

Série d'exercices sur les fonctions

Ex1 – calculerAireCercle.m

Écrivez une fonction appelée *calculerAireCercle*, qui reçoit un paramètre en entrée, *rayon*, et retourne un paramètre de sortie, *aire*. La fonction doit calculer l'aire d'un cercle en utilisant la formule:

$$aire = \pi * rayon^2$$

Ex2 – calculerAireRectangle.m

Écrivez une fonction appelée *calculerAireRectangle*, qui reçoit deux paramètres en entrée, *longueur* et *largeur*, et retourne un paramètre de sortie, *aire*. La fonction doit calculer l'aire d'un rectangle en utilisant la formule:

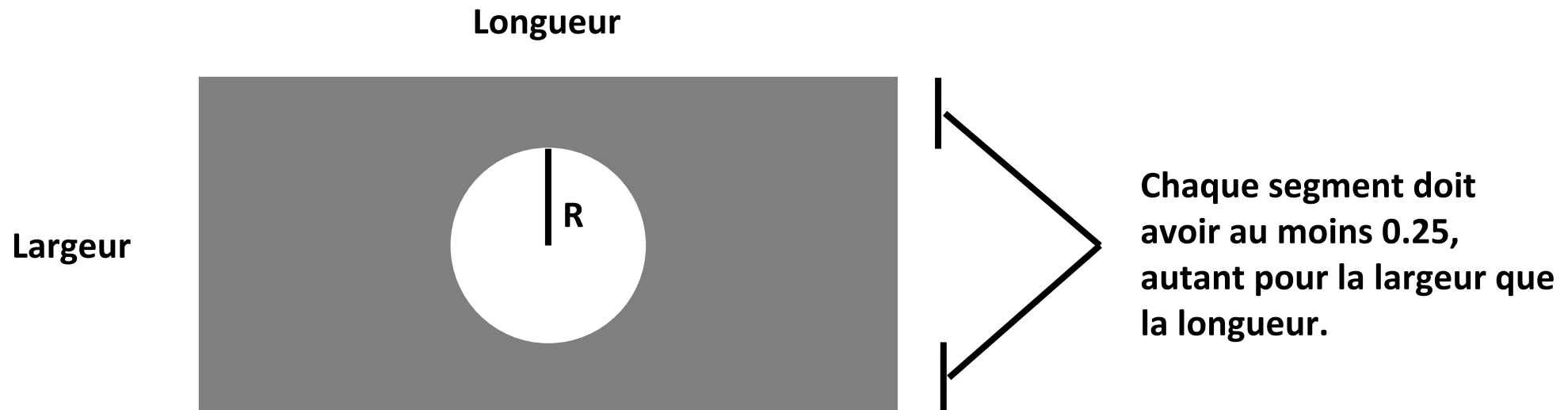
$$aire = longueur * largeur$$

Série d'exercices sur les fonctions

Ex3 – calculerAirePiece.m

Écrivez une procédure appelé *calculerAirePiece* qui demande à l'utilisateur d'entrer les dimensions d'un rectangle (longueur et largeur) et le rayon d'un cercle. La procédure vérifie ensuite que le diamètre du cercle est plus petit que la longueur et la largeur en comptant un facteur de sécurité égal à 0.5 (le facteur de sécurité est une constante). Si le facteur de sécurité n'est pas respecté, un message en informe l'utilisateur et la procédure se termine.

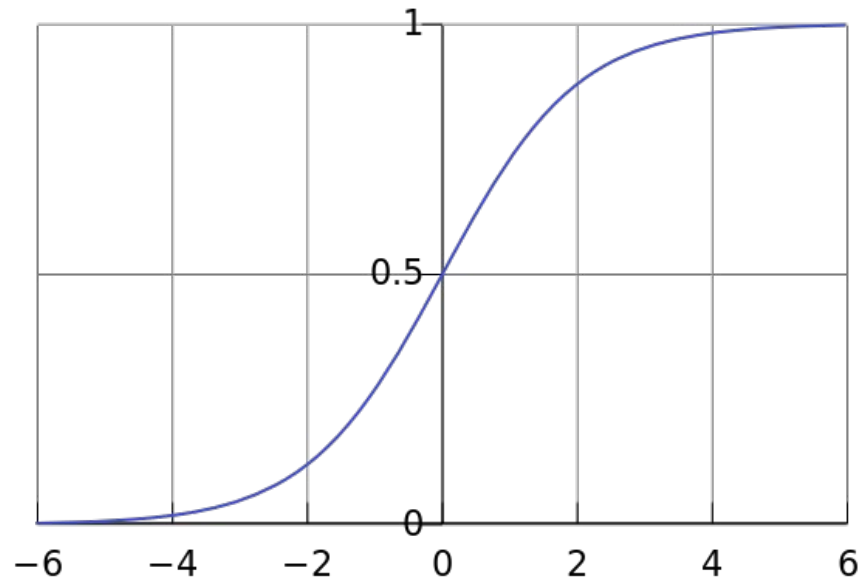
Si le facteur de sécurité est respecté, la procédure fait ensuite appel au deux fonctions défini précédemment pour calculer l'aire de la surface en gris dans la figure qui suit. Pour ce faire il suffit de soustraire l'aire du cercle à l'aire du rectangle. La procédure affiche l'aire calculé à l'écran, avec 4 chiffres après la virgule.



