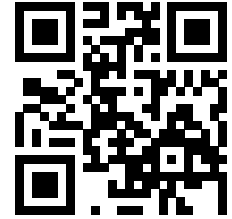


NOM : Hanon Ymous
(000000)
Place : 0

#0000



Programmation I (SMA/SPH) : EXAMEN Semestre 1

22 décembre 2016

INSTRUCTIONS (à lire attentivement)

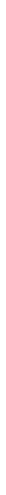
IMPORTANT! Veuillez suivre les instructions suivantes à la lettre sous peine de voir votre série annulée dans le cas contraire.

1. Vous disposez d'une heure quarante-cinq minutes pour faire cet examen (10h15 – 12h00).
2. Vous devez **écrire à l'encre noire ou bleu foncée**, pas de crayon ni d'autre couleur.
3. Vous avez droit à toute documentation papier.
En revanche, vous ne pouvez pas utiliser d'ordinateur personnel, ni de téléphone portable, ni aucun autre matériel électronique.
4. Répondez aux questions directement sur la donnée ; utilisez aussi le verso des feuilles, **MAIS** n'utilisez *que* le verso de la feuille sur laquelle se trouve la question, et non **pas** celui de la feuille précédente!
Ne joignez aucune feuilles supplémentaires ; **seul ce document sera corrigé**.
5. Lisez attentivement et *complètement* les questions de façon à ne faire que ce qui vous est demandé. Si l'énoncé ne vous paraît pas clair, ou si vous avez un doute, demandez des précisions à l'un des assistants.
6. L'examen comporte quatre exercices indépendants, qui peuvent être traités dans n'importe quel ordre, mais qui ne rapportent pas la même chose (les points sont indiqués, le total est de 84) :
(1) questions de cours : 21 points ; (2) conception : 24 points ; (3) chercher les erreurs : 24 points et (4) déroulement de programme : 15 points.
Tous les exercices comptent pour la note finale.



Question 1 – Questions de cours [sur 21 points]

- 1.1 [2 points] Dans l'espace de dimension 3, un point est représenté par ses trois coordonnées, et un segment de droite est défini par deux points. Définissez les types C++ pour représenter ces deux notions (point et segment) :



- 1.2 [3 points] On a la fonction de prototype suivant :

```
void printMovie(string name, int year = 2016,  
               string country = "USA", string genre = "comedie" );
```

Pour chacun des films suivants, écrivez l'appel à la fonction `printMovie` *en utilisant le plus de valeur par défaut possible* :

— « Les Visiteurs 25 », 2016, France, comédie :

— « Zootopia », 2016, USA, comédie :


— « Arrival », 2016, USA, drame :



Question 1

1.3 [5 points] La fonction suivante est-elle correcte? Critiquez et corrigez (directement sur le code) tous les points qui vous semblent nécessaires. Expliquez (brièvement) vos critiques.

```
void add_matrices(const vector<vector<double>>& A,
                 const vector<vector<double>>& B,
                 const vector<vector<double>>& C
                 )
{
    for (size_t i(1); i <= A.size(); ++i) {
        for (size_t j(1); j <= B.size(); ++j) {
            C[i][j] = A[i][j] + B[i][j];
        }
    }
    return C;
}
```

suite au dos 



- 1.4 [6 points] Qu'affiche le programme suivant si l'on entre « HoHoHoHoHoHo » puis « oH » ?
Et si l'on entre « oH » puis « HoHoHoHoHoHo » ?
Justifiez brièvement vos deux réponses.

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

int f(string const& x, string const& y)
{
    int z(0);

    if (x.size() < y.size()) throw string("Non!");

    for (size_t i(0); i + y.size() < x.size(); ++i) {
        if (x.substr(i, y.size()) == y) {
            cout << i << " ";
            ++z;
        }
    }
    cout << endl;

    return z;
}

int main()
{
    string a, b;
    cin >> a >> b;

    try {
        cout << "** ";
        cout << f(a,b) << " **" << endl;
        cout << "Voilà!" << endl;
    }
    catch (string& err) {
        cerr << err << endl;
    }

    return 0;
}
```



1.5 [5 points] Quel est le résultat de l'appel $g(6)$ avec la fonction g suivante ? Expliquez votre réponse.

