

Nom : \_\_\_\_\_

Prénom : \_\_\_\_\_

Identifiant : \_\_\_\_\_ Groupe : \_\_\_\_\_

/20



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

22 août 2018  
DEV2

## DEV2 – Développement

### Algorithmique

Examen de 2<sup>e</sup> 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	/7	/7	/20

#### 1 Le casino

(6 points)

Dans le cours, nous avons présenté une version simplifiée d'un jeu de roulette dans un casino. En voici les règles.

##### Notre règle du jeu

Dans ce jeu simplifié, vous pourrez miser une certaine somme et gagner ou perdre de l'argent. Quand vous n'avez plus d'argent, vous avez perdu.

- La roulette est constituée de 50 cases numérotées de 0 à 49. Les numéros pairs sont de couleur noire, les numéros impairs sont de couleur rouge. Le jeu se joue à **un seul joueur** (et un croupier, qui manipule la roulette).
- Le joueur mise sur un numéro compris entre 0 et 49. En choisissant son numéro, il dépose la somme qu'il souhaite miser.
- Le croupier lance la roulette, lâche la bille et quand la roulette s'arrête, relève le numéro de la case dans laquelle la bille s'est arrêtée. (Dans notre programme, nous ne reprendrons



pas tous ces détails « matériels » mais ces explications sont aussi à l'intention de ceux qui ont la chance d'avoir évité les salles de casino jusqu'ici. ) Le numéro sur lequel s'est arrêtée la bille est, naturellement, le numéro gagnant.

- Si le numéro gagnant est celui sur lequel le joueur a misé (probabilité de 1/50, plutôt faible), le croupier lui rend sa mise et lui remet, en plus, 3 fois la somme misee.
- Sinon, le croupier regarde si le numéro misé par le joueur est de la même couleur que le numéro gagnant (c'est-à-dire s'ils sont tous les deux pairs, ou tous les deux impairs). Si c'est le cas, le croupier lui rend sa mise et remet, en plus, 50% (arrondi vers le bas) de la somme misee. Si ce n'est pas le cas, le joueur perd sa mise.
- Le jeu recommence tant que le joueur a de l'argent et souhaite continuer.

Nous aimerions en écrire une solution orienté objet respectant MVC. Voici le contrôleur que nous avons obtenu.

```
algorithme jeuRouletteControleur()
    jeu : JeuRoulette
    view : View
    jeu ← nouveau JeuRoulette()
    view ← nouvelle View(jeu)
    jeu.ajouterCrédit(view.demanderCrédit())
    tant que NON jeu.estFini() faire
        mise : entier
        mise ← view.demanderMise()
        si mise=0 alors
            jeu.quitterJeu()
        sinon si jeu.estMiseValide(mise) alors
            case : entier
            case ← view.demanderCase()
            jeu.miser(mise, case)
            jeu.lancerRoulette()
            view.afficherResultat()
        fin si
    fin tant que
fin algorithme
```

Nous vous demandons d'implémenter la partie modèle, c'est-à-dire la classe **JeuRoulette** (attributs, constructeur et méthodes) de ce jeu afin de fournir les méthodes suivantes :

```
méthode ajouterCrédit(duCrédit : entier)
méthode estFini() → booléen
méthode estMiseValide(uneMise : entier) → booléen
méthode miser(mise, case : entiers)
méthode lancerRoulette()
méthode quitterJeu()
// Les 2 méthodes suivantes seront utilisées par la vue
méthode getDernierGain() → entier
méthode getDernierRésultat() → entier // Case où s'est arrêtée la bille
```

Pour toutes ces méthodes, vous pouvez supposer que les paramètres sont corrects. Vous ne devez **pas** non plus vérifier que les méthodes sont appelées dans le bon ordre. Par exemple, on ne vous demande pas de vérifier qu'une mise a bien été placée avant de lancer la roulette.

## 2 Dessin par numéro

(7 points)

Vous avez probablement déjà rencontré des grilles de coloriage par couleur. Les différentes couleurs sont ici répertoriées dans une liste de chaînes. Pour connaître la couleur à utiliser pour colorier une case de la grille, on regarde le numéro qui est écrit dans la case : c'est la position (dans la liste des chaînes) du nom de la couleur à utiliser.

On vous demande de transformer une telle grille en une séquence d'instructions de coloriage en partant systématiquement d'en haut à gauche et en parcourant chaque ligne de gauche à droite. Les instructions possibles sont :

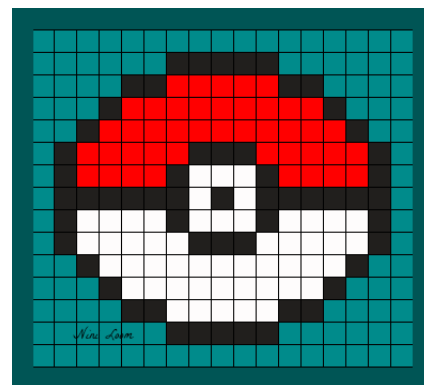
- Se placer en haut à gauche.
- Colorier  $\langle n \rangle$  cases.
- Prendre le crayon de couleur  $\langle \text{couleur} \rangle$ .
- Passer au début de la ligne suivante.

