

Trimestre Automne, 2018

Mohamed Lokbani

## IFT1169 – Examen Intra –

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

Nom : \_\_\_\_\_ | Prénom(s) : \_\_\_\_\_

Signature : \_\_\_\_\_ | Login : \_\_\_\_\_

Date : Vendredi 19 octobre 2018

Durée : 2 heures (de 16h30 à 18h30)

Local : Z-350, Pavillon Claire McNicoll

### Directives :

- Toute documentation est permise.
- Ordinateur, Téléphones **non** permis.
- Répondre directement sur le questionnaire.
- Les réponses **doivent être brèves, précises, claires et nettement présentées.**

1. \_\_\_\_\_ /20

2. \_\_\_\_\_ /20

3. \_\_\_\_\_ /20

4. \_\_\_\_\_ /20

5. \_\_\_\_\_ /20

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 (20 points)** Trouvez la meilleure correspondance pour les descriptions ci-dessous en y associant un nombre de 1 à 11. Le nombre ne peut-être utilisé qu'une seule fois. Nous avons complété une description à titre d'exemple.

Un \_\_\_\_\_ (1) analyste développe un ou plusieurs \_\_\_\_\_ (2) qu'il traduit en \_\_\_\_\_ (3) dans un langage donné.

Un \_\_\_\_\_ (4) est un regroupement de fichiers utilisés dans la génération d'un exécutable.

Un \_\_\_\_\_ (5) est un fichier pouvant être chargé en mémoire et exécuté.

Les fichiers \_\_\_\_\_ (6) sont des unités qu'on peut *compiler*, c'est-à-dire traduire en *code objet*.

Les fichiers \_\_\_\_\_ (7) sont des unités contenant des prototypes de fonctions, des constantes, des déclarations de types, et ainsi de suite.

Le code machine produit par la \_\_\_\_\_ (8) n'est *pas* un exécutable, mais y ressemble beaucoup.

Le \_\_\_\_\_ (9) est un *outil qui modifie le code source avant que le compilateur n'entre en oeuvre*.

\_\_\_\_\_ (10) se charge généralement des "*résolutions externes*" requises pour passer du \_\_\_\_\_ (11) à du code exécutable cohérent.

algorithmes	
code objet	
compilation	
éditeur de lien	
en-tête	
exécutable	
préprocesseur	
programme	
programmeur	<b>1</b>
projet	
source	

**Exercice 2 (20 points)** Soit la fonction générique suivante :

```
template <class T>
T Test(T a, T b){
    return a > b ? a : b ;
}
```

Pour chacun des appels suivants, donnez l'affichage obtenu en sortie, dite s'il peut générer une situation ambiguë ou incohérente. Si c'est le cas, expliquez l'incohérence et proposez une solution.

**2.1** `cout << "Test(10, 15) = " << Test(10, 15) << endl ;`

**2.2** `cout << "Test(10.1, 15.2) = " << Test(10.1, 15.2) << endl ;`

```
2.3 cout << "Test('k', 's') = " << Test('k', 's') << endl ;
```

```
2.4 cout << "Test(\"Aladdin\", \"Jasmine\") = " << Test("Aladdin", "Jasmine") << endl ;
```

**Exercice 3 (20 points)** ce programme compile et s'exécute correctement.

```
01 #include <iostream>
02 #include <string>
03 #include <algorithm>
04
05
06 struct Quelquechose{
07     void operator()(char & c) {
08         static int last = ' ';
09         if (last == ' ' && c != ' ' && ::isalpha(c))
10             c = ::toupper(c);
11         last = c;
12     }
13 };
14 int main() {
15
16     std::string data = "IFT1169 est un cours de programmation c++.";
17
18     std::for_each(data.begin(), data.end(), Quelquechose());
19
20     std::cout << data << std::endl;
21
22     return 0;
23 }
```

`isalpha` : vérifie si l'on a un caractère alphabétique

`toupper` : convertit le caractère donné en majuscule

**Que va afficher en sortie le précédent programme? Expliquez brièvement votre démarche.**



**Exercice 4 (20 points)** le but du programme est d'ajouter un mot à la fin de chacune des lignes d'un fichier donné. Le programme est exécuté comme suit :

**exo4.exe entree.txt sortie.txt Alpha**

**entree.txt** : est un fichier qu'on doit lire son contenu

**sortie.txt** : est un fichier dans lequel on écrit le résultat

**Alpha** : est le mot à ajouter à la fin de chaque ligne

Si par exemple le fichier en entrée contenait les lignes suivantes

**Un phoque attaque un kayakiste en le frappant avec une pieuvre**

Un kayakiste naviguant dans les eaux côtières de la Nouvelle-Zélande a eu droit à une drôle d'expérience avec le milieu marin néo-zélandais.