```
int g(int a) {  
    if (a <= 1) return 1;  
    return g(a/2) + g(a-1);  
}
```

suite au dos 



Question 2 – Conception [sur 24 points]

On vous demande de concevoir puis d'écrire certaines parties d'un programme C++ permettant de proposer des recettes de cuisine à partir d'ingrédients possibles (pas forcément tous utilisés, ce sont des suggestions).

Dans ce programme, chaque recette a :

- un nom,
- une liste d'ingrédients,
- un type de repas (comme « soupe », « salade », « plat de viande », « dessert », etc. ; simplement sous forme d'une chaîne de caractères),
- un type de cuisine (comme « chinoise », « française », « suisse », etc. ; également sous forme d'une chaîne de caractères)
- une indication si la recette est végétarienne ou non,
- son temps de préparation en minutes.

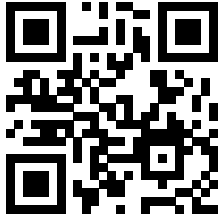
Chaque ingrédient est caractérisé par son nom, sa quantité (un nombre et une unité, séparément ; par exemple : 5, « ml ») et le nombre de calories par unité de quantité (par exemple : 12 pour dire « 12 calories par ml »).

1. [8 points] Définissez les structures de données nécessaires à un tel programme.
 2. [5 points] Coder la fonction `calories` qui calcule le total de calories d'une recette (simplement comme la somme des calories de ses ingrédients).
 3. [3 points] Supposez que vous ayez :
 - une variable `livre` contenant un ensemble de recettes (au sens défini ci-dessus) ;
 - une variable `frigo` contenant divers ingrédients (au sens défini ci-dessus) ;
 - et une fonction `cherche_recettes` qui retourne la liste des recettes du `livre` qui satisfont les critères donnés (type de repas, de cuisine, végétarienne ou non, temps maximum de préparation) et que l'on peut faire à partir du contenu du `frigo`.Comment écririez vous l'*appel* à cette fonction `cherche_recettes` pour rechercher toutes les recettes (présentes dans le `livre`) de soupes chinoises végétariennes que l'on peut préparer en moins de 45 minutes ?
 4. [8 points] En supposant
 - qu'il existe une fonction `sont_contenus` qui prend deux listes d'ingrédients L_1 et L_2 (dans cet ordre) et retourne `true` si L_1 est incluse dans L_2 (y compris ses quantités),
 - et que la fonction `cherche_recettes` recherche par défaut des plats suisses non végétariens que l'on peut préparer en moins d'une heure,écrivez le code de la fonction `cherche_recettes`.
-

Question 2

Anonymisation : #0000
p. 7



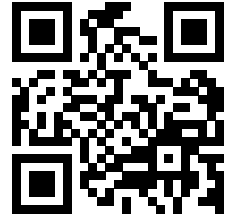


Anonymisation : #0000
p. 8

Question 2

Question 2

Anonymisation : #0000
p. 9





Question 3 – Chercher les erreurs [sur 24 points]

Le programme ci-contre (pages suivantes) a pour but de simuler la gestion de ressources en eau. Voici le déroulement attendu :

Water supplies:

```
1) ballon 1      : 100 l
2) reserve       : 50 l
3) ballon 2      : 200 l
4) source        : 400 l
5) bassin        : 250 l
```

Adding 100 l to ballon 2: now 300 l.

Asking for 200 l:

```
Got 100 l from ballon 1 (now empty)
Got 50 l from reserve (now empty)
Got 50 l from ballon 2 (remains 250 l)
```

I got 200 l of water.

Water supplies:

```
1) ballon 1      : 0 l
2) reserve       : 0 l
3) ballon 2      : 250 l
4) source        : 400 l
5) bassin        : 250 l
```

Il y a néanmoins huit erreurs de programmation de différente nature.

(Note : il y a des erreurs que le compilateur ne voit pas.)

Le compilateur g++ (6.2), avec toutes les aides de compilation (option -Wall), sort des pages de messages d'erreur dont voici le début :

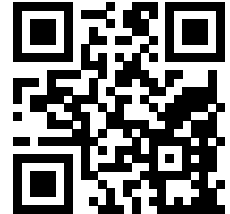
```
error.cc:10:1: error: expected ';' after struct definition
}
~

error.cc: In function 'bool get_water(std::vector<WaterSupply>&, double)':
error.cc:32:26: error: invalid use of member function 'std::vector<_Tp, _Alloc>::size_type
std::vector<_Tp, _Alloc>::size() const [with _Tp = WaterSupply; _Alloc =
std::allocator<WaterSupply>; std::vector<_Tp, _Alloc>::size_type = long unsigned int]'
(did you forget the '(') ?
    for (int i(0); (i < ws.size) and not drew; ++i) {
        ~~~~~

error.cc:33:19: error: 'class std::vector<WaterSupply>' has no member named 'water'
    double dif(ws.water - amount);
        ~~~~~

error.cc:54:1: warning: no return statement in function returning non-void [-Wreturn-type]
}
~

error.cc: In function 'int main()':
error.cc:79:25: error: no matching function for call to 'WaterSupply::WaterSupply(int)'
    WaterSupply bath(200);
        ~
```



Question 3

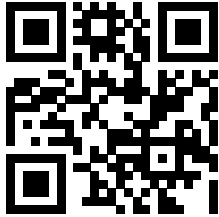
Indiquez et corrigez toutes les erreurs (directement sur le code).

Expliquez brièvement les erreurs/correction à côté du code ou sur la page supplémentaire.

On ôtera 1 point pour toute indication d'une erreur qui n'en est pas une.

