

Programmation  
Licence 1 UPEC 2022/2023  
Travaux Machine 2 : Conditions et boucles

## Exercices avec des imbrications d'instructions conditionnelles

### Exercice 1:

Le but de cet exercice est de concevoir un programme permettant d'aider un utilisateur jouant au "421". Dans ce jeu, le joueur et ses adversaires doivent lancer trois dés en différentes phases, et le vainqueur dépend principalement des combinaisons obtenues. Le programme que vous allez écrire a pour but d'indiquer à un utilisateur novice le nom de la combinaison qu'il a obtenue et combien de points elle vaut :

- Le 421 se compose d'un 1, d'un 2 et d'un 4. Il vaut 11 points.
- Une paire d'as se compose de deux dés valant 1 et d'un dé différent. La valeur du dé différent donne le nombre de points obtenus.
- Un brelan se compose de trois dés identiques. La valeur commune indique le nombre de points, sauf 111 qui donne 7 points.
- Une tierce se compose de trois valeurs consécutives, par exemple 4, 2 et 3 forment une tierce. Elle vaut 2 points.
- Toutes les autres combinaisons valent 1 point.

Voici les tâches à réaliser :

- **Question 1** : écrire un programme demandant à l'utilisateur la valeur de chacun des trois dés. L'utilisateur entrera les valeurs des dés dans n'importe quel ordre. Pour l'instant, le programme se contentera d'afficher les valeurs saisies telles quelles.
- **Question 2** : Modifier le programme afin qu'il affiche "Brelan" ainsi que le nombre de points correspondant si les valeurs forment un brelan.
- **Question 3** : Modifier le programme afin qu'il prenne également en compte les paires d'as.
- **Question 4** : Modifier le programme afin qu'il prenne également en compte les 421. Une partie de la difficulté des questions précédentes vient du fait que l'utilisateur entre les valeurs dans l'ordre qu'il souhaite. Si l'on trie les valeurs tout de suite après la saisie, cela simplifie la suite.
- **Question 5** : Ajouter ce qu'il faut à votre programme pour que, juste après la saisie des valeurs par l'utilisateur, les valeurs saisies soient triées. La valeur la plus petite doit être stockée dans une variable deMinimum, la valeur intermédiaire dans une variable deIntermediaire et la valeur maximale dans une variable deMaximum.
- **Question 6** : Modifier le programme afin qu'il prenne également en compte les tierces.
- **Question 7** : Simplifier vos réponses aux questions 2, 3, 4

## Exercices avec des boucles simples

### Exercice 2: Nombres

- Écrire un programme qui demande un nombre à l'utilisateur et affiche les entiers entre 0 et ce nombre. Que se passe-t-il si l'utilisateur entre un nombre négatif?
- Écrire un programme qui affiche les nombres de 9999 à 1
- Écrire un programme qui affiche les nombres pairs de 2 à 8888

— Écrire un programme qui affiche les nombres impairs de 7777 à 33

### **Exercice 3: Table de Multiplication**

Écrire un programme qui demande un nombre  $x$  à l'utilisateur et affiche sa table. Par exemple, pour  $x=6$ , on affichera :

```
0  x 6 = 0
1  x 6 = 6
2  x 6 = 12
3  x 6 = 18
4  x 6 = 24
5  x 6 = 30
6  x 6 = 36
7  x 6 = 42
8  x 6 = 48
9  x 6 = 54
10 x 6 = 60
11 x 6 = 66
12 x 6 = 72
```

Veiller à respecter l'alignement !

### **Exercice 4: Suite de Créteil**

La *Suite de Créteil*  $a_i$  à partir de  $n$  est définie de la façon suivante :

- $a_0 = n$
- si  $a_i$  est un multiple de 3, alors  $a_{i+1} = a_i/3$
- sinon  $a_{i+1} = 2 * a_i + 1$

1. Ecrire un programme qui affiche les  $k$  premiers éléments de la Suite de Créteil, à partir de  $n$ , où  $k$  et  $n$  sont entrés par l'utilisateur.
2. Ecrire un programme qui affiche la Suite de Créteil à partir d'un nombre  $n$  entré par l'utilisateur, jusqu'à ce que on trouve un terme qui est égal à 1

### **Exercice 5: Suite de la tour penchée**

La *Suite de la tour penchée*  $b_i$  est définie de la façon suivante :

- $b_0 = 1, b_1 = 1, b_2 = 1$
- $b_{i+3} = b_i + b_{i+1} + b_{i+2} + i$

1. Ecrire un programme qui affiche les  $k$  premiers éléments de la Suite de la tour penchée où  $k$  est choisi par l'utilisateur.
2. Ecrire un programme qui affiche la Suite de la tour penchée jusqu'à ce que on trouve un terme qui dépasse 50.000.

### **Exercice 6:**

Écrire un programme qui demande à l'utilisateur des nombres et qui s'arrête quand l'utilisateur a entré le même nombre deux fois de suite. Les nombres doivent être saisis un par un et pas tous en même temps (on sait pas encore faire ça en Robusta!)

**Vérifiez votre programme !** Pour cela, vérifiez que votre programme s'arrête bien à la lecture du 2e "5" lorsque vous saisissez chacune des suites suivantes :

- 1, 3, 5, 5.
- 1, 2, 3, 5, 5.

Essayez ensuite de concevoir un programme qui demande à l'utilisateur d'entrer des nombres et qui s'arrête quand l'utilisateur a entré deux fois le même, mais pas forcément de suite. Discutez votre idée avec votre enseignant encadrant pour identifier les difficultés de conception que vous rencontrez.

### **Exercice 7: Jeu de Nim à trois piles**

Récupérer le fichier du jeu de nim se trouvant sur EPREL. Modifier le programme afin de piocher les allumettes non pas dans une pile de 21 allumettes, mais dans 3 piles de 7 allumettes. Un joueur ne peut piocher que dans une pile par tour.

### **Exercice 8: Réalisez un programme qui permet de jouer plusieurs parties de 421.**

Le fonctionnement d'une partie sera le suivant :

1. Tirer au hasard la main de chaque joueur.
2. Décider quelle main gagne.
3. Décider le gagnant au bout de 14 manches.

Si vous avez le temps, ajouter un système de mise qui marcherait de la façon suivante :

1. Chaque joueur a un crédit initial.
2. Après avoir vu sa main, le joueur 1 mise une somme ou se retire.
3. Après avoir vu sa main, le joueur 2 égalise la mise, la double, ou se retire.
4. Après avoir vu sa main, le joueur 1 égalise la mise, ou se retire (et perd sa première mise).
5. Le gagnant remporte la mise.
6. Recommencer en changeant le premier joueur qui mise.
7. La partie se termine quand un joueur est en faillite ou au bout de 254 manches.