

**IFT1166 - INTRA**

Inscrivez tout de suite votre nom et code permanent.

Nom: \_\_\_\_\_ Prénom(s): \_\_\_\_\_

Signature: \_\_\_\_\_ Code perm.: \_\_\_\_\_

Date : 25 octobre 2003

Durée: 2 heures (de 15h00 à 17h00) Local: N-515 du Pavillon Principal (P.P).

**Directives:**

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

1. \_\_\_\_\_/15 (1.1 ; 1.2 ; 1.3 ; 1.4 ; 1.5)

2. \_\_\_\_\_/20 (2.1 ; 2.2 ; 2.3 ; 2.4 ; 2.5 ; 2.6)

3. \_\_\_\_\_/20 (3.1 ; 3.2)

4. \_\_\_\_\_/20 (4.1)

5. \_\_\_\_\_/25 (5.1)

Total: \_\_\_\_\_/100

**Question 1 (15 points)**

**1.1** Les tableaux en C++ commencent toujours par l'élément 0.

VRAI   
FAUX

**1.2** Écrire la déclaration d'une chaîne de caractères constante, initialisée avec votre nom et prénom.

**1.3** Lors du processus de compilation, un programme C++ est d'abord traité par \_\_\_\_\_.

1. un préprocesseur
2. un chargeur
3. un éditeur de liens
4. un correcteur orthographique
5. un optimisateur

**1.4** Avant d'utiliser une variable, elle doit être d'abord \_\_\_\_\_.

1. incluse.
2. définie.
3. imprimée.
4. préprocessée.
5. aucune des précédentes réponses, préciser alors : \_\_\_\_\_.

**1.5** Il est parfaitement possible d'initialiser un tableau. Il suffit pour cela de donner les valeurs de ses éléments, séparées par des \_\_\_\_\_, entre \_\_\_\_\_.

### Question 2 (20 points)

2.1 Quel sont le nom et le type de la fonction membre suivante :

```
Data* Data::test(int i) const { ... }
```

2.2 L'instruction correspondante à « p égale à 5 » est « p==5 » ; écrire alors l'instruction qui correspond à « z est plus petit que ou égale à 0 ou bien z est plus grand que 100 ».

2.3 Tout en expliquant votre réponse, quelle sera la valeur retournée par la fonction `foo` :

```
int foo(int j){
    int x=5;
    if(j=0){
        int x ;
        x = 3 ;
    }
    return x ;
}
```

2.4 Tout en expliquant votre réponse, quelle sera la valeur retournée par la fonction `f` :

```
int autre=0;
void g(int& valeur){
    valeur=13;
}
int f(){
    int autre=42;
    g(autre);
    return autre;
}
```

IFT1166

Automne-2003 : Intra

3/10

2.5 Tout en expliquant votre réponse, le code de la fonction suivante est-il correct ?

```
int &rest( int &r, int i ) {
    r = i*i ;
    return r ;
}
```

2.6 Tout en expliquant votre réponse, le programme suivant est-il correct, si oui que va-t-il afficher en sortie :

```
#include <iostream>
using namespace std ;

int main() {
    int *p = new int ;
    if( p!=NULL ) {
        *p = 99 ;
        int *q = p ;
        delete q ;
        cout << *p << endl ;
    }
    return 0 ;
}
```

IFT1166

Automne-2003 : Intra

4/10

### Question 3 (20 points)

3.1 Écrire une implémentation possible de la fonction `strlen` qui renvoie la longueur d'une chaîne de caractères (caractère nul final non compris).

3.2 Quelles modifications faudrait-il faire à la fonction précédente, si l'on voulait trouver le nombre d'éléments non nuls consécutifs dans une table d'entiers désignée par un pointeur `int*` ?

IFT1166

Automne-2003 : Intra

5/10

### Question 4 (20 points)

Que va afficher en sortie le programme suivant, qui compile et s'exécute correctement.

```
#include <iostream>
using namespace std;
class ex01 {
private :
    int x,y;
public :
    ex01 ( int abs=0, int ord = 0 ) {
        x = abs;
        y = ord;
    }
    cout << "zone 1 : (" << x << ", " << y << ") \n";
    ex01 (const ex01 &);
    ~ex01();
};
ex01::ex01(const ex01 & a) {
    x = a.x;
    y = a.y;
    cout << "zone 2 : (" << x << ", " << y << ") \n";
}
ex01::~ex01() {
    cout << "destruction de (" << x << ", " << y << ") \n";
}
int main() {
    void fct(ex01, ex01 *);
    cout << "début de main \n";
    ex01 a;
    ex01 b=ex01(2);
    ex01 c=a;
    ex01 * adr=new ex01(4,4);
    fct(a,adr);
    ex01 d = ex01(5,5);
    c = ex01(6,6);
    cout << "fin de main \n";
}
void fct(ex01 a, ex01 * b) {
    cout << "entrée fct \n";
    delete b;
    cout << "sortie fct \n";
}
```

IFT1166

Automne-2003 : Intra

6/10

### Question 5 (25 points)

Écrire la définition de la classe `MLTime` ainsi que toutes ses méthodes, qui permet de convertir un temps de la représentation militaire (00h, 01h, ..., 21h, 22h, ..., 23 heures) vers une représentation classique heure minute et am/pm. Ainsi par exemple :

Représentation internationale	Représentation américaine
Hh:mm:ss	Hh:mm:ss
00:00:00	12:00:00 a.m
06:30:30	6:30:30 a.m
12:00:00	12:00:00 p.m
23:45:45	11:45:00 p.m

La classe `MLTime` doit contenir entre autre :

- Un constructeur par défaut permettant d'initialiser les membres de la classe avec la valeur zéro.
- Une méthode `set_data`, dont les arguments sont utilisés pour initialiser les membres de la classe. Par ailleurs, cette méthode s'assurera de la validité du format de ses arguments.
- Une méthode pour afficher le temps dans la représentation internationale.
- Une méthode pour afficher le temps dans la représentation américaine.

Écrire un programme principal qui utilisera les différentes méthodes de votre classe.

**Attention** : La conception de votre classe et votre programme doit être conforme aux principes de base de la programmation orientée objet.

JFT1166

Automne-2003 : Intra

9/10

JFT1166

Automne-2003 : Intra

10/10