

Révision Intra

Instructions

Complétez chaque exercices du document à l'aide des notions vues en classe. Complétez chaque exercice dans le temps alloué. Faites les exercices sur papier pour se pratiquer à l'environnement de l'examen.

Question 1.

6 minutes

Écrivez un script qui saisit un nombre au clavier et affiche tous les chiffres de ce nombre l'un après l'autre. Assumez que la valeur saisie sera un nombre entier positif.

Solution 1.

```
saisie = input('Entrez une valeur : ');  
  
while (saisie > 0)  
    fprintf('%g\n', mod(saisie, 10));  
    saisie = fix(saisie / 10);  
end
```

Question 2.

5 minutes

Écrivez une fonction qui reçoit trois nombres. Le premier est une valeur à tester et les deux autres sont les limites minimum et maximum d'un intervalle. La fonction retourne vrai si la valeur à tester est à l'intérieur de l'intervalle donné. Validez que les paramètres sont des entités numériques avec la fonction `isnumeric`. Écrivez ensuite un script de test qui va tester l'utilisation de la fonction précédente en utilisant la fonction `assert`. Faites 2 tests à l'intérieur du script de test.

Solution 2.

```
function [valide] = estDansIntervalle(nombre, min ,max)
assert(isnumeric(nombre) && isnumeric(min) ...
       && isnumeric(max))
valide = nombre < max && nombre > min;
end
```

```
assert(estDansIntervalle(15, 10, 20) == true)
assert(estDansIntervalle(10, 100, 200) == false)
```

Question 3.

6 minutes

Écrivez une fonction qui reçoit un nombre entier positif et affiche tous les diviseurs de ce nombre. Écrivez ensuite un script qui saisit un nombre et affiche tous les diviseurs de ce nombre en utilisant la fonction créé précédemment.

Solution 3.

```
function q31AfficheDiviseur(nombre)
for i=1:nombre
    if (mod(nombre, i) == 0)
        fprintf('%g\n', i);
    end
end
```

```
saisie = input('Entrez une valeur : ');
q31AfficheDiviseur(saisie)
```

Question 4.

6 minutes

Écrivez une fonction qui reçoit deux nombres entiers positifs et retourne le nombre de diviseurs communs à ces deux nombres. À titre d'exemple, les nombres 8 et 12 possèdent 3 diviseurs communs (1, 2 et 4).

Solution 4.

```
function [nDiviseur] = nDiviseurCommun(nb1, nb2)
if nb1 < nb2
    minimum = nb1;
else
    minimum = nb2;
end

nDiviseur = 1;
for i=2:minimum
    if mod(nb1, i) == 0 && mod(nb2, i) == 0
        nDiviseur = nDiviseur+1;
    end
end
end
```

Question 5.

5 minutes

Le nombre Pi peut être calculé à l'aide de la série suivante :

$$\sqrt{12} \sum_{k=0}^{\infty} \frac{(-3)^{-k}}{2k+1}$$

Écrivez un script qui approxime la valeur de Pi en effectuant la somme des 1000 premiers termes de la série ci-dessus et qui affiche la valeur obtenue.

Solution 5.

```
NB_TERMES = 1000;
somme = 0;

for i=0:NB_TERMES-1
    somme = somme + (-3)^-i/(2*i+1);
end

fprintf('Approximation de pi : %g\n', sqrt(12)*somme);
```

Question 6.

7 minutes

Écrivez une fonction qui reçoit un nombre de dés n , simule le lancé de n dés à 6 faces et retourne le nombre de 6 obtenus.

Indice : La fonction `randi(X)` retourne un nombre entier aléatoire entre 1 et X .

Solution 6.

```
function [n6] = n6SurND6(nD6)
n6 = 0;

for i=1:nD6
    if randi(6) == 6
        n6=n6 + 1;
    end
end
end
```

Question 7.

7 minutes

Écrivez une fonction qui reçoit un tableau de nombres, un nombre minimum et un nombre maximum. La fonction retourne le nombre de valeurs du tableau entre le minimum et le maximum.

Solution 7.

```
function [n] = nDansIntervalle(tab, min, max)
n = 0;
for i=1:numel(tab)
    if estDansIntervalle(tab(i), min, max)
        n = n + 1;
    end
end
end
```

Question 8.

7 minutes

Écrivez une fonction qui reçoit un tableau de nombres, un nombre minimum et un nombre maximum. La fonction retourne un tableau des valeurs du tableau initiale entre le minimum et le maximum.

Solution 8.

```
function [tabIntervalle] = dansIntervalle(tab, min, max)
tabIntervalle = [];
for i=1:numel(tab)
    if estDansIntervalle(tab(i), min, max)
        tabIntervalle = [tabIntervalle, tab(i)];
    end
end
end
```


Question 9.

10 minutes

Écrivez une fonction qui reçoit un tableau de nombres réels et une valeur « limite ». Elle retourne le nombre de valeurs dans le tableau supérieures ou égales à la valeur limite et la moyenne de ces valeurs.

Solution 9.

```
function [ nSuperieure, moyenne ] = ...
    nSuperieurMoyenne(tab, limite)
somme = 0;
nSuperieure = 0;
moyenne = 0;

for i=1:numel(tab)
    if tab(i) >= limite
        nSuperieure = nSuperieure + 1;
        somme = somme + tab(i);
    end
end

if nSuperieure > 0
    moyenne = somme / nSuperieure;
end
end
```