

Algorithmique PEIP2

TD1 : Boucles et conditionnelles

Tous les algorithmes dans ce TD sont à écrire en pseudo-code, selon la structure vue en cours. Un exemple est donné à l'Exercice 1. Pour chacun de ces algorithmes, il faut être capable de justifier qu'il termine, et que la valeur renvoyée est correcte.

1 Analyse d'algorithme

On considère l'algorithme suivant.

Algorithme 1 : algo

Entrées : $n \geq 1$: entier

Sorties : r : booléen

$i \leftarrow 2$

$r \leftarrow \text{Vrai}$

tant que $i < n$ **et** r **faire**

si $n \bmod i = 0$ **alors**

$r \leftarrow \text{Faux}$

fin

$i \leftarrow i + 1$

fin

retourner r

1. Que fait cet algorithme ?
2. Toutes les itérations de boucle sont-elles nécessaires ? Comment l'améliorer ?

2 Conditionnelles

Utiliser la structure Si-Alors-Sinon pour répondre aux questions suivantes.

1. Écrire un programme `bissextile` qui prend en paramètre un entier a représentant une année et qui renvoie `Vrai` si cela correspond à une année bissextile, et `Faux` sinon.

Pour rappel, une année est bissextile si elle est multiple de 400, ou bien un multiple de 4 et non-multiple de 100. Par exemple, 2012 est bissextile mais 2014 ne l'est pas ; 2000 est bissextile, mais 1900 ne l'est pas.

2. Écrire un algorithme `jours` qui prend en paramètre un entier m représentant un mois (1 pour janvier, 12 pour décembre), et un entier a représentant une année. Cet algorithme devra calculer le nombre de jours de ce mois. À titre de rappel, les mois de janvier, mars, mai, juillet, août, octobre et décembre ont 31 jours, les mois d'avril, juin, septembre et novembre ont 30 jours, et le mois de février a 28 jours sauf pendant les années bissextiles où il a 29 jours.

On essaiera d'exploiter les régularités du nombre de jours des mois afin d'avoir les conditions les plus concises possibles.

3 Enumérations

Dans ces exercices, on s'intéresse à l'énumération de nombres aux propriétés particulières. L'objectif est de parvenir au résultat avec le moins d'itérations de boucle possible.

1. Écrire un algorithme prenant en paramètre un entier n , et qui affiche tous les entiers pairs inférieurs à n .
2. Écrire un algorithme prenant en paramètre un entier n , et qui affiche tous les carrés parfaits inférieurs à n .

4 Boucles

1. Écrire un algorithme `log` prenant en paramètre deux nombres $b, x > 1$, et qui renvoie la partie entière de $\log_b(x)$.
Il s'agit du plus grand entier q tel que $b^q \leq x$. Cela peut se calculer en n'utilisant que des multiplications et une boucle `Tant que`.
2. On suppose que l'on dispose de la fonction `EstPremier(n)` qui détermine si l'entier n est premier ou non. Écrire un algorithme `NonPremiersConsécutifs` qui prend en paramètre un entier k , et qui renvoie le plus petit entier n tel que tous les entiers compris dans l'intervalle $[n + 1; n + k]$ ne sont pas premiers.
3. Écrire un algorithme `EntierMystère` qui prend en paramètre un entier mystère x à faire deviner à un utilisateur. Cet algorithme demande de façon répétée à l'utilisateur de rentrer une valeur jusqu'à ce qu'il ait deviné la valeur x auquel cas il le félicite et arrête le jeu, et en précisant si la valeur à deviner est supérieure ou inférieure à celle proposée en cas d'échec.

5 Tirage au sort

On dispose de la fonction `Random()` qui renvoie un nombre choisi aléatoirement et uniformément dans l'intervalle $[0, 1[$.

1. Écrire un algorithme `pièce` qui prend en paramètre un nombre $p \in [0, 1]$, et qui renvoie `Pile` avec probabilité p , et `Face` avec probabilité $1 - p$.
2. Écrire un algorithme `dé6()` qui simule un lancer de dé à 6 faces, et renvoie donc un entier aléatoire uniforme compris entre 1 et 6.
3. On cherche à simuler l'expérience suivante. On dispose d'un dé à 6 faces, et d'un dé à 8 faces. On fait une séquence de lancers, en utilisant d'abord le dé à 6 faces, et en changeant de dé après chaque résultat pair. Écrire un algorithme `lancers` qui prend en paramètre un entier n , et qui simule n lancers de dés selon l'expérience décrite ci-dessus.

6 Impôt sur le revenu

On cherche à écrire un algorithme `impôt` qui calcule l'impôt sur le revenu d'un ménage, en prenant en paramètre la valeur r du revenu imposable, et le nombre p de parts fiscales du foyer.

- Le revenu net imposable est obtenu en diminuant de 10% le revenu imposable.
- Le quotient familial qf est obtenu en divisant le revenu net imposable par le nombre de parts.
- Ensuite, appliquer le barème 2021 pour calculer l'impôt brut. Seule la partie du quotient familial appartenant à la tranche d'impôt est imposée selon le taux correspondant. Par exemple, avec $qf = 10184 = 10084 + 100$, l'impôt sera $10084 \times 0\% + 100 \times 11\% = 11$.

tranche de l'impôt	taux d'imposition
$[0; 10084]$	0%
$[10084; 25710]$	11%
$[25710; 73516]$	30%
$[73516; 158122]$	41%
$[158122; +\infty[$	45%