

Trimestre Hiver, 2010

Mohamed Lokbani

IFT1169 – Examen Final –

Inscrivez tout de suite : votre nom et le code permanent.

Nom: _____ | Prénom(s): _____ |

Signature: _____ | Code perm: _____ |

Date : mardi 20 avril 2010

Durée : 3 heures (de 18h30 à 21h30)

Local : S-142 ; Pavillon Roger Gaudry

Directives:

- Toute documentation est permise.
- Calculatrice **non** permise.

- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises, claires et nettement présentées.**

1. _____ /15 (1.1, 1.2, 1.3, 1.4, 1.5)

2. _____ /15 (2.1, 2.2, 2.3, 2.4)

3. _____ /15 (3.1)

4. _____ /15 (4.1)

5. _____ /20 (5.1, 5.2, 5.3)

6. _____ /20 (6.1)

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 (15 points)

1.1 Les « templates » peuvent être utilisées pour représenter une large collection de _____.

- a) programmes
- b) classes
- c) fichiers
- d) fonctions
- e) aucune des précédentes réponses. En effet le C++ ne supporte pas les « templates ».

1.2 Le code associé à une fonction « template » est généré:

- a) lorsque la déclaration de la fonction « template » (le prototype) apparaît dans le programme C++.
- b) lorsque une fonction « template » est rencontrée dans un programme C++.
- c) lorsque un appel à la fonction est rencontré dans un programme C++.
- d) à l'exécution du programme, lorsque la fonction est exécutée.

1.3 Décrire brièvement (stratégie) comment vous allez procéder lors de l'écriture de fonctions « templates » (quelconques).

1.4 Avez-vous rencontré des problèmes lors de la compilation de programmes utilisant des « templates »? Si oui, décrire brièvement ces problèmes et comment vous avez réussi à les solutionner.

1.5 Le programme suivant ne passe pas l'étape de la compilation. Expliquer brièvement le(s) erreur(s), et proposer les corrections nécessaires.

```
01 #include <iostream>
02
03 using namespace std;
04
05
06 template<class T>
07 void f( int & x ) {
08     return x;
09 }
10
11 int main() {
12
13     int y = 3, z;
14
15     z = f(y);
16
17     return 0;
18 }
19
```

Exercice 2 (15 points) Le programme suivant passe correctement l'étape de compilation :

```
01 #include <iostream>
02 #include <exception>
03 #include <stdexcept>
04 #include <cstdlib>
05
06 using namespace std;
07
08 const int MAX = 10;
09 int foo(int bar) {
10     cout << "L'entree est: " << bar << endl;
11     if ( bar > MAX ) {
12         out_of_range e("L'entree est en dehors du segment autorise");
13         throw e;
14     }
15     return bar++;
16 }
17
18 int main ( int argc, char **argv ) {
19     int entree = atoi(argv[1]);
20     int sortie = entree;
21     try {
22         sortie = foo (entree);
23         cout << "La sortie est: " << sortie << endl;
24     } catch ( exception &e ) {
25         cout << "exception a ete attrapee: " << e.what() << endl;
26     }
27     return 0;
28 }
29
```

Que va-t-on obtenir en sortie lors de l'exécution du précédent programme dans les situations suivantes :

2.1 exo2.exe 5

Affichage :

Explications :

2.2 exo2.exe 10

Affichage :

Explications :

2.3 exo2.exe 15
Affichage :
Explications :

2.4 exo2.exe
(Sans aucun argument)
Affichage :
Explications :

Exercice 3 (15 points) Le programme suivant passe correctement l'étape de compilation :

```
01 #include<iostream>
02 #include<vector>
03
04 using namespace std;
05
06
07 class UnConteneur {
08 public:
09     UnConteneur() : unEntier(0) { }
10     ~UnConteneur() { }
11     void set(int* t) {
12         unEntier = t;
13     }
14     int get(void) {
15         return *unEntier;
16     }
17 private:
18     int* unEntier;
19 };
20
21 UnConteneur* creerConteneur(int i) {
22     UnConteneur* ret = new UnConteneur;
23     ret->set(&i);
24     return ret;
25 }
26
27 const int vecTaille = 5;
28
29 int main(int argc, char** argv) {
30     vector<UnConteneur*> valeur;
31     for(int i = 0; i < vecTaille; i++) {
32         valeur.push_back(creerConteneur(i));
33     }
34     for(int i = 0; i < vecTaille; i++) {
35         cout << valeur[i]->get() << " ";
36     }
37     cout << endl;
38     for(int i = 0; i < vecTaille; i++) {
39         delete valeur[i];
40     }
41     return 0;
42 }
```

3.1 Alors que l'affichage en sortie attendu devait être normalement [0 1 2 3 4], nous obtenons en réalité un affichage similaire à [2293552 4599648 4599648 4599648 4599648]. Il est clair que cette sortie est erronée. Expliquer les raisons et proposer les corrections nécessaires pour que le programme s'exécute correctement.

