Polytech Nice Sophia
SI3 / 2021-2022
Erick Gallesio

Programmation Procédurale

Feuille 1

Petits exercices en C pour commencer

> *Les exercices de cette feuille sont simples. Leur but est de vous familiariser avec le langage C et le compilateur. La plupart de ces exercices prennent leurs données sur le fichier standard d'entrée et placent leurs résultats sur le fichier standard de sortie. Bien sûr, vous pouvez utiliser les redirections Unix si vous voulez travailler sur des vrais fichiers.*

1 Valeur absolue

Écrire un programme qui saisit un entier et qui affiche sa valeur absolue.

2 Saisie d'entiers positifs

Écrire un programme qui saisit des entiers positifs et qui s'arrête lorsqu'un entier négatif est saisi. Il affiche alors le nombre d'entiers positifs qui ont été saisis.

```
$ ./entiers
Entrer un entier: 12
Entrer un entier: 8
Entrer un entier: 1
Entrer un entier: -1
Nombre d'entiers saisis: 3
```

3 Maximum et somme d'une suite d'entiers positifs

Modifier le programme de l'exercice précédent pour qu'il affiche le plus grand entier qui a été saisi et la somme des tous les entiers saisis.

```
$ ./comptage
Entrer un entier: 12
Entrer un entier: 8
Entrer un entier: 1
Entrer un entier: -1
Le maximum des 3 nombres saisis est 12. La somme est 21.
```

4 Commande cat-num

Ecrire le programme `cat-num` qui affiche en les numérotant les lignes lues sur le fichier standard d'entrée.

```
$ ./cat-num < essai
    1 Un fichier qui contient
    2 deux lignes
$
```

5 Conversions

1. Ecrire un programme qui affiche une table de conversion degré Celsius → degré Fahrenheit sachant que ces deux graduations sont liées par la formule:

$$F = \frac{9 \times C}{5} + 32$$

   Votre programme devra afficher une table de conversion pour les temperatures comprises entre 0°C et 20°C, par pas de 0.5°C.

   Consultez la documentation de `printf` pour afficher une table de conversion **bien alignée**.

```
$ ./conv-v1
 +-------+-------+
 |  0.0C | 32.0F |
 |  0.5C | 32.9F |
 |  1.0C | 33.8F |
 |  1.5C | 34.7F |
 |  2.0C | 35.6F |
 ...
```

2. Modifier votre programme pour que les valeur converties soit toujours arrondies à des valeurs entieres. Vous utiliserez pour cela:

   - un cast
   - la fonction `rint`

6 Comptage

Ecrire un programme qui affiche le nombre d'occurences pour chacun des chiffres et des lettres qu'il a lu sur le fichier standard d'entrée.

```
$ echo "abc 123 cba xy 42" | ./comptage
1:  1 fois
2:  2 fois
3:  1 fois
4:  1 fois
a:  2 fois
b:  2 fois
c:  2 fois
x:  1 fois
y:  1 fois
```

7 Une version simplifiée de la commande *wc*

Ecrire un programme qui affiche le nombre de caractères, lignes et mots lus sur le fichier standard d'entrée. On considère ici, comme dans la commande standard `wc` , qu'un mot est une suite de caractères délimitée par les caractères SPACE, TAB, ou NEWLINE.

```
$ cat in_wc.txt
7       mots
2    lignes et 45 caractères
$ ./wc < in_wc.txt
lines: 2
words: 7
chars: 45
```

Fun fact

Une écriture possible pour ce programme pourrait tenir en une seule ligne:

```
e,n,j,o,y;main(){for(++o;n=-~getchar();e+=11==n,y++)o=n>0xe^012>n&&'``'^n^65?!n:!o?++j:o;printf("%8d%8d%8d\n",e^n,j+=!o&&y,y);}
```

Cette version, qui utilise quelques particularités du C originel, est bien compilable par un compilateur C standard actuel! Elle a été récompensée en 2019 au 26e **International Obfuscated C Code Contest** dans la catégorie des *one-liners* (les programmes qui ne tiennent que sur une seule ligne). La page Wikipedia précédente contient d'autres contributions «intéressantes» qui ont été récompensées à ce concours.

Comme on peut le voir, le langage C permet de faire des programmes qui sont assez illisibles. Pour autant, il est très **important que vous essayiez d'écrire des programmes lisibles et facilement compréhensibles** (sauf si vous participez à un concours de programmes du style de l'IOCCC bien sûr 😉).