

مجلس الخدمة المدنية

دائرة المباريات

مباراة للتعيين في بعض الوظائف وللتعاقد
على بعض المهام لدى تعاونية موظفي الدولة

لوظيفة : مبرمج

المدة: ساعتان

مسابقة في البرمجة: Programming

C و OOP (C++ or Java) و .net (VB.net or ASP.net or C#)

Problème 1. Langage C

Soit un vecteur v de 10 nombres entiers non nuls.

Ecrire une suite d'instructions C permettant d'afficher le résultat suivant :

- "En Alternance" : si chaque nombre positif est suivi d'un autre négatif et chaque nombre négatif est suivi d'un autre positif.
- "Non Alternance" : dans les autres cas.

Exemples :

v

2	-3	17	-1	4	5	-1	8	-9	3
---	----	----	----	---	---	----	---	----	---

 Non Alternance

v

12	-3	3	-8	33	-2	1	-8	11	-3
----	----	---	----	----	----	---	----	----	----

 En Alternance

v

-6	15	-4	2	-3	2	-9	8	-6	36
----	----	----	---	----	---	----	---	----	----

 En Alternance

Problème 2. Programmation Orientée Objet (POO)

Supposons que nous voulons modéliser une bibliothèque qui contient de nombreux types d'articles:

- **Printed:** (ex.: les livres, les revues, magazines et documents);
- **Multimedia:** (ex.: vinyle, cassettes, CD et DVD).

Chaque élément (Item) de la bibliothèque doit avoir un numéro d'identification et un titre.
Chaque élément imprimé (printed item) doit avoir un nombre de pages, et chaque élément multimédia doit avoir une longueur, en secondes.

- Définir les classes Item, Printed, et Multimedia, tout en respectant les relations appropriées dans la hiérarchie de classes, et les champs privés appropriés.
- Créer des constructeurs pour les trois classes, en utilisant super() si nécessaire. Pour chaque classe, créer un seul constructeur qui fixe toutes les propriétés de la classe.

- c) Créer des méthodes toString () pour chacune des trois classes qui renvoient une chaîne de caractères correspondante, comme indiqué dans l'astuce ci-dessous.

Astuce: Pour un élément, la chaîne de caractères se compose de son id suivie par son titre. Pour les éléments imprimés et multimédias, l'identifiant et le titre doivent être suivis du nombre de pages ou de la longueur avec l'unité appropriée (pages ou secondes). Par exemple:

7985 Alice in Wonderland (105 pages)
3565 In a Sentimental Mood (597 secondes)

- d) Compléter la classe suivante en implémentant la méthode printItems. Le main de cette classe crée un tableau d'objets Item et l'affiche. La méthode printItems () doit afficher tous les éléments (Items) dans le tableau qui lui est donné comme paramètre, chaque élément sur une ligne.

```
public class PrintItems {
    public static void main(String[] args) {
        Item[] items = new Item[4];
        items[0] = new Printed (7985, "Alice in Wonderland", 105);
        items[1] = new Multimedia (3565, "In a Sentimental Mood", 597);
        items[2] = new Printed (2365, "Building Java Programs", 874);
        items[3] = new Item (5823, "Complete Wreck Diving");
        printItems(items);
    }

    public static void printItems (Item[] items) {
        // ???
    }
}
```

- e) Nous considérons la classe ItemException:

```
class ItemException extends Exception{
    public String toString(){
        return "the array is too big";
    }
}
```

Modifier la classe PrintItem et la méthode printItem afin de lancer une exception ItemException si la longueur du tableau pris comme argument dans la méthode printItem est supérieur à 10.

Problème 3. Dot Net

Considérons les définitions suivantes:

```
struct Time {
    public int h, m, s; // hours, minutes, seconds
}
struct Date {
    public int day; // 1.. 31
    public int month; // 1..12
}
```

```

        public int year;
    }
    struct CompleteDate {
        public Time time;
        public Date date;
    }

