

# CS-119(a) – ICC-C Série 1

2024-02-20

## Exo1 Echanger deux variables

Comment fait-on pour échanger le contenu de deux variables ? Si on a

```
1 int a = 10, b = 30;
```

quelle(s) instruction(s) doit-on écrire pour que la variable `a` ait la valeur de `b` et la variable `b` ait la valeur de `a`, peu importe leurs valeurs initiales ?

Indice : Si on écrit juste `a=b` on perd la valeur de `a`, puisque `a` prendra la valeur de `b` avant qu'on ait pu sauvegarder sa valeur.

### Solution de l'exercice 1

```
1 int a = 10, b = 30;
2 int c = b;
3 b = a;
4 a = c;
```

## Exo2 Shift-right

Etant donné un vecteur de 5 éléments

```
1 int vec[5] = {1, 2, 3, 4, 5};
```

comment doit-on procéder afin de déplacer toutes les valeurs vers la droite d'une position, et de bouger la dernière valeur en première position ? Après l'exécution du code, le vecteur devrait contenir les valeurs `{5, 1, 2, 3, 4}`.

Affichez le contenu du vecteur à l'aide de la fonction `printf` pour vous assurer que le code fait effectivement ce que vous souhaitez.

### Solution de l'exercice 2

```
1 int vec[5] = {1, 2, 3, 4, 5};
2 int last = vec[4];
3 vec[4] = vec[3];
4 vec[3] = vec[2];
```

```

5  vec[2] = vec[1];
6  vec[1] = vec[0];
7  vec[0] = last;
8
9  printf("vec = [%d, %d, %d, %d, %d]\n",
10    vec[0], vec[1], vec[2], vec[3], vec[4]);

```

## Exo3 Préfixes

Etant donné une chaîne de caractères

```

1  char chaine[] = "string";
2

```

on aimerait afficher tous ses préfixes, sans créer d'autres strings. La sortie doit contenir "s", "st", "str", "stri", et "strin" dans n'importe quel ordre. Vous pouvez par contre modifier la variable `chaine`.

### Solution de l'exercice 3

```

1  char chaine[] = "string";
2  chaine[5] = '\0';
3  print("%s\n", chaine);
4  chaine[4] = '\0';
5  print("%s\n", chaine);
6  chaine[3] = '\0';
7  print("%s\n", chaine);
8  chaine[2] = '\0';
9  print("%s\n", chaine);
10 chaine[1] = '\0';
11 print("%s\n", chaine);

```

## Exo4 ASCII art

En utilisant la fonction `printf` dessinez des formes géométriques. Par exemple, un rectangle de 10x10 caractères en utilisant `|` et `-`, et un losange de côté 5 en utilisant les caractères `/` et `\`.

Utilisez le caractère retour à la ligne '`\n`'. Les caractères spéciaux `\` et `%` doivent être répétés pour être affichés correctement, i.e., "`\\ %%`".

## Solution de l'exercice 4

```
1 #include <stdio.h>
2
3 int main() {
4     // Draw some shapes
5     printf(
6         "Rectangle & losange:\n"
7         "*-----*\n"
8         " |         |     /\\\"    \n"
9         " |         |     /  \\"    \n"
10        " |         |     /       \\"  \n"
11        " |         |     /         \\" \n"
12        " |         |     /           \\" \n"
13        " |         |     \\\"         / \n"
14        " |         |     \\\"         / \n"
15        " |         |     \\\"       / \n"
16        " |         |     \\\"     / \n"
17        " |         |     \\\"/\n"
18        "*-----* \n"
19    );
20 }
```

Pour des dessins plus complexes en ASCII art : <https://www.asciiart.eu/>

```
      / ,  
     /\  \| /  /\  
    |\\_-=._//|  
   \."   "./  
  .-".-.  
 /  - -\ \  //^\\ / ^\\\  
 \  @ @/ . ' `` , / | 0| | 0| \, " `` ' .  
 (_ <_ ) / , ` ' \. --- ./ ' ` , \ \  
 _ ) ( ^ / ^ \ , . " ( ) " . , / ^ \ ` \ \  
 , - ( ^ _ ) ) \ / ^ ( ' . ' - ' . ' ) ^ \ \  
 " - \ ) _ _ / / " ^ " . _ : _ . " ^ " \ \  
 _ _ | | _ _ ` / . ' `` = . , _ `` = `` _ , . = " `` ' . \ `  
jgs (( _ _ | _ _ )) jgs ) (
```