

Trimestre Hiver, 2014

Mohamed Lokbani

IFT1169 – Examen Final –

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

Nom : _____ | Prénom(s) : _____ |

Signature : _____ | Login DIRO : _____ |

Date : mardi 22 avril 2014

Durée : 3 heures (de 18h30 à 21h30)

Local : Z-350 ; Pavillon Claire McNicoll

Directives :

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

1. _____ /10 (1.1. à 1.5)

2. _____ /12 (2.1 à 2.6)

3. _____ /15 (3.1)

4. _____ /20 (4.1)

5. _____ /24 (5.1 à 5.6)

6. _____ /19 (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 (10 points) Encerclez la bonne réponse et donnez une courte explication.

1.1 [VRAI | FAUX] il faut utiliser l'option de compilation « `-std=c++98` », pour que le compilateur « `g++` » puisse compiler avec succès les nouveautés introduites dans la dernière version du standard « `C++` ».

1.2 [VRAI | FAUX] Il est possible d'utiliser la fonction « `free` » pour libérer l'espace mémoire alloué avec « `new` ».

1.3 [VRAI | FAUX] Il n'est pas possible de modifier la taille d'un tableau alloué dynamiquement avec « `new` ».

1.4 [VRAI | FAUX] La mémoire a été allouée pour un tableau avec l'appel « `new[]` » et cette même mémoire a été libérée avec l'appel « `delete` ». Le compilateur va rattraper votre erreur et générer une erreur de compilation pour non-concordance des appels.

1.5 [VRAI | FAUX] Le programme va planter si « `new` » n'arrive pas à allouer de la mémoire dans l'instruction suivante « `Test* p = new Test();` », sachant que « `Test` » est une classe quelconque.

Exercice 2 (12 points) Soit la classe générique « Z », et un ensemble de cas d'amitiés déclarés dans la classe « Z ». On vous demande d'expliquer l'impact d'une telle déclaration.

2.1

```
template class <T>
class Z {
    // ...
    friend void g1();
};
```

2.2

```
template class <T>
class Z {
    // ...
    friend void g2(Z<T>&);
};
```

2.3

```
template class <T>
class Z {
    // ...
    friend void A::g3();
};
```

2.4

```
template class <T>
class Z {
    // ...
    friend void B<T>::g4(Z<T>&);
};
```

2.5

```
template class <T>
class Z {
    // ...
    friend class W;
};
```

2.6

```
template class <T>
class Z {
    // ...
    friend class X<T>;
};
```

Exercice 3 (15 points) Le programme suivant compile et s'exécute correctement.

```
01 #include <iostream>
02
03 using namespace std;
04
05 void TestFlags() {
06     cout << cin.bad() << endl;
07     cout << cin.fail() << endl;
08     cout << cin.eof() << endl;
09     cout << endl;
10 }
11
12 int main(){
13
14     int p;
15
16     cout <<"-----\n";
17     cout <<" cin.rdstate(): "<< cin.rdstate() << endl;
18     TestFlags();
19
20     cout <<"cin.good(): "<< cin.good();
21
22     // La personne entre le caractère G
23     cout <<"\n\nEntrer un nombre entier: "<<endl;
24
25     cin >> p;
26
27     cout <<"\n\n-----\n";
28     cout <<" cin.rdstate(): "<< cin.rdstate() << endl;
29     TestFlags();
30
31     cout <<"cin.good(): "<< cin.good()<<"\n\n";
32
33     cin.clear();
34
35     cout <<"-----\n";
36     cout <<"cin.fail(): "<< cin.fail()<<endl;
37
38     return 0;
39 }
```

Tout en expliquant votre réponse, que va afficher en sortie le précédent programme, sachant qu'à la ligne « 23 », l'utilisateur a fourni le caractère « G » comme réponse.