```

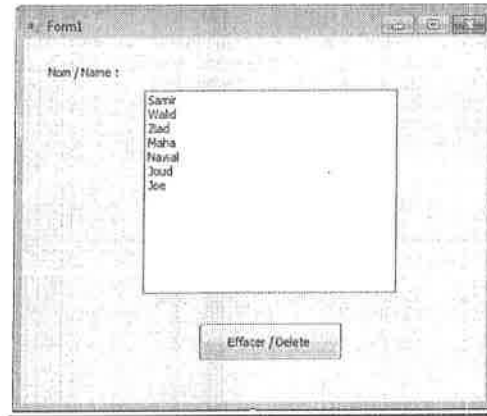
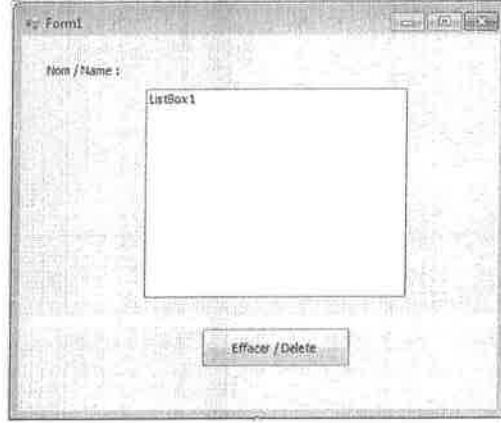
Ecrire les méthodes suivantes:

- a) Bissextile(): prend en paramètres un entier représentant une année et retourne true si cette année est bissextile et false sinon. On rappelle qu'une année est bissextile si elle est divisible par 400 ou si elle est multiple de 4 sans être multiple de 100.
- b) JourAn(): prend en paramètre une structure de type Date et retourne le numéro du jour dans l'année. Par exemple, le numéro de jour de la date 13/5/2004 est 134.
- c) ComparerTemps(): prend en paramètres deux structures de type Time et retourne -1 si la première est antérieure (avant) à la seconde, 1 si elle est postérieure à la seconde, et 0 si elle lui est égale.
- d) ComparerDate(): prend en paramètres deux structures de type Date et retourne -1 si la première est antérieure (avant) à la seconde, 1 si elle est postérieure à la seconde, et 0 si elle lui est égale.
- e) ComparerDateComplete(): prend en paramètres deux structures de type CompleteDate et retourne -1 si la première est antérieure (avant) la seconde, 1 si elle est postérieure à la seconde, et 0 si elle lui est égale.
- f) TrierDatesCompletes() qui prend en paramètres un tableau de CompleteDate ainsi que sa taille et effectue le tri par ordre chronologique décroissant (de la date la plus récente à la plus ancienne).

Problème 4. Dot Net

Une base de données Access nommée "C:\consultation.mdb" contient une table QUESTIONS (QuestID, QuestNom).

- a) Durant le chargement de la page/Form, écrire le code convenable pour se connecter à la base de données et récupérer toutes les valeurs du champ QuestNom et les afficher dans ListBox1.



- b) Ecrire le code correspondant au bouton [Effacer/Delete] pour effacer de la base de données et de Listbox1 l'enregistrement correspondant à la valeur sélectionnée dans Listbox1.

مباراة للتعيين في بعض الوظائف وللتعاقد
على بعض المهام لدى تعاونية موظفي الدولة

لوظيفة : مبرمج

المدة: ساعتان

مسابقة في البرمجة: Programming

C , OOP (C++ or Java) و .net (VB.net or ASP.net or C#)

Problem 1. C language

Given a vector of 10 integers not null.

Write a sequence of instructions in C to display the following result:

- "Alternating" if each positive number is followed by a negative and each negative number is followed by a positive.

- "No Alternating" in other cases.

Examples:

v	2	-3	17	-1	4	5	-1	8	-9	3
---	---	----	----	----	---	---	----	---	----	---

No Alternating

v	12	-3	3	-8	33	-2	1	-8	11	-3
---	----	----	---	----	----	----	---	----	----	----

Alternating

v	-6	15	-4	2	-3	2	-9	8	-6	36
---	----	----	----	---	----	---	----	---	----	----

Alternating

Problem 2. Object Oriented Programming (OOP)

Suppose we want to model a library which contains many kinds of items:

- **Printed:** (ex.: books, journals, magazines, and documents);
- **Multimedia:** (ex.: vinyl, tapes, CDs, and DVDs).

Every item in the library must have an ID number and a title. Every printed item must have a number of pages, and every multimedia item must have a length, in seconds.

- a) Define the classes Item, Printed, and Multimedia, making sure they have the appropriate relationships in the class hierarchy, and the appropriate private fields.
- b) Create constructors for the three classes, using super() where necessary. For each class, create only one constructor which sets all the properties of the class.

