Travaux Pratiques 2
L2.1
Itérations, aiguillage,fonctions

Nous allons utiliser les différentes formes d'itérations. Nous verrons ensuite comment créer un menu.

---

1. **Affichage de caractères et de leur code ASCII**
   Ecrire un programme qui affiche les caractères de 'A' jusqu'à 'Z' et de '0' jusqu'à '9' suivis la valeur de leur code ASCII.
   Exemple : `caractere A code 65`

2. **Recherche d'un nombre caché**
   Ecrire un programme qui demande à l'utilisateur de trouver un nombre entier secret compris entre 0 et 1000. Pour cela il est guidé par le programme qui lui indique à chaque essai si la valeur saisie est supérieure ou inférieure au nombre secret. L'utilisateur doit trouver le nombre secret en au plus dix essais.
   Pour le moment, vous entrerez vous même la valeur du nombre secret, vous verrez plus tard comment obtenir un nombre aléatoire.
   Lorsque le jeu se termine, le programme affiche le le nombre d'essais `gagne en ... essais` ou `perdu`.

3. **Lecture de caractères**
   Écrire un programme qui lit 2 caractères avec 2 appels à `scanf` puis les affiche sur 2 lignes. (un caractère par ligne entre deux '*').
   Le spécificateur de format "%c" indique à `scanf` qu'un caractère doit être lu .
   Avec ce format les séparateur sont pris en compte. Testez le programme précédent en entrant les caractères de différentes mamières: un par ligne,deux sur la même ligne , des espaces...
   Si l'on veut que `scanf` ignore les séparateur, il faut en ajouter un `espace` dans le format. Remplacer `scanf("%c")` par `scanf(" %c")` et refaites les tests.

4. **Menu**

   - Ecrire un programme proposant une série de choix alphabétiques à l'utilisateur. Le programme affiche uniquement le choix de l'utilisateur, un message d'erreur est affiché si le choix n'appartient pas à l'ensemble des choix possibles.
   - Ajouter un choix de sortie et réafficher le menu (utiliser une fonction d'affichage) tant que le choix de sortie n'a pas été sélectionné.

   !

5. **Passage par adresse**
   Le but de l'exercice est de visualiser le fonctionnement du passage par adresse. Lire un entier dans la fonction `main`, l'afficher afficher son adresse. Appeler une fonction `void ajouteDix(int *n)` qui augmente `*n` de 10. Dans les fonctions `main` et `ajouteDix` afficher l'adresse et la valeur des variables avant et apres l'appel. Faites le même test avec une fonction d'echange.

6. **Papier, caillou, ciseaux**

On veut écrire un programme permettant de jouer au jeu "papier, caillou, ciseau". On codera caillou par 0, papier par 1 et ciseaux par 2. Chaque joueur propose son pari en tapant 0, 1 ou 2. Pour déterminer lequel des deux joueurs a gagné, on utilise l'algorithme suivant :

Soit $J1$ le pari du joueur 1 et $J2$ celui du joueur 2.

- Si $J1$ et $J2$ sont identiques le match est nul,
- $J1$ gagne si $J1 = (J2 + 1) \bmod 3$,
- $J2$ gagne dans le cas restant.

Pour ceci :

- Ecrire une fonction `LireInf2` qui effectue la saisie contrôlée d'un entier entre 0 et 2.
- Ecrire une fonction `arbitre` qui recoit les paris des deux joueurs et qui renvoie 0 si le match et nul, 1 si le joueur 1 a gagné et 2 si c'est le joueur 2.
- Ecrire un programme qui arbitre 10 parties et qui affiche le score.