

Petite révision

Quelques petits exercices pour vous rafraîchir la mémoire et vous remettre en forme. Dans les exercices pour lesquels on vous demande de programmer, cherchez à produire un résultat à la fois élégant et efficace.

Il est très probable que vous arriviez à de (bien) meilleures solutions en fin de session qu’au début en tentant de résoudre les mêmes problèmes. Je vous suggère de conserver vos résultats de début de session et de refaire l’exercice peu après Noël, pour voir à quel point vous aurez cheminé cet automne.

Un solutionnaire complet sera rendu disponible sur le site du cours une fois que vous aurez eu le temps de vous débattre un peu avec ces problèmes. Le solutionnaire n’utilisera pas les techniques qui seront enseignées durant la session, et devrait donc être à votre portée avant le début du cours.

Dans les pages qui suivent, vous trouverez parfois écrit C/ C++. Ne considérez pas cela comme s’il s’agissait d’un même langage, mais bien comme une indication du fait que la question s’applique à ces deux langages.

Q00 – Quelques petits vrai ou faux

Les questions qui suivent ont en partie trait au langage C++ mais s'appliquent aussi (en partie, du moins) au langage C.

		Vrai	Faux
Q00.0	En C++, un tableau est une structure de données.	☐	☐
Q00.1	En C++, un tableau est un objet.	☐	☐
Q00.2	Il est possible de programmer une pile sans avoir recours à de l'allocation dynamique de mémoire.	☐	☐
Q00.3	Il est possible de programmer une pile en ayant recours à de l'allocation dynamique de mémoire.	☐	☐
Q00.4	Il est possible de programmer une file sans avoir recours à de l'allocation dynamique de mémoire.	☐	☐
Q00.5	Il est possible de programmer une file en ayant recours à de l'allocation dynamique de mémoire.	☐	☐
Q00.6	Que vaut l'expression <code>0=='\0'</code> ?	☐	☐
Q00.7	Qu'affichera l'extrait de code suivant?	☐	☐
	<pre>#include <iostream> int main() { using std::cout; if(3) cout << "Vrai"; else cout << "Faux"; }</pre>		
Q00.8	Qu'affichera l'extrait de code suivant?	☐	☐
	<pre>#include <iostream> int main() { using std::cout; int val = 0; if(val = 3) cout << "Vrai"; else cout << "Faux"; }</pre>		

Q01 – Typique...

Q01.0 En C++, quel est le type de chacun des littéraux suivants?

Littéral	Type
3	
"allo"	
1.0f	
-8	
"A"	
'A'	
4.0	
7L	
L"allo"	
L'A'	
3U	
3UL	
0x0a	
12.0L	
"3"s	
3s	

Q01.1 En C++, Java, C#, *etc.*, quelle est la valeur entière, exprimée en base 10, des littéraux suivants?

Littéral	Valeur
3	
0x0a	
0x000a	
0xff	
03	
8	
2.7f	
08	

Q02 – Questions à adresser

Soit la fonction `full_cool()` qui prend en paramètre l'adresse de deux booléens, compare les booléens pointés par chacun, et retourne `true` seulement s'ils sont de même valeur.

Q02.0 Écrivez le prototype de cette fonction.

Q02.1 Écrivez la définition de cette fonction.

Q02.2 Écrivez un petit programme qui appelle cette fonction correctement.

Q03 – Comprendre certains trucs (partie 0)

Soit le programme C++ suivant :

```
int main() {  
    const char *texte = "Coucou";  
}
```

Combien de *bytes* sont réservés en mémoire par le compilateur pour la séquence de caractères pointée par `texte` dans cet exemple?

Q04 – Comprendre certains trucs (partie 1)

Soit le programme C++ suivant :

```
int main() {  
    const char *texte = "Coucou";  
    texte = "Yo";  
}
```

Ce programme entraîne-t-il une fuite de mémoire? Expliquez votre réponse. Si vous estimez qu'il y a problème, comment devrions-nous le corriger?

Q05 – Comprendre certains trucs (partie 2)

Soit le programme C++ suivant :

```
int main() {  
    int *tableau = new int[1000];  
    delete tableau;  
}
```

L'exécution de ce programme entraîne-t-elle une fuite de mémoire? Expliquez votre réponse. Si vous estimez qu'il y a problème, expliquez comment il devrait être corrigé.

Q06 – Comprendre certains trucs (partie 3)...

Soit le programme C++ suivant :

```
#include <iostream>
int main() {
    using std::cin;
    const char *t = "Wow";
    char c;
    cin >> c;
    while (c != '.') {
        t += c;
        cin >> c;
    }
}
```

Ce programme fonctionne-t-il correctement et sans danger? Expliquez votre réponse. Si vous estimez qu'il y a problème, expliquez comment il devrait être corrigé.

Q07 – Comprendre certains trucs (partie 4)...

