Algorithmes Séance 5

Structures Itéartives à Condition

Chiffres aléatoires

(1) Ecrire un algorithme qui génère une séquence de 5 chiffres aléatoires.

Exemple 1	*Exemple 2*	*Exemple 3*
14828	02989	50163

```
Algorithme Sequence_Aléatoire_1
Début
  seq ← ""
  Pour i de 0 à 4 faire
    seq ← seq + ConvCh(aléa(0, 9))
  Fin Pour
  Ecrire(seq)
Fin
```

(2) Ecrire un algorithme qui génère une séquence de chiffres aléatoires qui se termine deux chiffres égaux.

- Est-ce qu'on peut utiliser la structure **pour** de l'algorithme précédent ?
- Quelle est la structure la plus appropriée ?

Exemple 1	*Exemple 2*	*Exemple 3*
4932966	77	92645656207383713411

```
Algorithme Sequence_Aléatoire_2
Début
 seq ← ""
 Répéter
   seq ← seq + ConvCh(aléa(0, 9))
 Jusqu'à (Long(seq) >= 2) et
         (seq[Long(seq)-1] = seq[Long(seq)-2])
 Ecrire(seq)
Fin
```

```
Algorithme Sequence_Aléatoire_3
Début
 seq ← ""
 TantQue (Long(seq) < 2) ou
         (seq[Long(seq)-1]≠seq[Long(seq)-2]) Faire
   seq ← seq + ConvCh(aléa(0, 9))
 Fin TantQue
 Ecrire(seq)
Fin
```

(3) Ecrire un algorithme qui génère une séquence aléatoire de 5 chiffres distincts (différents). Utiliser deux méthodes.

Exemple 1	*Exemple 2*	*Exemple 3*
71980	72891	65218

```
Algorithme Sequence_Aléatoire_4
Début
 seq ← ""
 Pour i de 0 à 4 Faire
   Répéter
     car ← ConvCh(aléa(0, 9))
   Jusqu'à (pos(car, seq) = -1)
   seq ← seq + car
 Fin pour
 Ecrire(seq)
Fin
```

```
Algorithme Sequence_Aléatoire_1
Début
 seq ← ""
 Pour i de 0 à 4 Faire
   existe ← Vrai
   TantQue existe Faire
      car ← ConvCh(aléa(0, 9))
      existe ← pos(car, seq) ≠ -1
   Fin TantQue
   seq ← seq + car
 Fin pour
 Ecrire(seq)
Fin
```

Puzzle Game

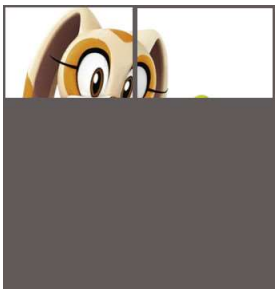
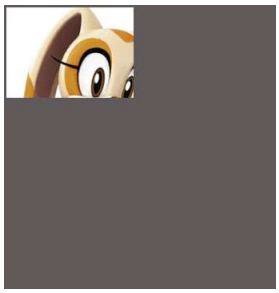
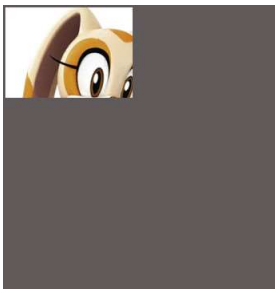
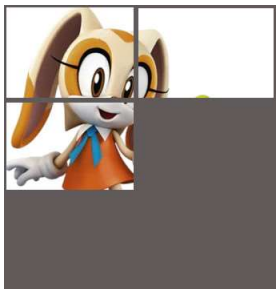
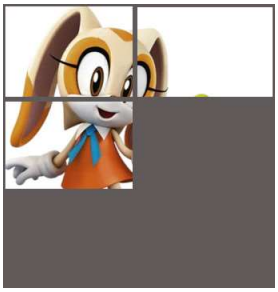
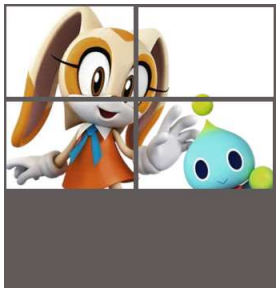
Pour s'amuser un peu le petit abdou a inventé un jeu. Le jeu nécessite un puzzle à six pièces et un dé à six faces.
Le but du jeu est de reconstituer le puzzle en suivant les règles en un minimum de coups.
Au début, le puzzle est démonté et le jeu se déroule comme suit :

- On lance le dé deux fois en notant sa valeur à chaque fois `de1` et `de2`
- On décide de :
  - Monter une pièce du puzzle, si `de2 > de1`
  - Monter deux pièces du puzzle, si `de2 = de1`
  - Démonter une pièce du puzzle, si `de2 < de1`
- Le jeu se termine lorsque le puzzle est reconstitué.

