Série d’exercices sur les tableaux

Ex1 – solBaseTableau.m
Initialisez, de façon statique (utilisez zeros), un tableau ligne de 20 éléments et remplissez le avec les nombres de 1 à 20.

Ex2
Initialisez , de façon statique (utilisez zeros), un tableau ligne de 20 éléments et remplissez le avec les nombres de 21 à 40.

Ex3
Initialisez , de façon statique (utilisez zeros), un tableau colonne de 20 éléments et remplissez le avec les nombres de 21 à 40.

Ex4
Initialisez un tableau vide (utilisez []) et remplissez-le, un nombre à la fois, avec les nombres de 21 à 40, à l’aide de l’opérateur de concaténation. indice: tab = [tab 12];

Ex5
Initialisez deux tableaux lignes (avec zeros). Le premier doit contenir les nombres de 40 à 21 et le second de 20 à 1. Concatenez les pour obtenir un troisième tableau qui contient les nombres de 40 à 1.

Série d’exercices sur les tableaux

Ex6
Refaite l’exercice 4 avec un tableau colonne. indice: tab = [tab; 12]

Ex7
Refaite l’exercice 5 avec des tableaux colonnes.

Ex8 - solTrouverMin1.m
Écrivez une fonction qui reçoit un tableau comme paramètre d’entrée et retourne le nombre minimum contenu dans ce tableau.

Ex9 - solTrouverMin2.m
Modifiez le code de l’exercices 8 pour que la fonction retourne le minimum du tableau comme premier paramètre de sortie et l’index où se trouve la première occurence de ce nombre.

Ex: tab = [104 54 76 89 54 99];
[minimum index] = trouver_min2(tab);

minimum est égal à 54
index est égal à 2

Série d'exercices sur les tableaux

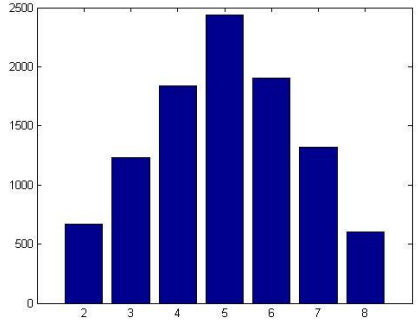
Ex10 solSommeDe.m

Dans une procédure, simulez le lancer de deux dés à 4 faces et additionnez leur résultat. Vous devez simulez ce lancer de dé 10000 fois et compter le nombre d'occurence de chacune des valeurs résultantes possible: {2,3,4,5,6,7,8}. Vous avez déjà résolu ce problème lors du mini-test, mais faites le maintenant en utilisant un tableau.

Question à répondre pour vous même, si vous comparez cette solution avec celle du mini-test, de quel façon est-ce que les tableaux ont simplifié le code nécessaire à la résolution de ce problème?

Est-ce que l'utilisation de tableau rend humainement possible le développement du code nécessaire au lancé de deux dé à 1000 faces?

Pour le plaisir, si vous fournissez votre tableau de comptes ainsi qu'un tableau contenant les valeurs possibles à la fonction bar (voir solution), voici ce que vous obtiendrez:



Série d’exercices sur les tableaux

Ex11 solTracerCercle.m
Vous devez écrire une procédure permettant de tracer un cercle. Elle doit recevoir comme paramètres d’entrées le rayon du cercle et le nombre de points dont doitêtre composée le cercle.

1) La première étape est de déterminer les points qui devront être généré. Évidemment, plus le nombre de points requis sera grand, plus grande sera la précision du cercle. Une façon de procéder est de travailler en radian. Sachant que le cercle est défini de 0 à 2π, il est possible de déterminer l’espacement requis entre les points d’un ensemble pour que ceux-ci soit espacés uniformément autour du cercle, en divisant 2π par le nombre de points requis moins 1.

Définissez une fonction privée qui reçoit en paramètre d’entrée le nombre de points et qui retourne une liste de tous les angles qui devront être utilisé. Attention, 0 et 2π doivente faire parti de l’ensemble.