- c) For each of the three classes, write a toString() method which returns a corresponding string as shown in the hint below.

Hint: For an item, the string should have its id followed by its title. For printed and multimedia items, the id and title should be followed by the number of pages or length with the appropriate unit (pages or seconds). For example:

7985 Alice in Wonderland (105 pages)
3565 In a Sentimental Mood (597 seconds)

- d) Complete the following class by implementing the printItems method. The main of this class creates an array of Item objects and displays it. The printItems() method should display all of the items in the array given to it as a parameter, with each item on its own line.

```
public class PrintItems {
    public static void main(String[] args) {
        Item[] items = new Item[4];
        items[0] = new Printed (7985, "Alice in Wonderland", 105);
        items[1] = new Multimedia (3565, "In a Sentimental Mood", 597);
        items[2] = new Printed (2365, "Building Java Programs", 874);
        items[3] = new Item (5823, "Complete Wreck Diving");
        printItems(items);
    }

    public static void printItems (Item[] items) {
        // ???
    }
}
```

- e) We consider the class ItemException:

```
class ItemException extends Exception{
    public String toString(){
        return "the array is too big";
    }
}
```

Modify the class PrintItem and the method printItem in order to throw an exception ItemException if the length of the array taken as argument in the method printItem is greater than 10.

Problem 3. Dot Net

Consider the following definitions:

```
struct Time {
    public int h, m, s; // hours, minutes, seconds
}
struct Date {
```

```

        public int day; // 1.. 31
        public int month; // 1..12
        public int year;
    }
    struct CompleteDate {
        public Time time;
        public Date date;
    }

```

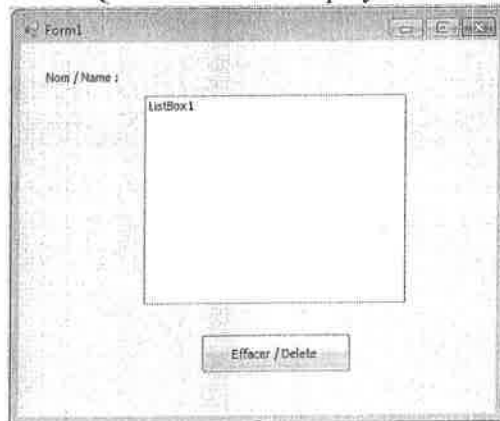
Write the following methods:

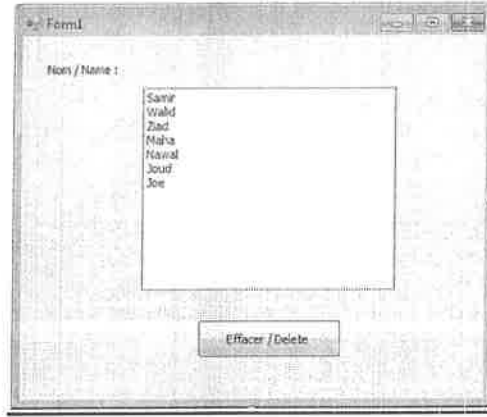
- a) LeapYear(): takes as parameters an integer representing a year and returns true if it's a leap year and false otherwise. A year is leap if it's divisible by 400 or multiple of 4 without being multiple of 100.
- b) DayOfYear(): takes as parameters a structure of type Date and returns the number of the day in the year. For example, the number of the day in the date 13/5/2004 is 134.
- c) CompareTime(): takes as parameters two structures of Time type and returns -1 if the first one precedes the second one, 1 if it follows the second one, and 0 if they are equal.
- d) CompareDate(): takes as parameters two structures of Date type and returns -1 if the first date precedes the second one, 1 if it follows the second one, and 0 if they are equal.
- e) ComparerCompleteDate(): takes as parameters two structures of Date type and returns -1 if the first date precedes the second one, 1 if it follows the second one, and 0 if they are equal.
- f) SortCompleteDates(): takes as parameters an array of type CompleteDate and its number of elements and sorts it in decreasing chronological order (from the most recent date to the oldest one).

Problem 4. Dot Net

An Access database "C:\consultation.mdb", contains a table QUESTIONS (QuestID, QuestName).

- a) During loading of page/Form, write the necessary code to connect to DB and retrieve all the values of the field QuestName and display them in a ListBox1.





- b) Write suitable code corresponding to button [Effacer/Delete] to delete from the database and from the Listbox1, the record corresponding to the selected value in Listbox1.