

CS-119(a) – ICC-C Série 7

2024-04-09

Rappel Pour faire lire des valeurs d'un fichier texte à votre programme il suffit d'utiliser la redirection de l'entrée :

```
./exo < fichier.txt
```

Exo1

Qu'affiche ce code ? Réfléchissez-y sans le faire tourner dans un premier temps.

```
1 #include <stdio.h>
2
3 int inspect(int *tab, int size)
4 {
5     int big = tab[0];
6     for (int i = 1; i < size; i++)
7     {
8         if (big < tab[i])
9         {
10            big = tab[i];
11        }
12    }
13    return big;
14 }
15
16 int main()
17 {
18     int v[] = {12, 1, 70, -4, 122, 223, 1, 2, 1000};
19
20     printf("%d %d\n", inspect(v, 4), inspect(v + 4, 5));
21 }
```

Exo2

Qu'affiche ce code ? Réfléchissez-y sans le faire tourner dans un premier temps.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void process(char* string)
5 {
6     char* gnirts = string + strlen(string) - 1;
7     while (gnirts - string > 0)
8     {
9         char c = string[0];
10        string[0] = gnirts[0];
11        gnirts[0] = c;
12
13        string++;
14        gnirts--;
15    }
16 }
17
18 int main()
19 {
20     char texte[] = "super star";
21     process(texte);
22     printf("%s\n", texte);
23 }
```

Exo3 Parenthèses simples

Écrivez un programme qui lit depuis l'entrée standard une suite de parenthèses et qui décide si c'est une suite "bien formée". La suite contient uniquement les caractères '(' et ')'. Dans une suite bien formée chaque parenthèse ouverte a une parenthèse fermée qui lui correspond et la suite de parenthèses comprise entre ces deux parenthèses est elle-même bien formée. La plus simple suite bien formée est la suite vide. Par exemple, les suites suivantes sont bien formées

()
()
()
()
()

mais ces autres suites ne le sont pas :

)
)
)()
)()
)()

Indice En parcourant la suite depuis le début, à n'importe quel point le nombre de parenthèses ouvertes doit être toujours supérieur ou égal au nombre de parenthèses fermées, et à la fin de la suite les deux nombres doivent être égaux - on a autant de parenthèses ouvertes que fermées.

Exo4 Parenthèses complexes

Nous aimerions refaire l'exercice précédent, mais avec plusieurs types de parenthèses : les accolades et les crochets sont désormais aussi autorisées. Les caractères de la suite seront parmi les six suivants : '{', '}', '[', ']', '(', et ')'.
Voici des exemples supplémentaires de suites bien formées :

[{}(){}]
[{}{}]
{}[{}]

et de suites mal-formées :

[]
[]{}
[)]
[{}]

Attention, la solution simple de l'exercice précédent risque de ne pas marcher, car pour la suite "[{}]" il y a toujours plus de parenthèses ouvertes que fermées du même type, mais pourtant entre les crochets on trouve uniquement une accolade ouverte, ce qui ne représente pas une suite bien formée...

Indice Une solution élégante à ce problème utilise une pile. En parcourant la suite, on **push** chaque parenthèse ouverte sur la pile, et pour chaque parenthèse fermée on effectue un **pop**. Si l'opération **pop** ne fournit pas une parenthèse ouverte du même type que la parenthèse fermée qui l'a déclenchée, c'est que la suite n'est pas bien formée... A la fin la pile doit être vide.

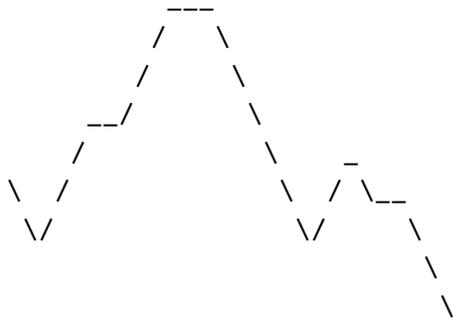
Exo5 (*) Ain't no mountain high enough

Bob est parti faire un tour à vélo. Tous les 100 mètres sa montre connectée enregistre le caractère / s'il se trouve dans une montée, le caractère \ si c'est une descente, et le caractère _ si la route est plate. On aimerait afficher sur l'écran le profil du parcours de Bob. Souvenez-vous qu'en C le caractère \ s'écrit '\\'.

Par exemple, pour la suite

```
\\///_//_//_\\\\///_\\_\\
```

on aimerait afficher :



Bonus : pouvez-vous aussi colorier la montagne ?