Exemple, si on demande un cercle à 8 points, l’ensemble des angles seront:

Par saut de 2π/7

0, 0.8976, 1.7952, 2.6928, 3.5904, 4.4880, 5.3856, 6.2832 (2π)

Série d'exercices sur les tableaux

2) Maintenant que l'on a la liste des angles, il faut générer les coordonnées cartésiennes correspondante. Pour ce faire, il suffit d'utiliser les relations trigonométriques suivantes:

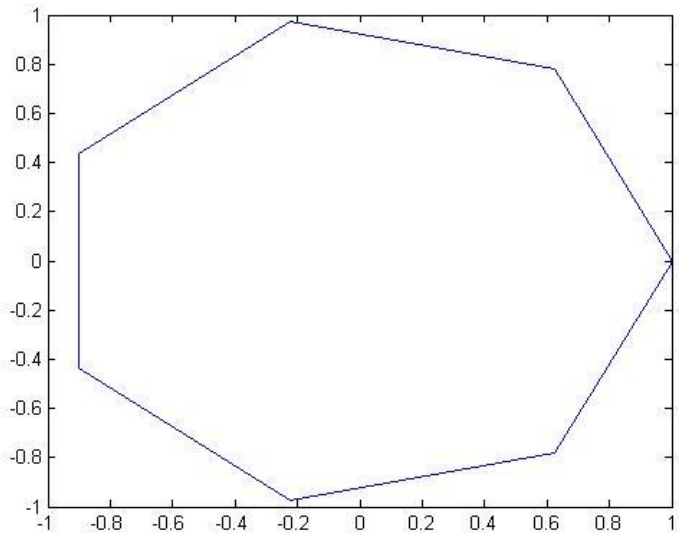
$$x = r * \cos(a) \qquad y = r * \sin(a)$$

Toujours selon l'exemple à 8 points (avec un rayon de 1), on peut remplir la table suivante (vous devez utiliser un tableau pour chaque colonne):

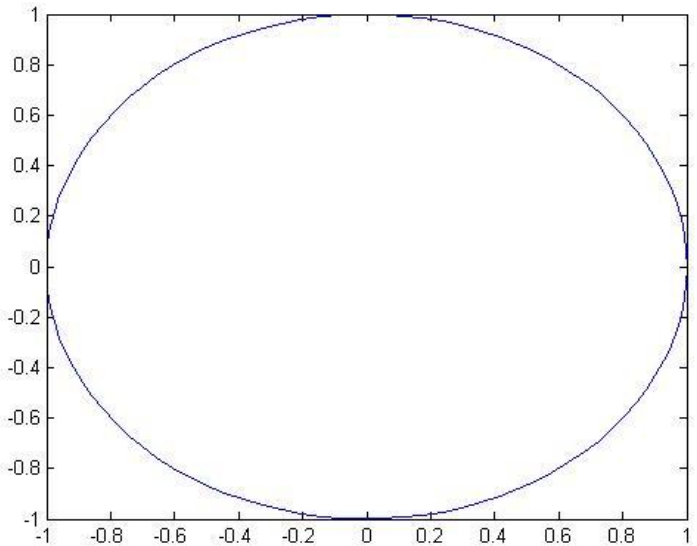
a	x	y
0.0000	1.0000	0.0000
0.8976	0.6235	0.7818
1.7952	-0.2225	0.9749
2.6928	-0.9010	0.4339
3.5904	-0.9010	-0.4339
4.4880	-0.2225	-0.9749
5.3856	0.6235	-0.7818
6.2832	1.0000	0.0000

3) Si vous affichez maintenant le résultat à l'aide de la procédure plot, voici ce que vous devriez obtenir:

Pour 8 points:



Pour 100 points:



Série d’exercices sur les polynomes

Ex1 evaluerPolynome.m
Écrivez une fonction permettant d’évaluer la valeur d’un polynôme encodé avec la méthode Matlab. Il s’agit principalement d’implémenter la fonction **polyval**.

Ex2 polynomeDerivee.m
Écrivez une fonction permettant d’évaluer la dérivée d’un polynôme encodé avec la méthode Matlab. Il s’agit principalement d’implémenter la fonction **polyder**.

Ex3 sommePolynome.m
Écrivez une fonction permettant de calculer la somme de deux polynômes encodés avec la méthode Matlab.