
R1.01 INITIATION AU DÉVELOPPEMENT
FEUILLE DE TD N°1
Introduction aux structures algorithmiques

Exercice 1 *Structures algorithmiques*

Le but de cet exercice est d'introduire et de manipuler les structures algorithmiques : *séquence* et *conditionnelle*. Les algorithmes de cet exercice ne sont pas destinés à devenir des programmes mais permettent de voir les notions sur un cas *de la vie courante*.

Les algorithmes que nous allons décrire sont à destination d'une personne chargée de servir des rafraîchissements à des clients. Cette personne a à sa disposition une machine qui distribue l'équivalent d'un verre de liquide lorsqu'on appuie sur son bouton, il dispose aussi de gobelets. L'algorithme est déclenché lorsqu'un client commande une boisson, son résultat doit être que le client dispose d'un gobelet rempli. On suppose qu'il n'y a pas de gobelet posé sur la machine au moment de la commande et on veut que cela soit aussi le cas une fois le client servi.

Version 0.1 Voici une première version beta de l'algorithme à destination du serveur :

```
Lorsqu'un client commande
  Appuyer sur le bouton
  Poser le verre sous le verreur
  Donner le verre au client
```

Cet algorithme est-il correct ? C'est-à-dire donne-t-il le bon résultat ? Si non pourquoi ? Si oui peut-il être amélioré ?

Version 0.2 La deuxième version de l'algorithme est la suivante

```
Lorsqu'un client commande
  Poser le verre sous le verreur
  Appuyer sur le bouton
  Appuyer sur le bouton
  Donner le verre au client
```

Mêmes questions que pour l'algorithme précédent.

Version 0.3 Proposer une version correcte et optimisée de l'algorithme.

Version 1.0 La machine dispose d'un voyant qui s'allume quand la réserve de boisson est vide. Dans ce cas, il faut que le serveur change la bonbonne pour en mettre une nouvelle. Pour cela on dispose des instructions **Enlever la bonbonne** et **Mettre une nouvelle bonbonne**.

Compléter l'algorithme précédent pour prendre en compte le rechargement de la machine. Vérifier que votre algorithme fonctionne à l'arrivée du premier client, quelque soit l'état de la bonbonne. Peut-on optimiser le temps d'attente du client si on met des hypothèses sur l'état initial de la bonbonne ?

Version 1.1 On vient d'acquérir une nouvelle machine qui dispose de quatre bonbonnes et de quatre boutons avec voyants. Les boutons et les bonbonnes sont numérotés de 1 à 4. Cette machine permet donc de distribuer quatre types de boissons différentes. On va considérer que dans la bonbonne correspondant au bouton 1 il y a de la limonade, pour le bouton 2 il y a du thé glacé, pour le bouton 3 du jus d'orange et pour le bouton 4 un soda. Cette machine n'a qu'un seul emplacement pour le verre (les quatre verseurs aboutissent au même endroit). Par conséquent, maintenant lorsqu'un client commande il indique quelle boisson il désire.

Ecrire un algorithme qui permet au serveur de servir la bonne boisson au client. Bien préciser le ou les paramètres de votre algorithme.

Exercice 2 *Les boucles*

Un scientifique spécialiste de la protection de la galinette cendrée (espèce en voie de disparition pour cause de sur-chasse) a effectué un prélèvement de plusieurs centaines d'œufs de cet oiseau afin d'effectuer une étude d'envergure. La première étape de cette étude consiste à trier les œufs contenant un mâle de ceux contenant une femelle. Pour cela un test très simple consiste à passer l'œuf sous une lampe à ultra-violet. Si l'œuf devient vert c'est qu'il contient un mâle, s'il devient rouge c'est qu'il contient une femelle. Voici les instructions qu'il a données à la personne chargée de ce travail sachant qu'il souhaite obtenir à la fin une boîte contenant les œufs mâles, une boîte contenant les œufs femelles ainsi que le nombre d'œufs de chaque sorte.

```
algo trier_oeufs
paramètre:
    boite_entree: la boîte d'œufs
début:
    allumer la lampe à UV
    prendre une boîte vide boite_f
    écrire FEMELLES sur boite_f
    prendre une boîte vide boite_m
    écrire MALES sur boite_m
    pour chaque oeuf o de boite_entree
        passer o sous la lampe à UV
        si o est vert sous la lampe UV
            mettre o dans boite_m
        sinon
            mettre o dans boite_f
    prendre une ardoise
    écrire nb_males = 0 sur l'ardoise
    prendre une boîte vide boite_m2
    écrire MALES sur boite_m2
    pour chaque oeuf o de boite_m
        mettre o dans boite_m2
        ajouter 1 au nombre associé à nb_males sur l'ardoise
    écrire ce nouveau nombre sur l'ardoise en face de nb_males
    écrire nb_femelles = 0 sur l'ardoise
    prendre une boîte vide boite_f2
    écrire FEMELLES sur boite_f2
    pour chaque oeuf o de boite_f
        mettre o dans boite_f2
        ajouter 1 au nombre associé à nb_femelles sur l'ardoise
    écrire ce nouveau nombre sur l'ardoise en face de nb_femelles
    retourner nb_males,boite_m2,nb_femelles,boite_f2
```

2.1 Selon vous cet algorithme est-il correct ? (donne-t-il le bon résultat ?)

La personne en charge de cette tâche estime qu'elle pourrait aller deux fois plus vite en suivant un autre algorithme.

