



Réponse au examen du 1^{er} 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):

Indiquez ce qui sera affiché après l'exécution du programme suivant (remplir les pointillés):

```
#include <iostream>
using namespace std;
int main()
{
int a(2019), b(0);
int &c(a);
cout << c<<endl;// .....c=2019..... (1pt)
b+=a+1;
cout << b<<endl;//.....b=2020..... (1pt)
c--;
cout << a<<endl;// .....a=2018..... (1pt)
a++;
cout <<a<<c<<endl;//.... a= 2019, c=2019... (1pt)
return 0;
}
```

Réponse au exerciceN°02 (05 points)

1. Indiquez ce qui sera affiché le programme ci-contre (remplir les pointillés) (03.5 pts).

```
int main()
{
int Tab[5] = {10,20,30,40,50};
int *p(0);
p=Tab;
*(p+1)=200;
*(p+3)=400;
cout << Tab[0] << endl; // Tab[0]=.... 10..... (0.5pt)
cout << Tab[1] << endl;// Tab [1]= ... 200... (0.5pt)
cout << Tab[2] << endl; // Tab [2]= ...30.....(0.5pt)
cout << Tab[3] << endl; //Tab[3]=.....400.... (0.5pt)
cout << Tab[4] << endl; // Tab [4]= ...50 ..... (0.5pt)
cout << *p << endl; // *p =.....10..... (0.5pt)
cout << *(p+2)<<endl;// (*p+2) =.....30..... (0.5pt)
return 0;
}
```

2. Que fait la troisième instruction (p=Tab) ? (1.5pt)

La troisième instruction initialise le pointeur 'p' par l'adresse du tableau 'Tab'. Le pointeur contient alors l'adresse de la première composante du tableau (Tab[0]).

Réponse au exerciceN°03 (05 points)

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

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

Le programme copie les valeurs paires de la matrice 'Matrice' dans le tableau dynamique 'TabDynA' et les valeurs impaires dans le tableau dynamique TabDynB.

2. Donner le contenu du TabDynA et de TabDynB ? (2 pts).

TabDynA= [1954, 1962, 24] ; TabDynB= [01, 11, 05, 07, 01, 2019]

3. Quel est le rôle des instructions TabDynA.size() et TabDynB.size() dans le programme ? (1pt).

Les instructions TabDynA.size() et TabDynB.size() permettent de calculer les tailles des tableaux dynamiques TabDynA et TabDynB, respectivement.

Réponse au exerciceN⁰04 (06 points)

Programme qui :

1. Lit les moyennes de N étudiants et les stocker dans un tableau statique ;
2. Calcul et affiche les statistiques demandés (Nombre des moyennes supérieurs à 10, inférieurs à 10, entre 9.5 et 10, la moyenne générale de la promotion, la meilleure et la faible moyenne).

```
1
2 using namespace std;
3 int main()
4 {
5     int N (0),i(0);
6     cout << "Entrez le nombre des étudiants (N)" << endl;
7     cin >> N;
8     float Tab [N]; //Déclaration du tableau
9
10    for (i=0;i<N;i++) //Saisie des moyennes des étudiants
11    {
12        cout << "Entrer une moyenne ";
13        cin >> Tab [i];
14    }
15    // calcul les statistiques demandées
16    int nbrMoySup10(0),nbrMoyInf10(0),nbrMoy9510(0); //variables
17    float MeillMoy(Tab[0]), faibleMoy(Tab[0]),MoyGen(0); //variables
18
19    for(i=0;i<N;i++)
20    {
21        MoyGen+= Tab[i]/N; // calcul de la moyenne générale
22        if (Tab[i]> MeillMoy) {MeillMoy =Tab[i];} // Détermine la meilleure moyenne
23        if (Tab[i]< faibleMoy) {faibleMoy=Tab[i];} // Détermine la faible moyenne
24
25        if(Tab[i]>=10)
26        {
27            nbrMoySup10++; // Nombre des moyennes supérieurs à 10
28        }
29        else if (Tab[i]< 10)
30        {
31            nbrMoyInf10++; // Nombre des moyennes inférieurs à 10
32            if (Tab[i]>=9.5000000)
33            {
34                nbrMoy9510++; // Nombre des moyennes entre 9.5 et 10
35            }
36        }
37    }
38    // Affichage des résultats
39
40    cout << "Nombre des moyennes supérieurs à 10 = " << nbrMoySup10 << endl;
41    cout << "Nombre des moyennes inférieurs à 10 = " << nbrMoyInf10 << endl;
42    cout << "Nombre des moyennes entre 9.5 et 10 = " << nbrMoy9510 << endl;
43    cout << "Moyenne générale = " << MoyGen << endl;
44    cout << "Meilleure moyenne = " << MeillMoy << endl;
45    cout << "Faible moyenne = " << faibleMoy << endl;
46    return 0;
47 }
```