Exercice 4 (20 points) Voici un extrait d'un programme qui compile et s'exécute correctement.

```
01 UnFrame::UnFrame(const wxString& titre) : wxFrame(NULL, wxID_ANY, titre){
02     wxPanel *panelAffichage = new wxPanel(this, -1);
03
04     wxBoxSizer *sizer_intermed = new wxBoxSizer(wxVERTICAL);
05
06     wxStaticBoxSizer *cadre = new wxStaticBoxSizer(wxVERTICAL, panelAffichage,
07     _T("Informations : "));
08
09     wxFlexGridSizer *grille = new wxFlexGridSizer(4, 2, 5, 5);
10
11     wxStaticText *labelNom = new wxStaticText(panelAffichage, -1, _T("Nom :"));
12     grille->Add(labelNom, 0, wxALIGN_CENTER_VERTICAL | wxALIGN_RIGHT);
13
14     wxTextCtrl *txtNom = new wxTextCtrl(panelAffichage, -1, _T(""));
15     grille->Add(txtNom, 0, wxEXPAND);
16
17     wxStaticText *labelPrenom = new wxStaticText(panelAffichage, -1, _T("Prénom :"));
18     grille->Add(labelPrenom, 0, wxALIGN_CENTER_VERTICAL | wxALIGN_RIGHT);
19
20     wxTextCtrl *txtPrenom = new wxTextCtrl(panelAffichage, -1, _T(""));
21     grille->Add(txtPrenom, 0, wxEXPAND);
22
23     wxStaticText *labelDate = new wxStaticText(panelAffichage, -1, _T("Date de naissance :"));
24     grille->Add(labelDate, 0, wxALIGN_CENTER_VERTICAL | wxALIGN_RIGHT);
25
26     wxTextCtrl *txtDate = new wxTextCtrl(panelAffichage, -1, _T(""));
27     grille->Add(txtDate, 0, wxEXPAND);
28
29     wxStaticText *labelComm = new wxStaticText(panelAffichage, -1, _T("Commentaires :"));
30     grille->Add(labelComm, 0, wxALIGN_CENTER_VERTICAL | wxALIGN_RIGHT);
31
32     wxTextCtrl *txtComm = new wxTextCtrl(panelAffichage, -1, _T(""), wxDefaultPosition,
33     wxDefaultSize, wxTE_MULTILINE);
34     grille->Add(txtComm, 0, wxEXPAND);
35
36     cadre->Add(grille, 1, wxALL | wxEXPAND, 5);
37     grille->AddGrowbleCol(1);
38     grille->AddGrowbleRow(3);
39
40     sizer_intermed->Add(cadre, 1, wxALL | wxEXPAND, 5);
41
42     wxStaticLine *ligneHoriz = new wxStaticLine(panelAffichage, -1);
43     sizer_intermed->Add(ligneHoriz, 0, wxALL | wxEXPAND, 5);
44
45     wxBoxSizer *sizer_boutons = new wxBoxSizer(wxHORIZONTAL);
46
47     wxButton *btnValider = new wxButton(panelAffichage, -1, _T("Valider"));
48     sizer_boutons->Add(btnValider, 0);
49
50     sizer_boutons->AddSpacer(5);
51
52     wxButton *btnAnnuler = new wxButton(panelAffichage, -1, _T("Annuler"));
53     sizer_boutons->Add(btnAnnuler, 0);
54
55     sizer_intermed->Add(sizer_boutons, 0, wxALIGN_RIGHT | wxALL, 5);
56     panelAffichage->SetSizer(sizer_intermed);
57 }
```

Tout en développant votre réponse, dessiner le résultat obtenu. À noter que l'instruction « `_T("chaîne")` » est dans le même genre que « `string z("chaîne")` ». Commencer par faire un plan de l'interface (une grille), par la suite positionner les éléments dans l'interface (chaque case de la grille) en mettant en évidence le cadrage de chaque zone, avant de fournir le résultat final (sans les cadres intermédiaires de la grille).

