

Exercice 1

1. Écrire une fonction `premier` d'un paramètre entier `m` et qui renvoie 1 si `m` est premier, et 0 sinon.
2. Écrire une fonction `prochain-premier` prenant un paramètre entier `n` et qui renvoie le plus petit nombre premier plus grand ou égal à `n`.
3. Écrire un programme qui demande un entier `n` à l'utilisateur et affiche le premier nombre premier plus grand ou égal à `n`.

Exercice 2 :

Écrire un programme qui demande à l'utilisateur de saisir un entier tant que l'entier donné est strictement positif. Après la fin de la saisie (l'utilisateur a entré 0), le programme affiche la somme des nombres saisis, le minimum et le maximum.

Exercice 3 :

Écrire un programme qui permet de jouer au jeu du pendu. Le joueur 1 entre le mot, l'ordinateur l'enregistre et affiche une série de lignes blanches pour que le joueur 2 ne puisse pas lire le mot, il affiche ensuite une série d'étoiles qui correspond au nombre de lettres du mot à trouver. Le joueur 2 propose des caractères jusqu'à ce qu'il ait trouvé le mot, ou qu'il ait perdu (nombre d'essais > 10). À chaque fois que le joueur 2 propose un caractère l'ordinateur affiche le mot avec des * et les caractères déjà trouvés.

On utilisera l'appel système "`system("clear");`" pour effacer l'écran (afin que le second joueur ne voit pas les lettres entrées au clavier par le premier joueur). Les bibliothèques suivantes sont utilisées : `stdio.h`, `string.h` (pour les fonctions de manipulation des chaînes de caractères) et `stdlib.h`. La fonction `strlen` donne le nombre de caractères d'une chaîne de caractères (sa longueur). L'utilisateur entre une chaîne de caractères `mot` au clavier grâce à l'instruction `scanf("%s", mot)`. Rappelons également qu'une chaîne de caractères est toujours terminée par :

`'\0'`

Enfin, la fonction `strchr` pourra être utilisée : `strchr(mot, '*')` renvoie l'adresse de la première occurrence du caractère `'*'` dans la chaîne `mot` (et renvoie `NULL` s'il `mot` ne contient pas le caractère `'*'`).