

Ne PAS retourner ces feuilles avant d'en être autorisé!

Merci de poser votre carte CAMIPRO en évidence sur la table.

Vous pouvez déjà compléter ci-dessous

NOM en MAJUSCULES _____

Prénom _____

Numéro SCIPER _____

Signature _____

BROUILLON : Ecrivez aussi votre NOM-Prénom sur la feuille de brouillon fournie. Toutes vos réponses doivent être sur cette copie d'examen. Les feuilles de brouillon sont ramassées pour être immédiatement détruites.

Le contrôle écrit commence à :

15h15

A **16h45** retourner les feuilles avec la page 1 face à vous et faire glisser les copies vers les allées pour être ramassées. Attendre en silence le signal pour sortir.

*les contrôles écrits de PROGRAMMATION
sont SANS appareil électronique.*

Vous avez le droit d'avoir tous vos documents **personnels** sous forme papier: dictionnaire, livres, cours, exercices, code, projet, etc...

Vous pouvez utiliser un crayon à papier et une gomme

Ce contrôle écrit permet d'obtenir **19 points** sur un total de 100 points pour le cours complet.

1) (3 pts) Compléter la fonction `singleNumber()`

Détection de la valeur qui n'apparaît qu'une seule fois dans un vector

La fonction `singleNumber` reçoit en paramètre un vector non-trié contenant au moins 3 éléments. Nous savons aussi qu'il contient *une valeur qui n'apparaît qu'une seule fois* tandis que les autres valeurs apparaissent en double. Les valeurs en double ne sont pas forcément consécutives dans le vector.

La fonction `singleNumber` doit renvoyer la valeur qui n'apparaît qu'une seule fois. Par exemple, si on fournit un vector contenant ces valeurs {5, 3, 1, 2, 2, 5, 1} alors la fonction renvoie 3 car c'est la seule valeur qui n'apparaît qu'une seule fois.

```
1 // on suppose que les #include nécessaires sont faits
2 int singleNumber(vector<int>& nums)
3 {
4     // sorting the numbers from minimum to maximum
5     sort(nums.begin(), nums.end());
6
7     for (int i = 0; i < nums.size()-1; __ ① __ )
8     {
9         if (nums[i] != nums[i + 1])
10        {
11            return __ ② __;
12        }
13    }
14
15    return nums[nums.size()-1];
16 }
```

Choisir parmi une des réponses proposées en justifiant votre réponse par deux à trois phrases ; n'hésitez pas à faire des dessins comme support à votre explication :

- A: ① : `i = i + 1` et ② : `nums[i]`
- B: ① : `i = i + 1` et ② : `nums[i+1]`
- C: ① : `i = i + 2` et ② : `nums[i]`
- D: ① : `i = i + 2` et ② : `nums[i+1]`

Réponse choisie, avec sa JUSTIFICATION :

.....

.....

.....

2) (4 pts) Fonction récursive f ()

Le programme suivant donne un résultat correct quand la fonction récursive f () est appelée avec des valeurs v1 et v2 appartenant à l'intervalle] 0, 2¹⁶[.

```
1  #include <iostream>
2  using namespace std;
3
4  unsigned int f(unsigned int v1, unsigned int v2);
5
6  int main ()
7  {
8      unsigned int v1, v2;
9
10     do
11     {
12         cin >> v1 >> v2 ;
13     }while(v1==0 or v2==0);
14
15     cout << f(v1, v2);
16
17     return 0;
18 }
19
20 unsigned int f(unsigned int v1, unsigned int v2)
21 {
22     if(v2 > 1)
23         return f(2*v1, v2/2) + ((v2%2)? v1 : 0);
24
25     return v1;
26 }
```

2.1) Indiquer la valeur renvoyée par f () pour les appels suivants en détaillant les appels récursifs (valeur des paramètres, valeur renvoyée) s'il y en a.

f (1, 1) renvoie :	f (1, 2) renvoie :
--------------------	--------------------

<p>f (2, 3) renvoie :</p>	<p>f (7, 3) renvoie :</p>
----------------------------------	----------------------------------

2.2) Quel est le but de la fonction f () ?

.....

.....

.....

2.3) Maintenant supposons que v1 et v2 ont pour valeur 2¹⁶ qui est juste en dehors du domaine autorisé pour les paramètres de f(). En déduire (sans détailler l'appel de f()) la valeur renvoyée par f() ; pourquoi obtient-on ce résultat ?

.....

.....

.....

.....

3) (4 pts) String : que fait la fonction g ()

Le programme suivant donne un résultat correct.

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  string g(vector<string> &t)
7  {
8      if(t.size() == 0)
9          return "";
10     else if(t.size() == 1)
11         return t[0];
12
13     size_t len = t[0].size();
14     for(size_t i = 1; i<t.size();i++)
15     {
16         len = min(len, t[i].size());
17     }
18
19     string str = "";
20     size_t j = 0;
21     while(j < len)
22     {
23         size_t i = 0;
24         while((i < t.size()-1) && (t[i][j] == t[i+1][j]))
25             i++;
26
27         if(i == t.size() -1)
28             str += t[0][j];
29         else
30             break;
31
32         j++;
33     }
34     return str;
35 }
36
37 int main()
38 {
39     vector<string> tab = {"monday", "monkey", "monster"};
40     cout << g(tab) << endl;
41     return 0;
42 }
```

3.1) Quel est l'affichage obtenu à la ligne 40 avec cout :

.....
.....

3.2) Justifier votre réponse du 3.1) en indiquant comment la fonction **g()** traite le paramètre formel **t** pour obtenir ce résultat. Indiquer les résultats intermédiaires s'il y en a. Conclure en indiquant le but de cette fonction :

.....

.....

.....

.....

.....

.....

.....

.....

.....

3.3) Que se passe-t-il si on remplace la boucle des lignes 24-25 par celles-ci :

```
while(i < t.size()-1)
    if(t[i][j] == t[i+1][j])
        i++;
```

Préciser si une erreur est signalée par le compilateur ou pas. Si oui laquelle.
Si non que se passe-t-il à l'exécution ?

.....

.....

.....

.....

.....

.....

.....

4) (2 pts) évaluation d'expression

Donner la valeur affichée par l'instruction cout. Montrer les priorités mises en œuvre par le langage C++ en soulignant les sous-expressions et en indiquant la valeur et le type des calculs intermédiaires.

