

Session Automne, 2005

Mohamed Lokbani

IFT1166 – INTRA

Inscrivez tout de suite votre nom et code permanent.

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date : 18 octobre 2005

Durée: 2 heures (de 16h30 à 18h30) Local: N-515 du Pavillon Roger-Gaudry.

Directives:

- Toute documentation permise.
- Calculatrice **non** permise.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises et clairement présentées.**

1. _____ /22 (1.1, 1.2, 1.3)
2. _____ /15 (2.1, 2.2, 2.3)
3. _____ /12 (3.1)
4. _____ /16 (4.1, 4.2, 4.3, 4.4)
5. _____ /20 (5.1, 5.2)
6. _____ /15 (6.1)

Total: _____ /100

Exercice 1 (22 points)

1.1 Il existe toujours de multiples façons d'écrire un programme qui résout un certain problème. Un bon programmeur ne se contente cependant pas d'écrire un programme qui fonctionne sans se soucier de la lisibilité de son code et de l'élégance de sa solution. Pourquoi? Donnez trois raisons.

1.2 Écrire en C++, les instructions nécessaires (pas un programme) pour afficher en sortie le texte suivant en respectant le format d'affichage:

```
Deux_idiots_a_la_chasse,_voient_passer_un_deltaplane._Le_premier_epaule_et_tire
...▼
□ "Bah_tu_l'as_rate!!",_commente_le_second.▼
□ "Oui_mais_il_a_lache_sa_proie.",_repond_le_premier.▼
```

Dans le texte : « □ » correspond à une tabulation (un « TAB »), « ▼ » à un saut de ligne et « _ » à un espace blanc.

1.3. Faire ce qui suit en C++:

- ```
employeesAddresses[6] = "9999 Rue Sans Fin";
```

**Exercice 2 (15 points)** Tout en justifiant votre réponse, que vont afficher en sortie les différents fragments de code qui compilent et s'exécutent correctement (dans un programme complet) :

**2.1**

```
x = 9;
y = 6;
z = 4;
if ((x < y) && (y != 7)) cout << "A\n";
else if ((z < 7) || (y == 6)) cout << "B\n";
else cout << "C\n";
cout << "D\n";
cout<<"E"<<endl;
```

**2.2**

```
int A[8]={1,2,4,6,7,7,2,1};
int i = 2;
while (i <= 5){
 A[i] = A[i] - i;
 i++;
}
for(i=0;i<8;i++){
 cout << A[i] <<" ";
}
```

## 2.3

```
int main(){
 float num1;
 num1 = 123.456;
 cout << '|' << num1 << '|' << endl;
 cout << setiosflags(ios::fixed);
 cout << setiosflags(ios::showpoint);
 cout << setprecision(2) << num1 << endl;
 cout << '|' << setw(8) << num1 << '|' << endl;
 cout << '|' << setw(1) << num1 << '|' << endl;
}
```

**Exercice 3 (12 points)** Supposer les déclarations suivantes:

```
struct nourriture {
 char nom[15];
 int portion;
 int calories;
};
```

Et soit la déclaration:

```
struct nourriture repas[5];
```

Expliquez la différence entre les expressions:

```
repas[2].nom (et) repas->nom[2]
```

**Exercice 4 (16 points)** Soit le fragment de code suivant :

```
float a = 0.1;
float b = 0.3;
float c, *pa, *pb;

pa = &a;
*pa = 2 * a;
pb = &b;
c = 3 * (*pb - *pa);
```

En supposant que chaque nombre réel (un float) occupe 4 octets en mémoire, que l'adresse de la variable « **a** » est « **0x1130** », l'adresse de la variable « **b** » est « **0x1134** » et finalement l'adresse de la variable « **c** » est « **0x1138** ». **Tout en justifiant votre réponse**, répondre aux questions suivantes:

**4.1** Quelle est la valeur assignée à « **pa** »?

**4.2** Quelle est la valeur assignée à « **\*pa** »?

**4.3** Quelle est la valeur assignée à « **&(\*pa)** »?

**4.4** Quelle est la valeur assignée à « **\*(&c)** »?

**Exercice 5 (20 points)** Soit le programme suivant qui compile et s'exécute correctement :

```
#include <iostream>
using namespace std ;
const float num = 2.5;
void Test(float& a, int b){
 float c;
 c = a + num;
 ++a;
 ++b;
 cout << "Les valeurs de a, b, et c dans Test sont: "\
 << a << ' ' << b << ' ' << c << endl;
}
int main(){
 float r = 0;
 int s, t;

 s = 5;
 t = 10;

 cout << "Les valeurs de r, s, et t sont: "
 << r << ' ' << s << ' ' << t << endl;
 Test(r, t);
 cout << "Les valeurs de r, s, et t sont: "
 << r << ' ' << s << ' ' << t << endl;
 Test(r, s);
 cout << " Les valeurs de r, s, et t sont: "
 << r << ' ' << s << ' ' << t << endl;
}
```

**5.1** Que va afficher le programme en sortie?

**5.2** Si la méthode « **main** » contenait aussi l'appel « **Test(t,r)** »,

```
// etc.
int main(){
 float r = 0;
 int s, t;

 s = 5;
 t = 10;
 Test(t, r);
 // etc.
}
```

Est-ce que le programme va compiler et s'exécuter correctement ? Expliquer votre réponse.

**Exercice 6 (15 points)** Écrivez une fonction `afficheTriangle(N,f,c1,c2)` qui affiche un triangle de « N » lignes. Le triangle est rempli avec le caractère « f » et sa diagonale est une alternance entre les caractères « c1 » et « c2 », en commençant par « c1 » sur la première ligne. N'oubliez pas de déclarer l'entête de la fonction (le type de retour, le nom, les paramètres).

Exemple: l'appel « `afficheTriangle( 7, '.', '+', '-')` » va afficher ceci:

```
.....+
.....-
.....+
.....-
....-
..+
.-
+
```

---

**La simplicité de votre code et la clarté de sa présentation  
sont des éléments importants de la notation**

---