TD 3
Deug S2
Fonctions, transmission des paramètres

Dans cette séance de travaux dirigés, nous abordons l'utilisation des fonctions. Nous allons voir comment un problème peut être résolu en le décomposant en plusieurs sous-problèmes plus simples.
Une partie des exercices de cette feuille concerne les nombres premiers.

Fonctions ne renvoyant pas de valeur

1. **Table de multiplication**

   - Ecrire une fonction `Multiples` prenant en argument un entier `n` et qui affiche sur une ligne les 10 premiers multiples de `n`. Par exemple, `Multiples(2)` affichera :

     0 2 4 6 8 10 12 14 16 18

   - En déduire un programme `Table` qui affiche sur des lignes consécutives les 10 premiers multiples de `p` pour `p` variant de 1 à 10.

2. **Test de primalité**
   Un nombre est dit *premier* si ses seuls diviseurs sont 1 et lui-même. Par convention, 1 n'est pas premier.

   - Décrire la suite d'instructions permettant de déterminer si un nombre est premier (algorithme).
   - En déduire une fonction `AffichePremier` qui prend en argument un entier `n` et qui affiche un message indiquant si l'entier `n` est premier ou non.

Fonctions renvoyant une valeur

1. Ecrire une fonction `Deuxpremiers` qui reçoit deux entiers et affiche les messages :
   `Aucun des nombres n'est premier`
   `Un seul nombre est premier`
   `Les deux nombres sont premiers`
   On pourra transformer la fonction `AffichePremier` en une fonction `EstPremier` qui renvoie la valeur 1 si le nombre est premier et l'entier 0 sinon. Ce choix permet d'utiliser la valeur de retour par exemple dans un test.

2. **Nombre premier supérieur à un entier n**
   Ecrire une fonction `PremierSuperieur` prenant en argument un entier `n` et qui a pour valeur le premier nombre premier supérieur ou égal à `n`.

3. **Liste des nombres premiers**
   Ecrire une fonction `PremiersPremiers` prenant en argument un entier `n` et qui affiche les nombres premiers inférieurs à `n`.

4. **Nombres premiers jumeaux**
   Par définition, deux nombres premiers sont *jumeaux* si et seulement si leur différence est égale à 2. Par exemple, 11 et 13 sont jumeaux. Ecrire une fonction `Jumeaux` prenant en argument un entier `n` positif qui affiche les couples de nombres premiers jumeaux inférieurs à `n` .

5. **Décomposition en facteurs premiers**
   On cherche à afficher la décomposition en facteurs premiers d'un entier `n`.

   - Trouver un algorithme qui détermine la décomposition en facteurs premiers d'un entier `n`.
   - En déduire une fonction `Decomposition` qui affiche la décomposition d'un entier en facteurs premiers.
     On pourra bien entendu utiliser les fonctions écrites précédemment.
     Par exemple pour 8712, on aimerait avoir l'affichage :

     $$2\hat{\ }3 * 3\hat{\ }2 * 11\hat{\ }2$$

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 à 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.