Exemple : Si on donne la liste [ "blanc", "bleu", "noir", "rouge" ] et que la grille est :

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	2	2	2	2	2	1	1	1	1	1	1
2	1	1	1	1	2	2	3	3	3	3	3	2	2	1	1	1	1
3	1	1	1	2	3	3	3	3	3	3	3	3	3	2	1	1	1
4	1	1	2	3	3	3	3	3	3	3	3	3	3	3	2	1	1
5	1	2	3	3	3	3	3	2	2	2	3	3	3	3	3	2	1
6	1	2	3	3	3	3	2	0	0	0	2	3	3	3	3	2	1
7	1	2	2	2	2	2	2	0	2	0	2	2	2	2	2	2	1
8	1	2	0	0	0	0	2	0	0	0	2	0	0	0	0	2	1
9	1	2	0	0	0	0	0	2	2	2	0	0	0	0	0	2	1
10	1	1	2	0	0	0	0	0	0	0	0	0	0	0	2	1	1
11	1	1	1	2	0	0	0	0	0	0	0	0	0	2	1	1	1
12	1	1	1	1	2	2	0	0	0	0	0	2	2	1	1	1	1
13	1	1	1	1	1	1	2	2	2	2	2	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

cela correspond au dessin ci-contre et on peut traduire les 2 premières lignes par :

- Se placer en haut à gauche.
- Prendre le crayon de couleur bleu.
- Colorier 17 cases.
- Passer au début de la ligne suivante.
- Colorier 6 cases.
- Prendre le crayon de couleur noir.
- Colorier 5 cases.
- Prendre le crayon de couleur bleu.
- Colorier 6 cases.
- Passer au début de la ligne suivante.
- *Et ainsi de suite pour les autres lignes...*



Voici l'entête de l'algorithme demandé.

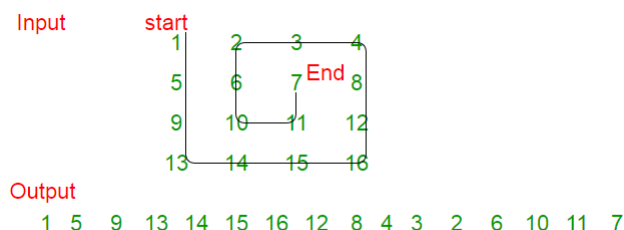
**algorithme** *instructions*(grille : tableau  $n \times m$  d'entiers, couleurs : Liste de chaînes)

### 3 La spirale

(7 points)

Vous recevez un tableau carré (2D) d'entiers et vous créez une liste des entiers lus suivant une spirale commençant dans le coin supérieur gauche et tournant dans le sens inverse des aiguilles d'une montre.

Exemple : Voici une image<sup>1</sup> de cette spirale pour un carré de côté 4 :



Autre exemple : votre algorithme reçoit :

1	2	3	4	5	6
7	8	9	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36

Et il retournera la liste 1, 7, 13, 19, 25, 31, 32, 33, 34, 35, 36, 30, 24, 18, 12, 6, 5, 4, 3, 2, 8, 14, 20, 26, 27, 28, 29, 23, 17, 11, 10, 9, 15, 21, 22, 16.

Aide : il y a de nombreuses manières d'approcher ce problème. Si vous ne voyez pas, en voici quelques unes :

- "Je vais tout droit tant que c'est possible". Ici, l'idée est de se donner une direction de départ, puis d'aller tout droit jusqu'à rencontrer soit le bord du tableau, soit une case déjà visitée. Quand on ne peut plus aller tout droit, il faut tourner. La difficulté est de retenir les cases déjà visitées, par exemple via un tableau de booléens.
- "Je vais tout droit mais je compte mes pas". Par rapport à l'approche précédente, plutôt que de retenir les cases déjà visitées, vous pouvez simplement analyser la situation et voir combien de cases vous devez visiter avant de tourner. Ce nombre diminue de façon prévisible, vous pouvez donc écrire un algorithme grâce à cela.
- "Les carrés imbriqués". Si vous prenez un peu de distance par rapport au problème, vous pouvez constater que suivre la spirale, c'est comme suivre des carrés imbriqués. Dans le premier exemple il y en a deux : un carré 4 x 4, suivi d'un carré 2 x 2. Dans le second exemple il y en a trois.

1. Source : <https://www.geeksforgeeks.org/print-given-matrix-counter-clock-wise-spiral-form> consulté le 1/06/18