Série d'exercices sur l'utilisation des fonctions Matlab

Ex4 – estimationLogistique.m

La courbe logistique est illustré dans la figure qui suit (en bas à gauche) et son équation est donnée en bas à droite. Écrivez une fonction qui permet d'obtenir la valeur d'un point x sur cette courbe.



$$f(x) = \frac{1}{1 + e^{-x}}$$

Série d'exercices sur l'utilisation des fonctions Matlab

Ex5 – estEntier.m

Écrivez une fonction appelé *estEntier* qui permet de tester si le nombre reçu en paramètre est un entier. Attention, vous ne devez pas utiliser la fonction *isinteger* de l'environnement Matlab, mais plutôt la fonction *fix*. Pour savoir si un nombre est un entier, il suffit de l'arrondir avec *fix* et de comparer le résultat au nombre original. Si le nombre est un entier, l'arrondir n'aura aucun effet sur sa valeur et le résultat de la comparaison sera positif.

La fonction doit retourner 1 si le nombre reçu en paramètre est un entier et 0 sinon.

Ex6 – estimTempsCroissancePop.m

La croissance d'une population de levure requise dans un procédé de fermentation de moût double à chaque minute. Il s'agit d'un rythme de croissance exponentielle. Nous voulons une fonction qui permet de savoir quand un certain nombre de cellules à été atteint. La solution à ce problème se trouve en solutionnant:

$$temps_requis = \frac{\log_e(pop_totale)}{\log_e(2)}$$

Écrivez une fonction qui reçoit en paramètre un nombre pour la population totale et qui retourne le temps requis pour l'atteindre.

Série d'exercices sur le switch

Ex7 – switchBase1.m

Écrivez une procédure qui affiche un menu comme suit:

- 1- Nord
- 2- Sud
- 3- Est
- 4- Ouest

Dans le programme, on doit retrouver une série de déclarations de constantes, une pour chacune des options précédentes. La valeur de chaque constante doit correspondre à son numéro dans le menu.

Après avoir présenté le menu, le programme demande à un usager d'entrer un choix, parmi les options.

Dépendant de l'option, le programme affiche:

Vous allez vers le {nord/sud/est/ouest}

Si le choix entré ne correspond à aucune option, le programme affiche:

Choix invalide!

Série d'exercices sur le for

Ex8 – forBase1.m

Écrivez une procédure qui contient les boucles for suivantes une à la suite de l'autre. Pour chaque boucle, le programme doit écrire chacune des valeurs du compteur.

boucle 1	boucle 2	boucle 3	boucle 4
départ: 1	départ: 10	départ: 100	départ: -100
incrément: 1	incrément: -1	incrément: 5	incrément: 50
fin: 10	fin: -20	fin: 200	fin: 100

Ex9 – forBase1.m

Écrivez une procédure qui contient deux boucles for imbriquées. La première va de 1 à 100 avec des incréments de 2 et la deuxième va de la valeur du compteur de la première boucle à 100 par des incréments de 2.

Ex, si la première boucle est rendu à 25, la deuxième va de 25 à 100, en incréments de 2

Affichez les valeurs du compteurs de la second boucle.

Série d'exercices sur le for

Ex10 – calculInterets.m

L'équation décrivant la valeur d'un placement soumis à un taux d'intérêt composé est la suivante:

$$\mathit{montant_final} = \mathit{montant_initial}(1 + \mathit{taux_interet})^{\mathit{duree}}$$

ou on a le montant initial, le montant final, le taux d'intérêt et la durée du placement.

Écrivez une fonction qui reçoit chacune des variables en paramètre et exécute le calcul en utilisant une boucle for. Pour ce faire, ne pas utiliser l'exposant ^durée, mais calculer la multiplication des taux d'intérêts de façon itérative.

Pour une durée de 5 par exemple, le calcul exécuté est:

$$\mathit{montant_final} = \mathit{montant_initial} * (1 + \mathit{taux_interet}) * (1 + \mathit{taux_interet}) * (1 + \mathit{taux_interet}) * (1 + \mathit{taux_interet}) * (1 + \mathit{taux_interet})$$

Série d'exercices sur le for

Ex11 – afficherCarreeEtoile.m

Écrivez une procédure qui reçoit en argument un nombre entier (utilisez la fonction écrite précédemment Ex.5). La procédure affiche un carré formé d '*', dont les dimensions sont définies par le nombre reçu en paramètre.

Ex: si la fonction a été appelé avec le nombre 5, la procédure affiche:

```
*****  
*****  
*****  
*****  
*****
```


Série d'exercices sur le for

Ex12 – estPremier.m

Écrivez une fonction qui permet de déterminer si un nombre est premier. Pour savoir si un nombre n est premier, il faut tester chacun des nombres entre 2 et \sqrt{n} , et déterminer si l'un d'entre eux est un diviseur de n (utiliser le modulo). Pour optimiser l'algorithme, tester d'abord si le nombre est pair, si c'est le cas, il n'est pas premier. Vous n'aurez pas à tester 2 comme diviseur par la suite. La fonction retourne 1 si le nombre est premier et 0 s'il ne l'est pas. Le nombre 2 est premier et doit être considéré séparément.

Ex13 – listePremier.m

Écrivez une procédure qui demande un nombre à un usager, la procédure affiche ensuite tous les nombres premiers entre 2 et le nombre reçu en paramètre (calculé entre 3 et le nombre et faire +1, puisque 2 est premier). La procédure utilise la fonction précédente. À la fin de l'exécution, la procédure affiche le nombre de nombres premiers qui ont été trouvés.