Soit le programme C++ suivant :

```
int main() {  
    char t0[] = "Wow";  
    char t1[] = "You";  
    for (int i = 0; t1[i] != '\0'; i++)  
        t0[i] = t1[i];  
}
```

Ce programme fonctionne-t-il correctement et sans danger? Expliquez votre réponse. Si vous estimez qu'il y a problème, expliquez comment il devrait être corrigé.

Q08 – Comprendre certains trucs (partie 5)...

Soit le programme C# suivant :

```
// ...  
static void Main(string [] args)  
{  
    string s = "J'aime mon prof!";  
    // ...  
}  
// ...
```

Complétez ce programme de manière à ce qu'il contienne une `string` nommée `s2` telle que `s2` soit une copie de `s`. Expliquez ce que vous voulez dire par « une copie de ».

Q09 – Comprendre certains trucs (partie 6)...

Soit le programme C# suivant :

```
// ...
class Personne
{
    string nom;
    public string Nom
    {
        get; private set;
    }
    public Personne(string nom)
    {
        Nom = nom;
    }
}
// ...
static void Main(string [] args)
{
    string s = "Bill";
    Personne p = new Personne(s);
    s = "Bob";
    // ...
}
// ...
```

Dans la méthode `Main()`, est-ce que la modification de `s` modifiera le nom de `p`? Expliquez votre réponse.

Q10 – Et on enchaîne!

Une chaîne `ASCIIZ` (chaîne de caractères du langage C) est une séquence contiguë en mémoire de caractères délimités à la fin par un caractère spécial (un *byte* de valeur 0, souvent écrit `'\0'` pour *valeur zéro sur huit bits*). On manipule habituellement une chaîne `ASCIIZ` à partir d'un pointeur sur son premier élément.

Sachant cela...

Q10.0

Écrivez la fonction `lg_chaine` qui prend en paramètre un pointeur sur le premier élément d'une chaîne `ASCIIZ` et qui retourne le nombre de caractères dans la séquence (excluant le délimiteur de la fin), sous la forme d'un entier non signé. Vous n'avez pas le droit d'utiliser d'objets ou d'appeler une fonction toute faite comme `strlen()`.

Q10.1

Écrivez la fonction `comp_chaines()` qui prend en paramètre deux chaînes `ASCIIZ`, réalise une comparaison lexicographique de ces chaînes (respectant l'ordre des codes `ASCII`) et retourne un entier négatif si la première chaîne est plus petite que la seconde, retourne un entier positif si la première est plus grande que la seconde, et retourne 0 si elles sont identiques. Vous n'avez pas le droit d'utiliser d'objets ou d'appeler une fonction toute faite comme `strcmp()`.

Q10.2

Plus difficile : écrivez la fonction `copier_chaine()` qui prend en paramètre deux chaînes `ASCIIZ` et copie les caractères de la seconde sur ceux de la première. Vous n'avez pas le droit d'utiliser d'objets ou d'appeler une fonction toute faite comme `strcpy()`. À quoi faut-il faire attention? Qu'indiqueriez-vous comme avertissements dans les commentaires documentant cette fonction? Pouvez-vous utiliser `new`? Réfléchissez, car c'est délicat.

Q10.3

Plus difficile : écrivez la fonction `concat_chaines()` qui prend en paramètre deux chaînes `ASCIIZ` et ajoute le texte de la seconde à la fin de la première. Le résultat retourné (la chaîne `ASCIIZ` concaténée) doit être une chaîne `ASCIIZ`. Vous n'avez pas le droit d'utiliser d'objets ou d'appeler une fonction toute faite comme `strcat()`. À quoi faut-il faire attention? Qu'indiqueriez-vous comme avertissements dans les commentaires documentant cette fonction? Pouvez-vous utiliser `new`? Réfléchissez, car c'est délicat.

Q10.4

Un(e) collègue vous propose la fonction `afficher_majuscules()` qui utilise votre fonction `lg_chaine()`. Son code va comme suit :

```
#include <iostream>
#include <locale>
// ...
void afficher_majuscules(const char *p) {
    using namespace std;
    for (int i = 0; i < lg_chaine(p); )
        cout << toupper(p[i++], locale{ "" });
}
```

Il se trouve que cette procédure est lente lorsque la chaîne est longue. Il/ elle prétend que c'est la faute de votre fonction (après tout, ça ne peut pas être de sa faute puisque sa fonction est toute simple). Êtes-vous de son avis? Expliquez pourquoi...

Q10.5

Trouvez une autre stratégie que celle utilisant une séquence délimitée de caractères pour représenter une chaîne de caractères. Vous n'avez pas le droit d'utiliser d'objets. Offrez sous forme de fonctions globales une version des fonctions `lg_chaine()`, `comp_chaines()`, `copier_chaine()` et `concat_chaines()` pour cette structure de données. Quels sont les avantages de votre stratégie sur la stratégie ASCII? Quels en sont les inconvénients? Le code de `afficher_majuscules()` de votre collègue à la question Q10.4 serait-il plus rapide? Devrait-il être modifié?