2.2 Donner un algorithme correct qui permet d'effectuer ce processus plus rapidement.

Pour les besoins de son étude, le scientifique souhaite étiqueter les œufs femelles d'un chiffre compris entre 1 et 4 de telle sorte qu'il y ait autant d'œufs de chaque étiquette à une unité près.

2.3 Proposer un algorithme qui permet d'étiqueter les œufs femelles sans connaître le nombre d'œufs contenu dans cette boîte.

Exercice 3 Conditionnelle

Voici des algorithmes permettant de trouver le minimum de trois nombres :

```
algo min1
paramètres: x,y,z trois nombres
résultat: le plus petit des trois nombres
début
    si x est plus petit que y
        mettre x dans res
    sinon
        mettre y dans res
    si y est plus petit que z
        mettre y dans res
    sinon
        mettre z dans res
    si x est plus petit que z
        mettre x dans res
    sinon
        mettre z dans res
    retourner res
```

```
algo min2
paramètres: x,y,z trois nombres
résultat le plus petit des trois nombres
début
    si x est plus petit que y
        si x est plus petit que z
            mettre x dans res
        sinon
            mettre z dans res
    sinon si y est plus petit que z
        mettre y dans res
    sinon
        mettre z dans res
    retourner res
```

```
algo min3
paramètres: x,y,z trois nombres
résultat le plus petit des trois nombres
début
    si x est plus petit que y
        mettre x dans res
    sinon
        mettre y dans res
    si z est plus petit que res
        mettre z dans res
    retourner res
```

3.1 Donner le résultat des trois algorithmes avec les valeurs suivantes :

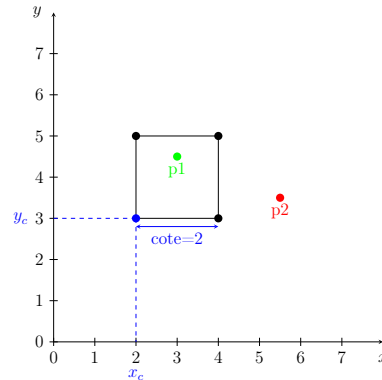
algo	x	y	z	résultat	algo	x	y	z	résultat	algo	x	y	z	résultat
min1	6	10	13		min2	6	10	13		min3	6	10	13	
min1	14	8	17		min2	14	8	17		min3	14	8	17	
min1	25	14	1		min2	25	14	1		min3	25	14	1	
min1	14	25	16		min2	14	25	16		min3	14	25	16	

3.2 Indiquer lequel ou lesquels des algorithmes sont corrects.

Exercice 4 Conditionnelle : géométrie

Le but de cet exercice est d'arriver à détecter si un point est à l'intérieur d'un carré ou non, puis de savoir si deux carrés s'intersectent ou non. On ne va considérer que des carrés dont les cotés sont parallèles au repère.

Pour définir un carré, nous allons donner trois nombres, $x_c, y_c, cote$ qui sont respectivement les coordonnées en x et y de son coin inférieur gauche et la longueur de ses cotés. Sur la figure ci-contre le carré est défini par $x_c = 2$, $y_c = 3$, $cote = 2$. Les points sont eux définis par leurs coordonnées en x et y . Le point $p1$ est défini par $x_p = 3$, $y_p = 4.5$.



4.1 Remplir le tableau suivant en mettant *vrai* si le point est dans le carré et *faux* sinon.

x_c	y_c	$cote$	x_p	y_p	résultat
2	3	2	3	4.5	
2	3	2	5.5	3.5	
2	3	2	4	5	
2	3	2	1	1	

4.2 Tester si un point fait partie d'un carré ou non

- Indiquer quels sont les paramètres de l'algorithme
- Indiquer le type du résultat
- Ecrire l'algorithme qui permet de répondre à la question

4.3 Tester si deux carrés s'intersectent.

- Remplir le tableau suivant

x_{c1}	y_{c1}	$cote_1$	x_{c2}	y_{c2}	$cote_2$	résultat
2	3	2	3	4.5	3	
2	3	2	5.5	3.5	8	
2	3	2	4	5	1	
2	3	2	1	1	4	

- Indiquer quels sont les paramètres de l'algorithme
- Indiquer le type du résultat
- Indiquer en une ou deux phrases comment on peut effectuer ce test
- Ecrire l'algorithme qui permet de répondre à la question