Programmation
Licence 1 UPEC 2022/2023
Travaux Machine 4 : Tableaux

Exercices

Exercice 1:

Écrire un programme qui tire un message au hasard parmi 5 messages fixés (que vous pouvez choisir vous-mêmes) et l'affiche. On créera, dans le programme, un tableau qui contiendra tous les 5 messages choisis par vous, de sorte qu'à chaque tir c'est le message d'indice correspondant qui est affiché.

Exercice 2:

Écrire un programme qui demande des nombres entre 0 et 99 à l'utilisateur et qui s'arrête dès qu'il a entré trois fois le même nombre. Par exemple : 1 2 3 1 7 4 23 4 23 0 16 12 23 (23 apparait trois fois, le programme s'arrête maintenant).

Vous devez utiliser un tableau d'entiers de taille 100, dont chaque case représentera le nombre de fois que l'indice a été saisi par l'utilisateur. Par exemple, pour la suite ci-dessus, ce tableau d'entiers démarrera rempli de 0 partout, puis, au fur et à mésure que les nouvelles valeurs sont saisies, le tableau évoluera de la façon suivante (on affiche que les cases entre 0 et 23) :

1. 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
2. 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
3. 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
4. 0 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
5. 0 2 1 1 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
6. 0 2 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0.
7. 0 2 1 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.
8. 0 2 1 1 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1.
9. 0 2 1 1 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2.
10. 1 2 1 1 2 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2.
11. 1 2 1 1 2 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 2.
12. 1 2 1 1 2 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 3. ´

Exercice 3:

Écrire un programme qui demande des nombres entre 0 et 19 à l'utilisateur et qui s'arrête quand l'utilisateur a tapé au moins une fois chacun de ces nombres. **Idée d'attaque :** Utiliser un tableau de taille 20, et une variable qui compte le nombre d'entiers différents que l'utilisateur a déjà tapé. Par exemple, s'il tape 1 2 3 1 0 4 13 4 13 6 11 12 13, cette variable contiendra 9.

Exercice 4:

Écrire un programme qui demande une série de nombres à l'utilisateur et stocke toute la série dans un tableau, puis demande un nombre supplémentaire `k` et multiplie chaque élément de la série par `k`. À la fin on affiche la nouvelle série.

Exercice 5:

Écrire un nouveau programme qui demande une série de nombres à l'utilisateur, les stocke dans un tableau, puis calcule le minimum de ce tableau.

Exercice 6:

Écrire un nouveau programme qui demande une série de nombres à l'utilisateur (vue comme un vecteur $v1$), puis une deuxième série (vue comme un vecteur $v2$), puis, si les deux séries ont le même nombre d'éléments, le programme affiche le vecteur $v1 + v2$. Si les deux séries n'ont pas le même nombre d'éléments, le programme affiche un message d'erreur.

Exercice 7:

Un palindrome est une suite (de nombres par exemple) où le premier nombre est identique au dernier, le 2e à l'avant-dernier, etc. Par exemple, 1,4,3,2,5,11,5,2,3,4,1 est un palindrome, alors que 1,4,3,2,5,11,5,2,4,3 ne l'est pas. Écrire un programme qui demande une série de nombres à l'utilisateur, puis vérifie si la série est un palindrome.

Exercice 8:

Le but de cet exo est de découvrir le fonctionnement et de réutiliser du code construit ailleurs. Concrètement, vous êtes censés compléter un jeu qui sert à dessiner un cochon avec les règles suivantes :

- il a un corps, une tête, deux oreilles, deux yeux, quatre pattes et une queue.
- le corps correspond à 1, les oreilles à 2, les yeux à 3, les pattes à 4, la tête à 5 et la queue à 6

À chaque nombre entré, l'ordinateur ajoute au cochon la partie du corps correspondante, sauf s'il en a suffisamment. Le jeu s'arrête quand le cochon est complet.

Une partie du programme qui sert à l'affichage du cochon en asciiart est disponible sur eprel, téléchargez le fichier `td4_dessin_cochon.txt`. Vous remarquerez l'utilisation d'un tableau (variable `cochon`) dont les valeurs contrôlent l'affichage du cochon.

1. Écrivez un programme qui contient ces lignes de code et les complètent, en rajoutant la déclaration de la variable `cochon` et son initialisation avec différentes valeurs. Notez le comportement du programme en fonction de valeurs d'initialisation : quelle image s'affiche et pour quelles valeurs initiales le cochon s'affiche complètement (sans élément d'image manquant).
2. Modifier votre programme pour qu'il initialise le tableau à 0, puis demande des nombres entre 1 et 6 à l'utilisateur (qui sont censés correspondre à des jets d'un vrai dé) et incrémente la case respective du tableau. Enfin, le programme devrait s'arrêter lorsque le dessin du cochon est complet – donc lorsque les valeurs que vous avez remarqué au point précédent qu'elles impliquent l'affichage complet du cochon sont atteintes.

Exercice 9: **Le jeu du pendu (exo facultatif).**

Le but de cet exercice est de réaliser un programme permettant de jouer au "pendu".

- Version sans décompte des erreurs Le programme aura en mémoire un mot. A chaque tour, le programme demandera à l'utilisateur de proposer un caractère, et affichera le mot en mémoire en masquant les lettres qui n'ont pas encore été proposé par l'utilisateur. Le programme s'arrêtera lorsque toutes les lettres du mot en mémoire auront été proposées. **Il y a de nombreuses façons de s'y prendre, la votre n'a pas besoin d'être celle de votre voisin !**
- Version avec décompte des erreurs et affichage du pendu Modifier le programme précédent afin que le programme s'arrête également si l'utilisateur a fait trop d'erreurs. On pourra si on le souhaite afficher un pendu de plus en plus complet à mesure que le nombre d'erreurs de l'utilisateur augmente.