

Programmation C

TP 7 - Les tableaux

I - Objectifs

Nous allons créer un petit jeu de type **mastermind**. Pour ceux qui ne connaissent pas, ce jeu de société se joue à deux, une des personne crée un code secret composé de 4 couleurs. Puis le 2em joueur doit découvrir cette combinaison. Vous pouvez retrouver les règles du jeu ici: <https://www.regles-de-jeux.com/regle-du-mastermind/>

Nous allons simplifier un peu les règles pour ce TP.

- Nous allons utiliser les initiales des couleurs pour lire la saisie utilisateur et stocker le code secret parmis ('R','V','B','J','O')
- Le **code** sera constitué de **4 couleurs** .
- Le code ne sera pas choisi par un joueur mais écrit en dure dans le programme (ex {'R','V','B','J'})
- Le nombre maximal de tentatives via une constante (ex: 3)

...

Donnez un code de 4 couleurs differentes et sans espaces parmis : {'R', 'V', 'B', 'J', 'O'}

> RVJO

Tentative 1/3

Couleurs mal placees: 1

Couleurs bien placees: 2

II - Préparatifs

- 1- Créer un nouveau projet C du nom de **"tp7_tableaux"**.
- 2- Remplacer le texte "hello world" par "TP7: Les Tableaux".

III - Instructions

- 1- Créer les variables et constantes nécessaires au programme (code, saisie_utilisateur, ...). Créer un tableau de char de taille 4 qui contient le code à trouver en utilisant les initiales des couleurs suivantes ('R', 'V', 'B', 'J', 'O').
- 2- Créer la boucle de jeu
 - 2.1- Demander au joueur une combinaison de 4 caractères à la suite sans espace (attention, utiliser la commande "fflush(stdin);" avant le scanf pour ne pas risquer de lire des restes entrés au clavier).
 - 2.2- Comparer les deux combinaisons et calculer le nombre de couleur bien placées.
- 3- Afficher le nombre de tentatives réalisés sur le nombre maximum de tentatives ainsi que le nombre de couleurs bien placées.
- 4- Gérer la fin de partie en cas de victoire (toutes les couleurs bien placées) ou cas de défaite (nombre de tentatives dépassés)

IV - Pour aller plus loin

- A- Ajouter le calcul et l'affichage du nombre de couleur présentes mais mal placées)
- B- Afficher à l'utilisateur la liste des couleurs qu'il peut utiliser.
- C- Remplacer le code secret en dure par un code généré aléatoirement.

```
Donnez un code de 4 couleurs differentes et sans espaces permis  
: {'R', 'V', 'B', 'J', 'O'}  
> RVJO
```

```
Tentative 1/3  
Couleurs mal placees: 1  
Couleurs bien placees: 2
```

V - SPOIL! (aide)

1- Créer des tableaux de char avec 4 cases pour le code et la saisie utilisateur

2- La boucle peut se faire sur le nombre de tentatives

2.1- attention à bien passer en argument l'adresse de la case du tableau soit "&tab[i]" soit via les pointeurs "(tab+i)"

2.2- Il faut faire une boucle qui permet de parcourir à la fois le tableau du code et celui de la saisie utilisateur.

4- Un simple couple de conditions + un break en cas de victoire pour sortir de la boucle de jeu sans faire l'intégralité des tentatives.

A- il y a plusieurs moyens de le faire. Je vous conseil de faire deux boucles imbriquées. la première qui vas parcourir l'ensemble du tableau du code. pour chacunes de ses cases, faire une autre boucle pour parcourir l'ensemble du tableau de saisie utilisateur et ainsi trouver si la couleur est présente et/ou au bon emplacement.

B- créer un tableau statique de la liste des couleur puis l'afficher via une boucle

C- générer un nombre aléatoire qui vous donnera l'indice de la case du tableau des couleurs possibles:

```
srand(time(NULL));
```

```
id_couleur_rand = rand() % taille_liste;
```

