

Nom : _____

Prénom : _____

Identifiant : _____ Groupe : _____

/20



Haute École de Bruxelles-Brabant
École Supérieure d'Informatique
Bachelor en Informatique

23 mai 2019
DEV2

DEV2 – Développement

Algorithmique

Examen de première session

Consignes

1. L'examen dure 3h.
2. On peut considérer que les données lues ou reçues ne comportent pas d'erreurs. Vous ne devez donc pas les tester.
3. Veillez à rendre vos solutions modulaires. Ceci est **très** important.
4. On vous encourage fortement, dans vos brouillons, à identifier les données, les inconnues et à commencer par une résolution sous forme de texte ou de dessins.

Question 1	Question 2	Question 3	Total
/6	/6	/8	/20

1 Tri sélectif

(6 points)

Ecrivez un algorithme qui reçoit en paramètre une liste d'entiers triés par ordre croissant et qui **modifie** cette liste de sorte qu'elle soit triée comme suit : les nombres impairs sont positionnés en début de liste en ordre décroissant alors que les nombres pairs sont positionnés en fin de liste en ordre croissant.

Exemple La liste [1,4,7,8,10,11,13,14,18] devient [13,11,7,1,4,8,10,14,18] après l'application de l'algorithme.

```
algorithme tri(↓↑ : List of Integers)
```



2 Jeu de piste

(6 points)

Soit un tableau deux dimensions dont chaque cellule contient une structure **Case**. Cette structure est composée de deux champs, représentant la ligne et la colonne d'une cellule du tableau. Chaque **Case** fait donc référence à une cellule du tableau, constituant ainsi un parcours des différentes cellules de celui-ci.

```
structure Case
|   lg : Integer
|   col : Integer
fin structure
```

Ecrivez un algorithme qui reçoit un tel tableau ainsi qu'une position de départ. L'algorithme renvoie un booléen indiquant si « le cycle » constitué des différentes coordonnées contient une et une seule fois chaque case du tableau.

Exemple Soit le tableau suivant et la cellule de départ (1,1).

(1,2)	(2,2)	(3,3)	(2,1)
(3,1)	(1,3)	(4,4)	(4,3)
(3,4)	(1,1)	(4,1)	(2,3)
(4,2)	(1,4)	(3,2)	(2,4)

Commencer à la case (1,1) donne le parcours suivant : (1,1) → (1,2) → (2,2) → (1,3) → (3,3) → (4,1) → (4,2) → (1,4) → (2,1) → (3,1) → (3,4) → (2,3) → (4,4) → (2,4) → (4,3) → (3,2) → (1,1).

Toutes les cellules du tableau ont été parcourues une seule fois, l'algorithme retourne donc **true**.

Exemple bis

Soit le tableau suivant et la cellule de départ (3,1).

(3,1)	(2,2)	(3,3)	(2,1)
(3,1)	(1,3)	(4,4)	(4,3)
(3,4)	(1,1)	(4,1)	(2,3)
(4,2)	(1,4)	(3,2)	(2,4)

Commencer à la case (3,1) donne le parcours suivant : (3,1) → (3,4) → (2,3) → (4,4) → (2,4) → (4,3) → (3,2) → (1,1) → (3,1).

Toutes les cellules du tableau ne sont pas contenues dans le cycle, l'algorithme retourne donc **false**.

3 Un seul zéro par ligne (6 points)

Ecrivez un algorithme qui reçoit en paramètre un tableau à deux dimensions et qui retourne vrai si chaque ligne contient exactement un zéro.

```
algorithme onlyOneZero(tab : Array of nxm Integers) → boolean
```

4 Un petit BlackJack ? (8 points)

Le blackjack est un jeu de casino dans lequel le joueur/la joueuse joue contre la banque. Le but du jeu est que la somme de ses cartes soit le plus proche de 21 sans pour autant dépasser cette valeur. Chaque carte vaut sa valeur, sauf les figures (valet, dame et roi) qui valent dix et l'as qui vaut **1 ou 11**.

Nous vous demandons d'écrire un algorithme qui reçoit une liste de **Carte** et qui retourne la somme **la plus avantageuse** pour le joueur. C'est à dire, celle qui est le plus proche de 21 sans pour autant dépasser cette valeur.

Nous mettons à votre disposition la structure **Carte** suivante :

```
structure Carte
    value : Integer
    color : String
fin structure // entier compris entre 1 et 13
```

```
algorithme sommeCarte(cartes : List of Carte)
```

5 Statistique (6 points)

Soit la structure **Etudiant** suivante :

```
structure Etudiant
    nom : String
    option : String
    cours : String
    cote : Integer
fin structure
```

1. Ecrivez un algorithme qui reçoit en paramètre un tableau d'Etudiant trié et affiche pour chaque cours, le nombre d'étudiants inscrits, la meilleure cote obtenue par chaque option ainsi que l'option où la moyenne des cotes est la meilleure. Vous pouvez considérer que le tableau est trié selon le tri que vous proposez à la question suivante. /5
2. Comment le tableau doit-il être trié pour pouvoir résoudre le problème avec un algorithme de rupture ? /1

Exemple L'affichage de l'algorithme ressemblera à :

```
DEV1
Meilleur Cote
    Gestion : 16
    Réseau : 19
    Industriel : 20
Meilleur Option
```

Gestion
Nombre d'étudiants
128

MAT1
Meilleur Cote
Gestion : 11
Réseau : 13
Industriel : 11
Meilleur Option
Industriel
Nombre d'étudiants
254

DEV2
Meilleur Cote
Gestion : 15
Réseau : 14
Industriel : 15
Meilleur Option
Réseau
Nombre d'étudiants
128

Notez bien que ce n'est pas parce qu'une option possède la meilleure note qu'elle a pour autant la meilleure moyenne.