Selon le quotidien anglais The Guardian, Kyle Mulinder profitait d'une superbe journée ensoleillée au large des côtes de Kaikoura, à l'Île du Sud de la Nouvelle-Zélande. Il observait depuis quelques minutes, avec les autres membres de son groupe de kayakistes, un phoque luttant sous l'eau avec une massive pieuvre.

Sa surprise fut grande lorsque le mammifère, semblant triompher dans le combat sous-marin, émergea de l'eau avec la pieuvre en bouche.

Le résultat obtenu dans le fichier « sortie.txt »

alpha  
Un phoque attaque un kayakiste en le frappant avec une pieuvrealpha  
alpha  
Un kayakiste naviguant dans les eaux côtières de la Nouvelle-Zélande a eu droit à une drôle d'expérience avec le milieu marin néo-zélandais.alpha  
alpha  
Selon le quotidien anglais The Guardian, Kyle Mulinder profitait d'une superbe journée ensoleillée au large des côtes de Kaikoura, à l'Île du Sud de la Nouvelle-Zélande. Il observait depuis quelques minutes, avec les autres membres de son groupe de kayakistes, un phoque luttant sous l'eau avec une massive pieuvre. alpha  
alpha  
Sa surprise fut grande lorsque le mammifère, semblant triompher dans le combat sous-marin, émergea de l'eau avec la pieuvre en bouche. alpha

On vous demande de compléter les champs manquants du programme suivant.

```

01 #include <iostream>
02
03 #include _____
04
05 using namespace std;
06
07 int main(int argc, const char * argv[]){
08
09     if (argc != _____) {
10
11         _____ << "besoin fichier entrée sortie mot" << endl;
12
13         exit _____;
14
15     }
16
17     _____ in_file(argv[_____]);
18
19     if (!in_file) {
20
21         _____ << "Ne peut pas ouvrir le fichier\n";
22
23         exit _____;
24
25     }
26
27     _____ out_file(argv[_____]);
28
29     if (!out_file) {
30
31         _____ << "Ne peut pas créer le fichier de sortie\n";
32
33         exit _____;
34
35     }
36
37     _____ add_string = argv[_____];
38
39     _____ str;
40
41
42
43     while (getline(_____, _____)) {
44
45         _____ << _____ << _____ << "\n";
46
47     }
48
49     return 0;
50 }

```

**Exercice 5 (20 points)** la fonction « main » suivante est syntaxiquement correcte, elle fait partie du fichier « exo5.cpp ».

```
01 int main() {
02
03     // on initialise la map (une autre technique)
04     map<string, int> wordMap = \
05         { { "india", 6 }, { "tango", 5 }, { "hotel", 9 }, { "alfa", 6 } };
06
07     cout << "MAP avant l'appel à erase_if" << endl;
08
09     map<string, int>::iterator pos;
10     for (pos = wordMap.begin(); pos != wordMap.end(); ++pos) {
11         cout << "clé: " << pos->first << "\t" << "valeur: " \
12             << pos->second << endl;
13     }
14
15     // On efface tous les éléments ayant la valeur 6
16     int nbreeltseffaces= erase_if(wordMap, 6);
17
18     cout << "Nombre d'éléments effacés = " << nbreeltseffaces << endl;
19
20     cout << "MAP après l'appel à erase_if" << endl;
21     for (pos = wordMap.begin(); pos != wordMap.end(); ++pos) {
22         cout << "clé: " << pos->first << "\t" << "valeur: " << pos->second << endl;
23     }
24
25     return 0;
26 }
```

En exécutant le programme au complet, on obtient l'affichage suivant :

```
>exo5.exe
MAP avant l'appel à erase_if
clé: alfa      valeur: 6
clé: hotel    valeur: 9
clé: india    valeur: 6
clé: tango    valeur: 5
Nombre d'éléments effacés = 2
MAP après l'appel à erase_if
clé: hotel    valeur: 9
clé: tango    valeur: 5
>Exit code: 0
```

On vous demande d'écrire **la fonction générique** « erase\_if ». Cette fonction prend deux arguments, la map et la valeur à effacer. Elle permet d'effacer tous les éléments dans la map (1<sup>er</sup> argument) ayant une valeur donnée (2<sup>e</sup> argument). Commencez par écrire une fonction classique pour traiter l'exemple mentionné dans la fonction « main ». Par la suite, généralisez cette fonction pour traiter une map ayant des clés et des valeurs génériques, par force de même type.