```
cout << (((7 << 2)& 48) - 16 ? 1 ^ 2 : 4 | 2) << endl;
```

5) (6 pts) Application de gestion d'un hotel contenant une erreur sémantique:

Le programme suivant compile correctement avec C++11.

Ce programme de gestion d'un hotel doit permettre :

- **d'ajouter** un client dans une des 10 chambres numérotées de 0 à 9 (commande **Add** indiquée en fournissant la lettre **A**)
- de **libérer** une des chambres déjà occupées (commande **Empty** indiquée en fournissant la lettre **E**)
- **d'afficher** la liste des noms de clients qui ont une chambre actuellement dans l'hotel (commande **Display** indiquée en fournissant la lettre **D**).

Nous supposons qu'un nom de client est limité à un seul mot (ça n'est pas l'erreur recherchée).

5.1) Tout d'abord examinons un comportement *correct* du programme au niveau de la fonction **readRoomNumber**. Expliquer comment cette fonction s'exécute si on fournit la valeur **-1** au clavier quand elle lit la variable **roomNumber** :

.....

.....

.....

.....

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5
6  struct Room
7  {
8      string client;
9      bool occupied;
10 };
11
12 void addClient(vector<string>& client, vector<Room>& hotel);
13 void emptyRoom(vector<string>& client, vector<Room>& hotel);
14 void displayClient(const vector<string>& client);
15 void readRoomNumber(unsigned int& roomNumber);
16
17 int main()
18 {
19     vector<string> client;
20     const unsigned int nb_room(10);
21     vector<Room> hotel(nb_room);
22
23     while(true)
24     {
25         cout << " Add Client(A), Empty Room(E), Display (D): ";
26         char command;
27         cin >> command;
28         switch(command)
29         {
30             case 'A': addClient(client,hotel);
31                     break;
32             case 'E': emptyRoom(client,hotel);
33                     break;
34             case 'D': displayClient(client);
35                     break;
36             default: cout << endl << "End of program" << endl;
37                     return 0;
38         }
39     }
40     return 0;
41 }
42
43 // Display all client names in the terminal
44 void displayClient(const vector<string>& client)
45 {
46     cout << endl;
47     for(auto c : client)
48         cout << c << endl;
49 }
50
51
52 // Read a room number in [0, 9]
53 void readRoomNumber(unsigned int& roomNumber)
54 {
55     do
56     {
57         cout << "Provide a Room Number in [0, 9]: " ;
58         cin >> roomNumber;
59     } while(roomNumber > 9);
60 }
61
62 // Suite du code sur la page suivante / Please read next page
63

```

```

1 // Read a client name (supposed to be a single word) and a room
2 // number. Then update the client list and hotel Room if the
3 // room is not currently occupied
4 void addClient(vector<string>& client, vector<Room>& hotel)
5 {
6     string clientName("");
7     cout << "Provide client name (single word): " << endl;
8     cin >> clientName;
9
10    unsigned int roomNumber(0);
11    readRoomNumber(roomNumber);
12
13    if(hotel[roomNumber].occupied != true)
14    {
15        hotel[roomNumber].occupied = true;
16        hotel[roomNumber].client = clientName;
17        client.push_back(clientName);
18    }
19    else
20    {
21        cout << "Sorry but this room is occupied" << endl;
22    }
23 }
24
25 // Read the room number to liberate ; if this room is occupied,
26 // remove the client name from the list of clients
27 // and update the hotel Room.
28 void emptyRoom(vector<string>& client, vector<Room>& hotel)
29 {
30    unsigned int roomNumber(0);
31    readRoomNumber(roomNumber);
32
33    if(hotel[roomNumber].occupied == true)
34    {
35        hotel[roomNumber].occupied = false;
36        hotel[roomNumber].client="";
37        client.pop_back();
38    }
39    else
40    {
41        cout << "This room is NOT occupied" << endl;
42    }
43 }

```

5.2) Maintenant on s'intéresse à un contexte pour lequel l'erreur sémantique n'a aucun impact (c'est comme si elle n'existait pas). Pour cela indiquer les actions effectuées par le programme lorsque le fichier **test1.txt** est redirigé sur l'entrée par défaut.

Inutile de ré-écrire les messages d'invitation

Contenu de test1.txt	Action effectuée...	...et affichage <i>résultant</i> éventuel
A DUBOIS 5		
D		
E 5		
D		
Q		

5.2) C'est maintenant que le contexte révèle l'erreur sémantique recherchée. Pour documenter le problème, indiquer l'action effectuée par le programme lorsque le fichier **test2.txt** est redirigé sur l'entrée par défaut.

Inutile de ré-écrire les messages d'invitation

Contenu de test2.txt	Action du programme...	...et affichage <i>résultant</i> éventuel
A DUMONT 0		
A DUVAL 8		
A DURAND 3		
D		
E 3		
D		
A DUPRAZ 2		
E 8		
D		
Q		

Où se trouve l'erreur sémantique dans le code (fonction + N° ligne) :

Pourquoi ?.....

Que faudrait-il faire pour la corriger :

.....

.....

.....

.....