

La plupart des exercices ont été imaginés/proposés par Henri Garreta.

1 Boucles et affichages

1.1 Tables de multiplication

Ecrire un programme qui affiche la table de multiplication de 5 :

```
1 x 5 = 5
2 x 5 = 10
...
10 x 5 = 50
```

Modifier le programme pour qu'il affiche les tables de multiplication de tous les entiers de 2 à 9.

1.2 Triplets pythagoriciens

Ecrivez un programme qui calcule tous les "triplets pythagoriciens" inférieurs à une valeur MAX donnée par l'utilisateur, c'est à dire les triplets (i, j, k) d'entiers positifs inférieurs à MAX tels que $i^2 + j^2 = k^2$ (par exemple (3,4,5) est un triplet pythagoricien).

1.3 Lancers de dés

On s'intéresse à un jeu de lancer de dés à 6 faces. On veut simuler la distribution de probabilité de la somme obtenue.

1. Ecrire un premier programme qui réalise N lancers d'un seul dé et qui calcule la distribution des valeurs obtenues, c'est-à-dire le nombre de fois où chacune des six faces apparaît au cours des N lancers. Vous définirez pour cela un tableau/une liste d'entiers dont vous incrémenterez la case i chaque fois que la face i est obtenue ($1 \leq i \leq 6$). Pour terminer, vous afficherez le contenu du tableau.
2. Ecrire un second programme qui réalise N lancers de 2 dés et qui calcule, pour chaque lancer, la somme des 2 dés. Pour obtenir la distribution de probabilité de la somme, vous définirez un tableau d'entiers dont vous incrémenterez la case i chaque fois que la somme obtenue a pour valeur i ($2 \leq i \leq 12$). Vous réaliserez ensuite un affichage graphique de la distribution obtenue sous la forme ci-dessous (ici, pour N=50) : sur chaque ligne, le nombre d'étoiles correspond au nombre de fois où la somme indiquée à gauche a été obtenue.

```
2 **
3 ****
4 ***
5 *****
6 *****
7 *****
8 *****
9 ***
10 **
11 ***
12 *
```

2 Fonctions

2.1 Recherche d'un élément dans une séquence

1. Écrire une fonction

`recherche_dicho (T, x)`

qui recherche et renvoie l'indice de l'élément x dans la séquence d'entiers T , dont les éléments sont rangés par ordre croissant. Cette fonction renvoie -1 si x n'appartient pas à T .

2. Modifier la fonction en

`recherche_dicho2 (T, x)`

pour qu'elle fonctionne si la séquence croissante a subi une permutation circulaire (par exemple $[5, 6, 1, 2, 3]$ est une permutation circulaire de la séquence croissante $[1, 2, 3, 4, 5, 6]$). On ne connaît évidemment pas le décalage.

2.2 Opérations sur les séquences

1. Écrire une fonction

`tri_selection(T)`

qui trie les éléments de la séquence T par ordre croissant en utilisant l'algorithme de tri par sélection décrit en cours et dont l'invariant est : « à l'étape i , je recherche le i -ème élément de la séquence et je le place à l'indice i de la séquence. »

2. Écrire une fonction

`compte_caracteres(T)`

qui renvoie un tableau associatif faisant le compte de chacun des caractères présents dans la chaîne de caractères T .

3. Écrire une fonction

`fusionne(T1, T2)`

qui, étant donné les séquences d'entiers triées par ordre croissant $T1$ et $T2$, calcule et renvoie une liste qui est la fusion de ces deux listes. Si les séquences sont

$T1 = \{1, 3, 7, 8, 8, 9, 24\}$

et

$T2 = \{2, 4, 6, 7, 11\}$,

la fonction doit renvoyer

$\{1, 2, 3, 4, 6, 7, 7, 8, 8, 9, 11, 24\}$.