

Trimestre Eté, 2007

Mohamed Lokbani

IFT1166 – Examen Intra –

Inscrivez tout de suite votre nom et le code permanent.

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date : mardi 29 mai 2007

Durée : 2 heures (de 17h30 à 19h30)

Local : N-515 ; 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. _____ /25 (1.1, 1.2, 1.3, 1.4)

2. _____ /15 (2.1)

3. _____ /15 (3.1, 3.2, 3.3, 3.4)

4. _____ /20 (4.1, 4.2, 4.3, 4.4, 4.5)

5. _____ /25 (5.1, 5.2, 5.3)

Total: _____ /100

Directives officielles

* Interdiction de toute communication verbale pendant l'examen.

* Interdiction de quitter la salle pendant la première heure.

* L'étudiant qui doit s'absenter après la première heure remettra sa carte d'étudiant au surveillant, l'absence ne devant pas dépasser 5 minutes. Un seul étudiant à la fois peut quitter la salle.

* Toute infraction relative à une fraude, un plagiat ou un copiage est signalée par le surveillant au directeur de département ou au professeur qui suspend l'évaluation.

F.A.S

Exercice 1 (25 points)

1.1 Expliquez brièvement les différences entre le fichier « test.o » produit par la commande « g++ -c test.cpp -o test.o » et le fichier « test.exe » produit par la commande « g++ test.o -o test.exe ».

1.2 Tout en expliquant brièvement votre réponse, que va retourner la commande « sizeof » pour les deux cas suivants :

```
char Couleur[ ] = "violet";  
sizeof(Couleur) : _____
```

```
char *pCouleur = "jaune";  
sizeof(pCouleur) : _____
```

1.3 Soit les déclarations suivantes :

```
double racine = 1.414213562, micro = 1.234e-9, mega = 1.234e9 ;
```

Tout en expliquant brièvement votre réponse, que vont-elles afficher en sortie les deux instructions suivantes, indépendantes l'une de l'autre :

```
a) cout << scientific << '\t' << racine << '\t' << micro << '\t' << mega << endl ;
```

b) `cout << fixed << '\t' << racine << '\t' << micro << '\t' << mega << endl ;`

1.4 Soit la fonction « Test » suivante :

```
void Test( int& intVal, double doubleVal ) {  
    intVal = intVal * 2;  
    doubleVal = double (intVal) + 3.5;  
}
```

Que va afficher en sortie le fragment de code suivant :

```
int unInt = 20;  
float unDble = 4.8;  
Test(unInt, unDble);  
cout << "unInt = " << unInt << " et unDble = " << unDble << endl;
```

- | | |
|---------------------------------|---------------------------------|
| (a) unInt = 20 et unDble = 43.5 | (b) unInt = 40 et unDble = 4.8 |
| (c) unInt = 20 et unDble = 4.8 | (d) unInt = 40 et unDble = 43.5 |
| (e) Aucune de ces réponses | |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

Exercice 2 (15 points) Soit la variable « n » du type « int » ayant une valeur initiale donnée. Écrivez les instructions booléennes suivantes :

```
// « n » est entre 5 et 15, sachant que n peut être égal aussi à 5 ou 15
```

```
bool a = _____ ;
```

```
// « n » n'est pas entre 5 et 15, sachant que n ne peut pas être égal à 5 ou 15  
// (interdiction d'utiliser l'opérateur de négation « ! »)
```

```
bool a = _____ ;
```

```
// « n » est un nombre pair
```

```
bool a = _____ ;
```

```
// « n » est un nombre impair (interdiction d'utiliser l'opérateur de négation « ! »)
```

```
bool a = _____ ;
```

```
// « n » n'est pas divisible par le nombre « 5 » (interdiction d'utiliser l'opérateur de  
// négation « ! »)
```

```
bool a = _____ ;
```

Exercice 3 (15 points) Soit le fragment de code suivant :

```
bool a,b,c,d;

// fragment de code pour initialiser les variables « a », « b », « c » et « d » ...

if (a && b)
    if (!c || !d)
        cout << "un" << endl;
    else if (d)
        cout << "deux" << endl;
    else
        cout << "trois" << endl;
else if (c != d)
    cout << "quatre" << endl;
else if (c)
    cout << "cinq" << endl;
else
    cout << "six" << endl;
```

3.1 Le programme va-il afficher, peu importe les valeurs de « a », « b », « c » et « d », un résultat en sortie ?

- (a) vrai | (b) faux | (c) pas de sens

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

3.2 Quel est l'ensemble de valeurs « a », « b », « c » et « d » qui va permettre l'affichage de la chaîne « un » ?

	a	b	c	d
(1i)	true	true	true	true
(2i)	true	true	true	false
(3i)	true	true	false	true
(4i)	true	true	false	false

- (a) Toutes les combinaisons | (b) Les combinaisons (1i) et (2i) uniquement
(c) Les combinaisons (1i), (2i), (3i) uniquement | (d) Les combinaisons (2i) et (3i) uniquement
(e) Les combinaisons (2i), (3i), (4i) uniquement | (f) Aucune de ces combinaisons

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

3.3 Quel est l'ensemble de valeurs « a », « b », « c » et « d » qui va permettre l'affichage de la chaîne « trois » ?

	a	b	c	d
(1i)	true	true	true	true
(2i)	true	true	true	false
(3i)	true	true	false	true
(4i)	true	true	false	false

- | | |
|--|--|
| (a) Toutes les combinaisons | (b) Les combinaisons (1i) et (2i) uniquement |
| (c) Les combinaisons (1i), (2i), (3i) uniquement | (d) Les combinaisons (2i) et (3i) uniquement |
| (e) Les combinaisons (1i), (3i), (4i) uniquement | (f) Aucune de ces combinaisons |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

