out.println();
    }
  }
}
```