Figure 1, Puzzle Complet

Figure 2, Dé à six faces

Le tableau suivant indique l'état du puzzle avant et après le lancement du dé.

Avant	de1 / de2	Après	Explication
	de1  de2 		de2 < de1 → enlèver une pièce
	de1  de2 		de2 = de1 → ajouter deux pièces
	de1  de2 		de2 > de1 → ajouter une pièce

Travail demandé

1. Utiliser les briques suivantes pour compléter les étapes du programme, **Utiliser uniquement la lettre adéquate** :
   A. Ajouter/Enlever des pièces au puzzle (variable `pieces`) en fonction de la valeur des dés `de1` et `de2`
   B. Initialiser à zéro le nombre de pièces `pieces` du puzzle
   C. Répéter les étapes 2, 3 et 4 jusqu'à `pieces = 6`
   D. Afficher le nombre actuel des pièces `pieces` du puzzle
   E. Lancer le dé deux fois, noter la 1ère valeur dans `de1` et noter la 2ème valeur dans `de2`

Num. étape	1	2	3	4	5
Etape	B				

2. Ecrire l'algorithme de chaque étape.

```
Algorithme Puzzle_Game
Début
  pieces ← 0
  Répéter
    de1 ← aléa(1, 6)
    de2 ← aléa(1, 6)
    Ecrire("Dé 1 =", de1, "- Dé 2 =", de2)
    Si de2 > de1 Alors
      pieces ← pieces + 1
    Sinon Si de1 > de2 et pieces - 1 ≥ 0 Alors
      pieces ← pieces - 1
    Sinon Si de1 = de2 et pieces + 2 ≤ 6 Alors
      pieces ← pieces + 2
    Fin Si
    Ecrire(pieces, "sont montées")
  Jusqu'à pieces = 6
Fin
```

Jeu d'alumettes

Le jeu d'allumettes se joue entre deux joueurs :

- On saisit le nombre d'allumettes `n` (un entier impair entre 15 et 35).
- On saisit les pseudos des deux joueurs `jo[0]` et `jo[1]`.
- Chaque joueur prend de 1 à 3 alumettes. Celui qui prend la dernière perd.

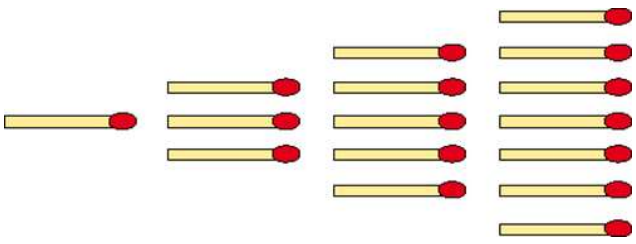


Figure 9, Jeu d'allumettes

On demande d'écrire un algorithme qui simule ce jeu.

```
Algorithme allumettes
Début
  Répéter
    Ecrire("Nombre d'allumettes [15, 35] ? "); Lire(n)
  Jusqu'à (15 ≤ n ≤ 35)
  Pour i de 0 à 1 Faire
    Répéter
      Ecrire("Pseudo joueur", i+1, "? ");Lire(jo[i])
    Jusqu'à jo[i] ≠ ""
  Fin Pour
  gagne ← -1
  nj ← 0
  nma ← 3
  TantQue gagne = -1 Faire
    Ecrire(jo[nj], "joue")
    si nma > n alors nma ← n fin si
    Répéter
      Ecrire("Nbre d'allumettes [1,", nma,"] ? "); Lire(nba)
    Jusqu'à 1 ≤ nba ≤ nma
    n = n - nba
    Ecrire("Nombre d'allumettes restantes :", n)
    nj ← (nj + 1) mod 2
    Si n = 0 Alors
      gagne ← nj
    Fin Si
  Fin TantQue
  Ecrire(jo[gagne], "gagne la partie.")
Fin
```