Bing

Oui, voici un autre QCM de 5 questions de type output code en pseudo code. Chaque question a une seule bonne réponse. Bonne chance!

Q1. Quel sera l'affichage du programme suivant?

[pseudo]

// Déclaration d'une variable globale

var x = 2

// Définition d'une fonction qui renvoie le cube d'un nombre

fonction Cube(x)

retourner x\*x\*x

fin fonction

// Définition d'une procédure qui affiche le produit de deux nombres

procédure Produit(x,y)

afficher x\*y

fin procédure

// Programme principal

var y = 3

afficher Cube(x) // Affiche le cube de x

Produit(x,y) // Affiche le produit de x et y

x = y // Affecte la valeur de y à x

afficher Cube(x) // Affiche le cube de x

a) 8 6 27

b) 8 6 8

c) 2 9 27

d) 2 9 8

Réponse: a) 8 6 27

Feedback: La fonction Cube renvoie le cube du paramètre effectif passé lors de son appel. La procédure Produit affiche le produit des deux paramètres effectifs passés lors de son appel. Les paramètres sont passés par valeur, donc les variables x et y ne sont pas modifiées par les appels des sous-programmes. La variable x est une variable globale, donc elle est accessible depuis le programme principal et les sous-programmes. La variable y est une variable locale au programme principal, donc elle n'est pas accessible depuis les sous-programmes. L'affichage du programme est donc:

Cube(x) = Cube(2) = 222 = 8

Produit(x,y) = Produit(2,3) = afficher 23 = afficher 6

x = y = x = 3

Cube(x) = Cube(3) = 33\*3 = 27

Q2. Quel sera l'affichage du programme suivant?

[pseudo]

// Déclaration d'une variable globale

var a = "Bonjour"

// Définition d'une fonction qui renvoie la longueur d'une chaîne de caractères

fonction Longueur(s)

var n = 0 // Variable locale qui compte le nombre de caractères

tant que s[n] existe // Tant qu'on n'a pas atteint la fin de la chaîne

n = n + 1 // On incrémente le compteur

fin tant que

retourner n // On renvoie le compteur

fin fonction

// Définition d'une procédure qui affiche la concaténation de deux chaînes de caractères

procédure Concaténer(s,t)

afficher s + t // On affiche la somme des deux chaînes

fin procédure

// Programme principal

var b = "le monde"

afficher Longueur(a) // Affiche la longueur de a

Concaténer(a,b) // Affiche la concaténation de a et b

a = b // Affecte la valeur de b à a

afficher Longueur(a) // Affiche la longueur de a

a) 7 Bonjourle monde 7

b) 7 Bonjourle monde 8

c) 7 Bonjour le monde 7

d) 7 Bonjour le monde 8

Réponse: b) 7 Bonjourle monde 8

Feedback: La fonction Longueur renvoie le nombre de caractères d'une chaîne de caractères passée en paramètre. La procédure Concaténer affiche la somme des deux chaînes de caractères passées en paramètre. Les paramètres sont passés par valeur, donc les variables a et b ne sont pas modifiées par les appels des sous-programmes. La variable a est une variable globale, donc elle est accessible depuis le programme principal et les sous-programmes. La variable b est une variable locale au programme principal, donc elle n'est pas accessible depuis les sous-programmes. L'affichage du programme est donc:

Longueur(a) = Longueur("Bonjour") = 7 (car il y a 7 caractères dans "Bonjour")

Concaténer(a,b) = Concaténer("Bonjour","le monde") = afficher "Bonjour" + "le monde" = afficher "Bonjourle monde" (sans espace entre les deux chaînes)

a = b = a = "le monde"

Longueur(a) = Longueur("le monde") = 8 (car il y a 8 caractères dans "le monde")

Q3. Quel sera l'affichage du programme suivant?

[pseudo]

// Déclaration d'une variable globale

var k = 1

// Définition d'une fonction qui renvoie la somme des n premiers entiers naturels

fonction Somme(n)

var s = 0 // Variable locale qui accumule la somme

pour i allant de 1 à n // On parcourt les entiers de 1 à n

s = s + i // On ajoute i à la somme

fin pour

retourner s // On renvoie la somme

fin fonction

// Définition d'une procédure qui affiche la factorielle d'un nombre

procédure Factorielle(n)

var f = 1 // Variable locale qui calcule la factorielle

pour i allant de 1 à n // On parcourt les entiers de 1 à n

f = f \* i // On multiplie f par i

fin pour

afficher f // On affiche la factorielle

fin procédure

// Programme principal

var m = 4

afficher Somme(k) // Affiche la somme des k premiers entiers naturels

Factorielle(m) // Affiche la factorielle de m

k = m // Affecte la valeur de m à k

afficher Somme(k) // Affiche la somme des k premiers entiers naturels

a) 1 24 10

b) 1 24 1

c) 4 24 10

d) 4 24 4

Réponse: a) 1 24 10

Feedback: La fonction Somme renvoie la somme des n premiers entiers naturels passés en paramètre. La procédure Factorielle affiche la factorielle du nombre passé en paramètre. Les paramètres sont passés par valeur, donc les variables k et m ne sont pas modifiées par les appels des sous-programmes. La variable k est une variable globale, donc elle est accessible depuis le programme principal et les sous-programmes. La variable m est une variable locale au programme principal, donc elle n'est pas accessible depuis les sous-programmes. L'affichage du programme est donc:

Somme(k) = Somme(1) = 1 (car la somme des entiers de 1 à 1 est égale à 1)

Factorielle(m) = Factorielle(4) = afficher 123\*4 = afficher 24 (car la factorielle de 4 est égale à 24)

k = m = k = 4

Somme(k) = Somme(4) = 10 (car la somme des entiers de 1 à 4 est égale à 10)

Q4. Quel sera l'affichage du programme suivant?

[pseudo]

// Déclaration d'une variable globale

var t = [2,5,7,9] // Tableau de taille 4

// Définition d'une fonction qui renvoie le maximum d'un tableau

fonction Maximum(t)

var max = t[0] // Variable locale qui contient le maximum provisoire du tableau

pour i allant de 1 à taille(t)-1 // On parcourt le tableau à partir du deuxième élément

si t[i] > max alors // Si on trouve un élément plus grand que le maximum provisoire

max = t[i] // On met à jour le maximum provisoire avec cet élément

fin si

fin pour

retourner max // On renvoie le maximum définitif du tableau

fin fonction

// Définition d'une procédure qui affiche le minimum d'un tableau

procédure Minimum(t)

var min = t[0] // Variable locale qui contient le minimum provisoire du tableau

pour i allant de 1 à taille(t)-1 // On parcourt le tableau à partir du deuxième élément

si t[i] < min alors // Si on trouve un élément plus petit que le minimum provisoire

min = t[i] // On met à jour le minimum provisoire avec cet élément

fin si

fin pour

afficher min // On affiche le minimum définitif du tableau

fin procédure

// Programme principal

var u = [3,6,8,4] // Tableau de taille 4

afficher Maximum(t) // Affiche le maximum du tableau t

Minimum(u) // Affiche le