Exercice 5 (24 points) Le programme suivant compile et s'exécute correctement.

```
01 #include <iostream>
02 #include <map>
03 #include <algorithm>
04
05
06 using namespace std;
07
08 void Mystere(map<string, int>& m1, map<int, string>& m2) {
09     map<string, string> mapres;
10     map<string, int>::iterator pos1;
11     map<int, string>::iterator pos2;
12     map<string, string>::iterator posres;
13
14     for (pos1=m1.begin(); pos1 !=m1.end(); ++pos1){
15         pos2 = m2.find(pos1->second);
16         if (pos2 != m2.end()){
17             mapres[pos1->first] = pos2->second;
18         }
19     }
20
21     for (posres=mapres.begin(); posres !=mapres.end(); ++posres)
22         cout << posres->first << " : " << posres->second << endl;
23     cout << "-----" << endl;
24 }
25
26 int main(){
27
28     map<string,int> m1={{"chien",1},{ "chat",2},{ "lion",3},{ "vache",4}};
29     map<int,string> m2={{1,"terre"},{2,"vent"},{3,"air"},{4,"feu"}};
30     Mystere(m1,m2);
31
32     map<string,int> m3 = {{ "cinq",105}, {"quatre",104}, {"un",101},{ "six",106},{ "trois",103},{ "deux",102}};
33     map<int,string> m4 = {{99,"un"}, {101,"deux"}, {103,"trois"},{105,"quatre"}};
34     Mystere(m3,m4);
35
36     map<string,int> m5 = {{ "a",42}, {"b",9}, {"c",7},{ "d",15},{ "e",11},{ "f",24},{ "g",7}};
37     map<int,string> m6 = {{1,"quatre"}, {3,"score"}, {5,"et"},{7,"sept"},{9,"age"},{11,"rien"}};
38     Mystere(m5,m6);
39
40     return 0;
41 }
```

5.1 Tout en développant votre réponse, que va afficher en sortie l'appel de la ligne « 30 »?

5.2 Tout en développant votre réponse, que va afficher en sortie l'appel de la ligne « 34 »?

5.3 Tout en développant votre réponse, que va afficher en sortie l'appel de la ligne « 38 »?

Le programme suivant compile et s'exécute correctement.

```
01 #include <iostream>
02 #include <vector>
03 #include <iterator>
04
05 using namespace std;
06
07 void Mystere(vector<int>& unvec) {
08     for (unsigned int i = 0; i < unvec.size(); i++) {
09         int n = unvec[i];
10         unvec.erase(unvec.begin()+i);
11         if (n % 2 == 0) {
12             unvec.push_back(i);
13         }
14     }
15     copy(unvec.begin(),unvec.end(),ostream_iterator<int>(cout," "));
16     cout << endl;
17 }
18
19 int main(){
20
21     vector<int> v1 = {5, 2, 5, 2};
22     Mystere(v1);
23
24     vector<int> v2 = {3, 5, 8, 9, 2};
25     Mystere(v2);
26
27     vector<int> v3 = {0, 1, 4, 3, 1, 3};
28     Mystere(v3);
29
30     return 0;
31 }
```

5.4 Tout en développant votre réponse, que va afficher en sortie l'appel de la ligne « 22 »?

5.5 Tout en développant votre réponse, que va afficher en sortie l'appel de la ligne « 25 »?

5.6 Tout en développant votre réponse, que va afficher en sortie l'appel de la ligne « 28 »?

Exercice 6 (19 points) Le fichier viande.txt contient ce qui suit:

Filet mignon: 1
Bavette: 7
Filet de porc: 2
Steak: 3
Côtelette d'agneau: 3
Bacon: 3
Cuisse de poulet: 2
Foie gras de canard: 2
Jambon: 2

Le premier élément du fichier représente le type de la viande, le second élément représente le stock disponible en nombre de jours. On vous demande d'écrire un programme qui réalise ce qui suit:

-1- Le programme prend en entrée le nom du fichier et l'option A. On peut avoir donc les deux appels suivants:

exo6.exe viande.txt
exo6.exe viande.txt A

Si le 2e argument est présent (A), le programme doit ajuster le stock des viandes disponibles. Voir pour cela le point -5-.

Le programme doit donc vérifier la présence des arguments, ouvrir le fichier viande.txt et lire son contenu. En cas d'erreur, le programme doit s'arrêter avec un message approprié.

-2- Les données doivent être préservées dans un conteneur approprié.

-3- Trie le conteneur par le nom de la viande dans un ordre alphabétique (Bacon, Bavette, etc.).

-4- Affiche le contenu du conteneur sur la sortie standard.

-5- Si l'option A est présente, ajuste le stock comme suit:

Ancienne valeur	Nouvelle valeur
Val = 1	Val = 2
Val = 2	Val = 4
Val > 2	Val = Val + 3

-6- Récupère le conteneur sur le nombre de jours dans un ordre croissant (produit avec le moins de jours en premier)

-7- Affiche le contenu du conteneur sur la sortie standard.

Lors de la correction, je vais tenir compte de la clarté de votre code, de son optimisation, de l'utilisation maximale de tout ce qui gravite autour des STL (conteneurs, algorithmes, etc.). SVP, évitez comme la peste, les boucles « for », « while », « do », etc.

