



Examen du 1<sup>er</sup> semestre du module **Programmation orientée objet en C++** (durée : 1h30)

Spécialités : Electronique des systèmes Embarqués, Instrumentation, Télécommunications et Microélectronique.

**ExerciceN°01 (04points):**

Déterminer pour chaque valeur de  $x=\{4,6,9,11\}$ , ce qui sera affiché le programme suivant:

```
#include <iostream>
using namespace std;
int main()
{
    int x(0);
    cout << "Entrer un nombre entier " << endl;
    cin >> x;
    if( (x%2==0) && (x%3==0))
    {
        cout<< x<<" est multiple de 2 et de 3"<<endl;
    }
    else if (x%2==0)
    {
        cout<< x<<" est multiple de 2"<<endl;
    }
    else if (x%3==0)
    {
        cout<< x<<" est multiple de 3"<<endl;
    }
    else
    {
        cout<< x<<" n'est ni multiple de 2 ni de 3"<<endl;
    }
    return 0;
}
```

**ExerciceN°02 (04 points)**

Déterminer ce qui sera affiché le programme suivant :

```
#include <iostream>
using namespace std;
int main()
{
    char Nom[7] = {'B', 'a', 'h', 'k', 'r', 'z', 'e'};
    char *ptr(0);
    ptr= Nom;
    *(ptr+2)= 'j'; *(ptr+3)= 'a'; *(ptr+5)= 'n';
    cout << Nom [0]; // Nom[0]=.....? (0.75pt)
    cout << Nom [1]; // Nom[1]= .....? (0.5pt)
    cout << Nom [2]; // Nom[2]= ..... ? (0.5pt)
    cout<< Nom [3]; // Nom[3]=.....? (0.5pt)
    cout<< Nom [4]; // Nom[4]= ..... ? (0.5pt)
    cout<< Nom [5]; // Nom[5]=.....? (0.5pt)
    cout<< Nom [6]; // Nom[6]= .....? (0.75pt)
    return 0;
}
```

**ExerciceN°03 (12 points)**

Lire attentivement le programme ci-dessous et répondre aux questions suivantes :

1. Que fait le programme ? (02 pts).
2. Donner le contenu du TabA et de TabB ? (02 pts).
3. Quel est le rôle de l'instruction TabA.push\_back dans le programme ? (02pts).
4. Les tableaux TabA et TabB sont-ils des tableaux statiques ? si oui ou non, justifier votre réponse (02pts).
5. Donner le code d'instruction qui permet d'afficher les contenus de deux tableaux (TabA et ~~TabA~~ TabB) (04pts).

Tab B

```

#include <vector>
using namespace std;
int main()
{
    int Matrice[3][3]={ {15,-12,20}, {5,-8, 8}, {15,1,-18} };
    vector <int> TabA, TabB;

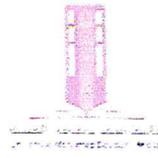
    for(int i=0;i<3;i++)
    {
        for(int j=0;j<3;j++)
        {
            if (Matrice[i][j]>=0)
            {
                TabA.push_back( Matrice[i][j] );
            }
            else
            {
                TabB.push_back( Matrice[i][j] );
            }
        }
    }
}

```

N.B les réponses des trois exercices devront être impérativement sur la double feuille.+

## Bonne chance

Dr. A. Zorig



Réponse au examen du 1<sup>er</sup> semestre du module **Programmation orientée objet en C++** (durée : 1h30)  
Spécialités : Electronique des systèmes Embarqués, Instrumentation, Télécommunications et Microélectronique.

**Réponse au exerciceN°01** (04 points):

Pour  $x=4$ , le programme sera affiché : *4 est multiple de 2* (1pt).

Pour  $x=6$ , le programme sera affiché : *6 est multiple de 2 et de 3* (1pt).

Pour  $x=9$ , le programme sera affiché : *9 est multiple de 3* (1pt).

Pour  $x=11$ , le programme sera affiché : *11 n'est ni multiple de 2 ni de 3* (1pt).

**Réponse au exerciceN°02** (04 points):

Contenu du tableau 'Nom' est le mot '**Bajarne**' (créateur de C++)

```
cout << Nom [0]; // Nom[0]= B (0.75pt)
```

```
cout << Nom [1]; // Nom[1]= a (0.5pt)
```

```
cout << Nom [2]; // Nom[2]= j (0.5pt)
```

```
cout<< Nom [3]; // Nom[3]= a (0.5pt)
```

```
cout<< Nom [4]; // Nom[4]= r (0.5pt)
```

```
cout<< Nom [5]; // Nom[5]= n (0.5pt)
```

```
cout<< Nom [6]; // Nom[6]= e (0.75pt)
```

**Réponse au exerciceN°03** (12 points)

1. Que fait le programme ? (2 pts).

Le programme copie les valeurs positives et nulles de la matrice 'Matrice' dans le tableau dynamique 'TabA' et les valeurs négatives dans le tableau dynamique TabB.

2. Donner le contenu du TabA et de TabB ? (2 pts).

```
TabA= [15, 20, 5,8,15,1] ; TabB= [-12, -8, -18]
```

3. Quel est le rôle de l'instruction TabA.push\_back dans le programme ? (02pts).

4. Cette instruction permet d'incrémenter les tailles des tableaux TabA et TabB (02pts).

5. Les tableaux TabA et TabB ne sont pas des tableaux statiques mais des tableaux dynamiques, puisque la déclaration des deux tableaux se fait par le mot '**vector**' (02pts).

6. Le code d'instruction permet d'afficher les contenu de deux tableaux (TabA et de TabB) (04pts)

```
cout << "TabA=[";
for (int i=0; i<TabA.size(); i++)
{
    cout <<TabA [i]<< "\t";
}
cout << "]"<<endl;
```

```
cout << "TabB=[";
for (int i=0;i<TabB.size();i++)
{
    cout <<TabB [i]<< "\t";
}
cout << "]"<<endl;
```