

Colle 03

Cette troisième colle vous fera reconnaître et compléter du code OCaml, et écrire quelques fonctions simples en OCaml.

Fonctions arithmétiques

- Écrivez mathématiquement une fonction `moyenne_trois` qui prend trois flottants x, y, z (on verra plus tard les listes et tableaux) et renvoie leur moyenne comme un autre flottant ;
- Donnez à l'écrit ses signatures sous formes curryfiée et décurryfiée ;
- Ecrivez en OCaml les *deux* variantes (`let moyenne_trois x y z = ...` pour la curryfiée et `let moyenne_trois (x, y, z) = ...` pour la décurryfiée).

Figures en mode texte

Écrivez une fonction qui affiche une figure comme celle-ci, pour $n \geq 6$ la hauteur et la largeur donnée en argument de la fonction. On rappelle qu'en OCaml, `print_string s` affiche la chaîne `s` (écrite entre guillemet ") sans sauter à la ligne, et `print_endline s` affiche `s` avec un saut de ligne en fin de ligne. La boucle `for` s'écrit `for x = X_inclus to Y_inclus do ... done`.

```
*   *
*   *
*   *
*   *
*   *
*****
```

QCM

Le(s)quel(s) des codes suivants sera une fonction OCaml valide ? (et il faudra pouvoir justifier)
Corrigez les morceaux qui ne vont pas pour les lignes incorrectes.

- `let minimum x y = if x < y then x else y;;`
- `let minimum = fun x -> fun y -> if x < y then x else y;;`
- `def est_plus_petit (x, y) = if x < y then True else False;;`
- `let est_plus_petit x y = if x < y then true else false;;`

Pseudo-code à compléter

Corrigez le pseudo-code suivant en remplissant les trous `XXX`, pour obtenir une fonction qui teste si un entier `n` donné en argument est un nombre premier ou non.

```
Fonction estPremier(n : entier) -> booléen =
  maxDiviseur = XXX
  Pour chaque diviseur = XXX à XXX (inclus) faire
    Si n XXX diviseur XXX XXX Alors Renvoyer XXX
  Fin Pour
  Renvoyer XXX
Fin Fonction
```