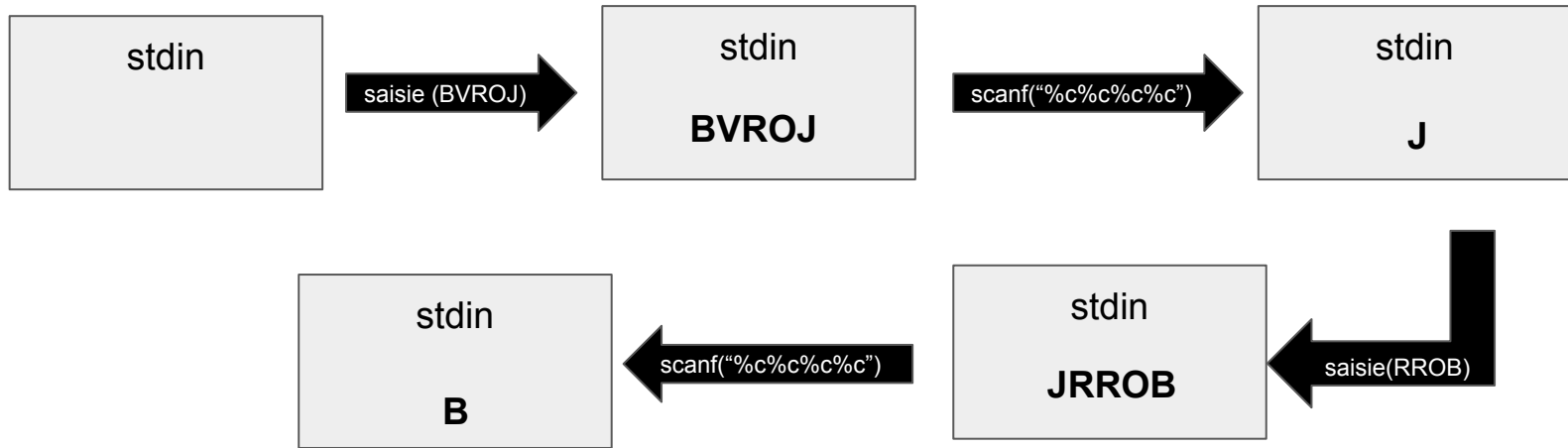
VI - ANNEXE



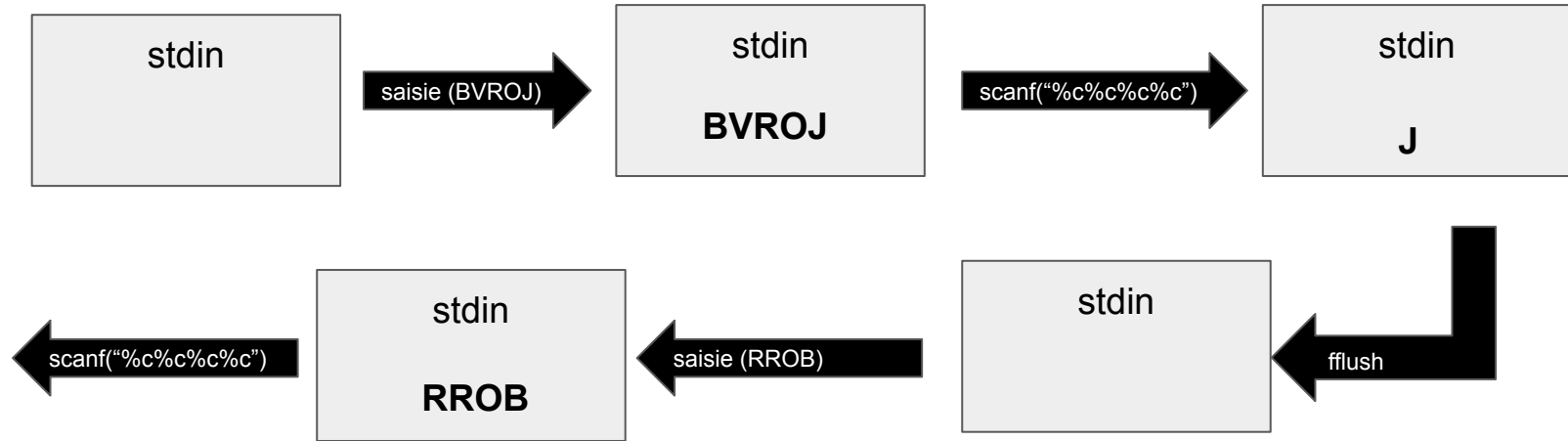
FFLUSH



FFLUSH



FFLUSH



mastermind

code_secret	R	V	B	J
-------------	---	---	---	---

saisie_utilisateur	R	V	J	B
--------------------	---	---	---	---