3.4 Quel est l'ensemble de valeurs « a », « b », « c » et « d » qui va permettre l'affichage de la chaîne « quatre » ?

	a	b	c	d
(1i)	true	false	true	true
(2i)	false	true	false	true
(3i)	false	true	true	false
(4i)	false	false	false	false

- | | |
|--|--|
| (a) Toutes les combinaisons | (b) Les combinaisons (1i) et (2i) uniquement |
| (c) Les combinaisons (1i), (2i), (3i) uniquement | (d) Les combinaisons (2i) et (3i) uniquement |
| (e) Les combinaisons (1i), (3i), (4i) uniquement | (f) Aucune de ces combinaisons |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

Exercice 4 (20 points) La fonction suivante permet de chercher l'élément le plus grand dans un tableau :

<pre>int maxEntree(const int* const Data, int Sz) { if (Data == NULL Sz <= 0) return -1; int Compteur = 1; // initialiser « unptr » pour pointer le premier élément // du tableau const int *unptr = _____ ; // initialiser ptrcourant pour pointer le second élément // du tableau sans modifier les autres variables const int *ptrcourant = _____ ; for (; Compteur < Sz; _____) { if (_____) unptr = ptrcourant; } return (_____); }</pre>	<pre>//1 //2 //3 //4 //5 //6 //7 //8 //9</pre>
--	--

4.1 Complétez la ligne « **4** » du programme

- | | |
|-----------|-------------|
| (a) Data | (b) *Data |
| (c) &Data | (d) Data[0] |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

4.2 Complétez la ligne « **5** » du programme

- | | |
|-------------|--------------|
| (a) unptr | (b) unptr++ |
| (c) Data++ | (d) &Data[1] |
| (e) Data[1] | |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

4.3 Complétez la ligne « 6 » du programme (**attention : cette question est dépendante de la question 4.4**)

- | | | |
|---------------------------------|--|-------------------------------|
| (a) Compteur++ | | (b) ptrcourant++ |
| (c) Compteur++, ptrcourant++ | | (d) Doit être laissée vide |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

4.4 Complétez la ligne « 7 » du programme

- | | | |
|-----------------------------|--|-----------------------------|
| (a) ptrcourant > unptr | | (b) &ptrcourant > &unptr |
| (c) *ptrcourant > *unptr | | (d) *ptrcourant < *unptr |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

4.5 Complétez la ligne « 9 » du programme

- | | | |
|---------------|--|-------------------------------|
| (a) *unptr | | (b) &unptr |
| (c) unptr | | (d) Doit être laissée vide |

Un seul choix possible, accompagné d'une courte explication (pourquoi celui-ci mais pas un des autres)

Exercice 5 (25 points) Le fragment de code suivant déclare et initialise deux structures identiques « UnTruc1 » et « UnTruc2 ». Il appelle la méthode « doublerA » pour doubler la valeur des membres de la structure « UnTruc1 » et la méthode « doublerB » pour doubler la valeur des membres de la structure « UnTruc2 ». Le but de cet exercice est de comparer ces deux approches.

```
typedef struct {
    int x;
    int y;
} UnTruc;

int main(){
    UnTruc UnTruc1, UnTruc2;
    UnTruc1.x = UnTruc2.x = 10;
    UnTruc1.y = UnTruc2.y = 250;
    UnTruc1 = doublerA(UnTruc1);
    doublerB(&UnTruc2);
    cout << "UnTruc1.x = " << UnTruc1.x << ", UnTruc2.x = " << UnTruc2.x << endl;
    cout << "UnTruc1.y = " << UnTruc1.y << ", UnTruc2.y = " << UnTruc2.y << endl;
    return 0;
}
```

Quand le programme est complété et exécuté, il produit l'affichage suivant en sortie :

```
UnTruc1.x = 20, UnTruc2.x = 20
UnTruc1.y = 500, UnTruc2.y = 500
```

5.1 Quel est la fonction « doublerA » ou « doublerB » qui consomme le plus de temps et d'espace ? Expliquez pourquoi ?

5.2 Complétez le programme en écrivant le code complet des fonctions « doublerA » et « doublerB » ainsi que leur prototype.

5.3 Soit le fragment de code suivant :

```
UnTruc *faireUnTruc(int a, int b){
    UnTruc UnTruc;
    UnTruc.x = a;
    UnTruc.y = b;
    return &UnTruc;
}
int main(){
    UnTruc *p;
    p = faireUnTruc(5, 50);
    cout << "p->x = " << p->x << ", p->y = " << p->y << endl;
}
```

```
typedef struct {
    int x;
    int y;
} UnTruc;
```

Le programmeur a eu comme intention d'utiliser la fonction « faireUnTruc » pour créer une structure dont ses paramètres vont être initialisés avec les paramètres passés en argument de la fonction i.e. « a » et « b » ; et de retourner un pointeur vers cette nouvelle structure créée. Cette nouvelle structure sera accessible ailleurs dans le programme comme dans la méthode « main ». Ainsi le programme pourrait afficher en sortie :

```
p->x = 5, p->y = 50
```

Cependant il y a un problème mémoire avec la fonction « faireUnTruc » ! Heureusement que le compilateur « g++ » l'a repéré et il nous a affiché un message d'avertissement au moment de la compilation du programme.

Décrivez la nature du problème et récrivez le cœur de la fonction « faireUnTruc » pour corriger ce problème. Attention l'entête de la fonction « faireUnTruc » doit rester la même.