

# CS-119(a) – ICC-C Série 2

2024-02-27

## Exo1 Décomposition en base 10

L'utilisateur rentre un nombre entier à 3 chiffres. Lisez-le avec la fonction `scanf` et donnez sa décomposition en base 10. Par exemple, si on rentre 142, on devrait voir la sortie :

centaines: 1, dizaines: 4, unités: 2

## Exo2 Moyenne

Bob aimerait calculer la moyenne d'un vecteur de 5 nombres. Il utilise ce code :

```
1 int vec[] = {1, 2, 3, 4, 5};
2 printf("La moyenne est %d\n",
3     1 / 5 * (vec[0] + vec[1] + vec[2] + vec[3] + vec[4]));
```

Pourtant, le code ne semble pas bien fonctionner. Pourquoi? Qu'affiche-t-il? Il essaye une autre version :

```
1 int vec[] = {1, 2, 3, 4, 5};
2 printf("La moyenne est %d\n",
3     vec[0]/5 + vec[1]/5 + vec[2]/5 + vec[3]/5 + vec[4]/5);
```

Le résultat a changé, mais ce n'est toujours pas ça. Pourquoi? Qu'affiche-t-il? "Cette fois c'est la bonne!" se dit Bob. Et il écrit :

```
1 int vec[] = {1, 2, 3, 4, 5};
2 int somme = vec[0] + vec[1] + vec[2] + vec[3] + vec[4];
3 double moyenne = somme / 5;
4 printf("La moyenne est %g\n", moyenne);
```

Ceci affiche effectivement la bonne moyenne. Est-ce que cette dernière version est juste pour autant?

Donnez un meilleur code pour calculer une moyenne. Lisez les 5 nombres depuis `stdin` avec `scanf`.

### Exo3 Affichage LCD

On aimerait écrire des nombres composés de quatre chiffres en affichage à segments. Affichez 2, 0, 2, et 4 de cette manière. On devrait voir apparaître :

```
  _  _  _
-|  |  |  -|  |
|-  |-  |-   |
```

Suggestion : définissez des constantes de type tableau de string `char[4][4]` qui contiennent les formes des chiffres et ensuite combinez-les pour écrire des chiffres en affichage à segments :

```
1 char four[4][4] = {
2     "  ",
3     "|_|",
4     "  |",
5     "  "
6 };
7
```

### Exo4 Affichage LCD (suite)

Lisez un chiffre depuis l'entrée standard avec `scanf` et écrivez-le en affichage à segments.

Suggestion : définissez un vecteur de constantes de type tableau de tableaux de string qui correspondent aux chiffres de 0 à 9. Utilisez le chiffre que vous lisez depuis l'entrée standard comme indice du tableau ainsi défini.

```
1 char lcd[10][4][4] = ...;
```

### Exo5 Horologe LCD

On aimerait écrire l'heure et les minutes en affichage à segments. Par exemple :

```
  _  _  _
|  |  .  -|  |
|  -|  .  |-  |-
```

Le code pour obtenir l'heure et les minutes de l'horologe de la machine est donné ci-dessous :

```
1 #include <time.h>
2
3 int main() {
4     ...
5     time_t t = time(NULL);
6     int heure = localtime(&t)->tm_hour;
7     int minute = localtime(&t)->tm_min;
8     ...
9 }
```