

Initiation à l'algorithmique

X2I0040, groupe 211

CC 1

27/03/2017

Durée: 60 Minutes

Nom, Prénom: _____

Préambule : Aucun document autorisé. Calculatrices et téléphones portables interdits. Les exercices ne sont pas classés par difficulté croissante. Nombre de pages : 2

1. Pour vérifier qu'un nombre N est **binaire**, on le divise successivement p fois par 10 (où p est le nombre de chiffres composant N moins un), en vérifiant à chaque division que le reste appartienne à l'ensemble $\{0, 1\}$.

Par exemple, pour vérifier que le nombre $N = 101$ est un nombre **binaire**, on procède comme suit :

```
1 n <- 101
2 r <- n mod 10 = 1
3 r = 1 est dans {0,1} alors... OK !
4 n <- n div 10 = 10
5 r <- n mod 10 = 0
6 r = 0 est dans {0,1} alors... OK !
7 n <- n div 10 = 1
8 r <- n mod 10 = 1
9 r = 1 est dans {0,1} alors... OK !
10 Toutes les restes appartenant à l'ensemble {0,1}, donc :
11     N est BINAIRE
```

Si dans l'une des étapes numéro 3, 6, ou 9, le reste r n'appartient pas à l'ensemble $\{0, 1\}$, alors l'algorithme termine en affichant que le nombre N n'est pas **binaire**.

Question 1 Traduisez l'algorithme précédent en code ALGOSCRIPT.

Question 2 (bonus) Écrivez un algorithme pour convertir n'importe quel nombre entier en **binaire**.

2. Étant donné un tableau T d'entiers, écrire une fonction **occurrences** qui retourne la quantité de fois qu'un élément e est présent dans ce tableau.

Exemple : Si $T = [3, 6, 3, 8, 5, 3, 9, 7, 4, 5, 3]$, l'élément $e = 3$ se trouve **4** fois dans le tableau.

3. On suppose disposer de la fonction :

```

1 fonction etoile(nomb : entier) : chaine
2 Variables
3     etoiles : chaine
4     i : entier
5 Debut
6     etoiles <- ""
7     pour i allant de 1 a nomb faire
8         etoiles <- etoiles + "*";
9     fin pour
10    retourner (etoiles)
11 Fin

```

En utilisant la fonction **etoile**, écrire un algorithme affichant le texte suivant :

```

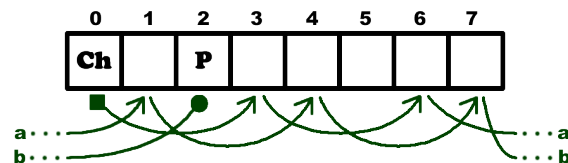
*
* *
* * *
* * * *
* * * * *
* * * * * *
* *

```

4. (**bonus**) Le jeu de *chasseur-proie* en une dimension consiste à faire partir un chasseur d'une position (généralement à la position 0) d'un tableau, et le faire chasser une proie située sur une autre position du tableau. Si le chasseur marche avec une vitesse (quantité de cases qu'il peut parcourir à chaque fois) égal à un, une proie située à la position k sera éliminée en k pas. Maintenant, considérons que ce pas soit une quantité donnée.

Quand le chasseur dépasse la dernière case du tableau, il recommence au début (case 0) et complète son avancement.

Exemple :



Dans cet exemple, avec un pas égal à trois, un chasseur a besoin de six mouvements pour chasser la proie : (3, 6, 1, 4, 7, 2).

On doit aussi analyser le cas où le chasseur ne peut jamais chasser la proie.

Question 1 Écrivez une fonction qui reçoit trois nombres entiers (la taille du tableau, la position de la proie et la vitesse du chasseur) et qui retourne la quantité de mouvements dont le chasseur à besoin pour chasser la proie.