Exercice 4 (15 points) Le programme suivant qui compile correctement, remplace normalement une occurrence par une autre dans un fichier donné :

```
01 #include<iostream>
02 #include<fstream>
03 #include<string>
04
05 using namespace std;
06
07 int main(int argc, char** argv) {
08     if (argc < 4) {
09         cout << "La ligne de commandes: nom de fichier, un mot et son remplaceant" << endl;
10         return 1;
11     }
12     ifstream ifs(argv[1]);
13     if (!ifs) {
14         cout << "Je ne peux pas ouvrir le fichier " << argv[1] << endl;
15         return 2;
16     }
17     string s;
18     while(!ifs.eof()) {
19         ifs >> s;
20         if (ifs.eof()) {
21             break;
22         }
23         if (s == argv[2]) {
24             cout << argv[3];
25         } else {
26             cout << s;
27         }
28         cout << " ";
29     }
30     cout << endl;
31     ifs.close();
32     return 0;
33 }
34
```

Soit le fichier « test.in » fourni en entrée:

Mars est une des planètes du système solaire. Mars fait partie des 5 planètes visibles à l'œil nu. Plusieurs missions spatiales d'observation puis d'exploration ont permis de mieux connaître les caractéristiques de la planète "Mars" : sa géographie, son atmosphère, etc. Mars a aussi influencé un grand nombre d'auteurs de science fiction.

Suite à l'exécution du précédent programme avec la commande :

```
exo4.exe text.in Mars Venus
```

Nous avons obtenu en sortie l'affichage suivant:

Venus est une des planètes du système solaire. Venus faisant partie des 5 planètes visibles à l'œil nu. Plusieurs missions spatiales d'observation puis d'exploration ont permis de mieux connaître les caractéristiques de la planète "Mars" : sa géographie, son atmosphère, etc. Venus a aussi influencé un grand nombre d'auteurs de science fiction.

Nous constatons que toutes les occurrences de Mars ont été remplacé par Venus, sauf une.

4.1 Corriger le précédent programme afin de réaliser le traitement demandé (juste les instructions nécessaires, mentionner les lignes où il faudra inclure ces corrections). Attention, il ne faudra pas cibler uniquement le caractère « " ». En effet, le mot peut-être entouré d'autres caractères.

Exercice 5 (20 points) Pour cet exercice, on vous demande 3 choses : de compléter les instructions, d'ajouter des instructions dans le programme, de donner l'affichage en sortie.

5.1 Compléter les instructions du précédent programme :

```

01  #include <_____>
02  #include <iostream>
03
04  using namespace std;
05
06  #define NBRETHRDS 8
07  #define VECTAILLE 100000
08  int *a, *b;
09  long somme=0;
10
11  void *unefonction(_____) {
12
13      int i, debut, fin, offset, taille;
14      long tid = (long)arg;
15      offset = tid;
16      taille = VECTAILLE;
17      debut = offset*taille;
18      fin = debut + taille;
19
20      cout << "thread: " << _____ << " debut= " << debut << "|fin=" << fin-1 << endl;
21
22      for (i=debut; i<fin ; i++){
23          somme += (a[i] * b[i]);
24      }
25
26      cout << "thread: " << _____ << " a termine.\n
27          La somme globale est maintenant= " << _____ << endl;
28
29      pthread_exit_____ ;
30      return _____ ;
31  }
32
33  int main (int argc, char *argv[]) {
34      long i;
35      void *status;
36      pthread_t threads[_____];
37      _____ attr;
38
39      a = new _____[NBRETHRDS*VECTAILLE*sizeof(_____)] ;
40      b = new _____[NBRETHRDS*VECTAILLE*sizeof(_____)] ;
41
42      for (i=0; i<VECTAILLE*NBRETHRDS; i++) a[i]= b[i]=1;
43
44      _____(&attr);
45      _____(&attr, _____);
46      for(i=0; i<NBRETHRDS; i++)
47          pthread_create(_____, &attr, _____);
48
49      pthread_attr_____(&attr);
50
51      for(i=0; i<NBRETHRDS; i++)
52          _____(threads[i], _____);
53
54      cout << "La somme finale est= " << somme << endl;
55
56      _____ a;
57      _____ b;
58      _____(NULL);
59
60      _____ 0;
61  }
62

```

5.2 Protéger les zones critiques du programme avec les instructions nécessaires. Pour chacune de vos instructions, mentionner dans quelles lignes, elles vont s'insérer.

5.3 En supposant que toutes les protections nécessaires ont été incluses dans le programme, donner l'affichage obtenu en sortie après le lancement du précédent programme à un instant « i » et expliquer l'affichage obtenu.

Exercice 6 (20 points) Écrire la fonction « filtrer » qui prend un seul argument, une liste de mots passée par référence. Chaque mot est dans la forme minuscule et est composé de 3 caractères uniquement. Votre fonction doit retirer de la liste tous les mots qui ont au moins un caractère en commun. Par exemple, si la liste contenait:

```
ris, eau, tue, use, euh, son, bon, ohm, bec, est, mes, ane, web, oie, but
```

```
ris va éliminer : use son est mes oie  
eau va éliminer : tue euh bec ane web but  
bon va éliminer : ohm
```

La liste filtrée va contenir au final:

```
ris, eau, bon
```

Il est permis de décomposer votre fonction en d'autres fonctions.

Remise des notes avant le dernier cours prévu le 27 avril.
Bon Été!