```
1  #include <iostream>
2  #include <iomanip>
3  #include <string>
4  #include <vector>
5  using namespace std;
6
7  struct WaterSupply {
8      string name;
9      double water;
10 }
11
12 void add_water(WaterSupply ws, double amount)
13 {
14     cout << "Adding " << amount << " l to " << ws.name << ':';
15     ws.water += amount;
16     cout << " now " << ws.water << " l." << endl;
17 }
18
19 void refill_water_supplies(vector<WaterSupply> ws, double amount)
20 {
21     for (auto w : ws) {
22         if (w.water < amount) {
23             add_water(w, 10 * amount * (1 - w.water/amount));
24         }
25     }
26 }
27
28 bool get_water(vector<WaterSupply>& ws, double amount)
29 {
30     cout << "Asking for " << amount << " l." << endl;
31     bool drew(false);
32     for (int i(0); (i < ws.size) and not drew; ++i) {
33         double dif(ws.water - amount);
34         if (dif >= 0) {
35             ws[i].water -= amount;
36             drew = true;
37             cout << " Got " << setw(3) << amount << " l from " << ws[i].name
38                 << " (remains " << ws[i].water << " l)" << endl;
39         }
40     }
```

suite au dos 



```
40     else {
41         if (ws[i].water > 0) {
42             cout << " Got " << setw(3) << ws[i].water << " l from " << ws[i].name
43                 << " (now empty)" << endl;
44         }
45         ws[i].water = 0;
46         amount = -dif;
47     }
48 }
49 if (not drew) {
50     cout << "Not enough water. Refilling!" << endl;
51     refill_water_supplies(ws, amount);
52     drew = get_water(ws, amount);
53 }
54 }
55
56 void display_supplies(vector<WaterSupply>& ws)
57 {
58     cout << "Water supplies:" << endl;
59     int i(1);
60     for (auto w: ws) {
61         cout << right << setw(3) << i << " ) "
62             << left << setw(14) << w.name << ": "
63             << right << setw(4) << w.water << " l"
64             << endl;
65         ++i;
66     }
67 }
68
69 int main(){
70     vector<WaterSupply> ws = {
71         { "ballon 1", 100 },
72         { "reserve", 50 },
73         { "ballon 2", 200 },
74         { "source", 400 },
75         { "bassin", 250 }
76     };
77     display_supplies(ws);
78     add_water(ws[2], 100);
79     WaterSupply bath(200);
80     bool got_water(get_water(ws, bath));
81     if (got_water){
82         cout << "I got " << bath << " l of water." << endl;
83     }
84     display_supplies(ws);
85     return 0;
86 }
```

Question 3

Anonymisation : #0000
p. 13





Question 4 – Déroulement de programme [sur 15 points]

Voici (sur deux pages) un code d'étudiant bien modularisé, mais malheureusement mal nommé...

Expliquez comment se déroule l'*exécution* du code ci-dessous. Qu'affiche-t-il ?

On vous demande bien ici d'expliquer les *principales* étapes du déroulement, pas de paraphraser le code en français. Proposez également un nom pertinent pour les deux fonctions mal nommées et pour la structure S.

```
1  #include <iostream>
2  #include <algorithm>
3  #include <vector>
4  #include <string>
5  using namespace std;
6
7  struct S {
8      string signature;
9      vector<string> w;
10 };
11
12 string sort_string(string str) {
13     sort(str.begin(), str.end()); // trie les caractères de str
14     return str;
15 }
16
17 string mauvais_nom(string str) {
18     size_t i(0);
19
20     while (i < str.size()-1) {
21         if (str[i] == str[i+1]) {
22             str.replace(i, 1, "");
23         } else {
24             ++i;
25         }
26     }
27     return str;
28 }
29
30 void ajoute(string const& w, vector<S>& v) {
31     const string wsign(mauvais_nom(sort_string(w)));
32     bool added(false);
33     for (size_t i(0); (i < v.size()) and not added; ++i) {
34         if (v[i].signature == wsign) {
35             v[i].w.push_back(w);
36             added = true;
37         }
38     }
39     if (!added) {
40         v.push_back( { wsign, { w } } );
41     }
42 }
```



Question 4

```
43
44 vector<S> fait_qqchose(vector<string> const& tab) {
45     vector<S> v;
46     for(auto w : tab) {
47         ajoute(w, v);
48     }
49     return v;
50 }
51
52 void print(vector<S> const& v) {
53     for (auto c : v) {
54         for (auto w : c.w) {
55             cout << w << " ";
56         }
57         cout << endl;
58     }
59 }
60
61 int main() {
62     print( fait_qqchose({
63         "aimer",
64         "aspirine",
65         "image",
66         "magie",
67         "maire",
68         "paniers",
69         "parisien"
70     }));
71     return 0;
72 }
```



Anonymisation : #0000
p. 16

Question 4
