Contrôle Continu 2 - UE47

14 décembre 2012

1 Suites arithmétiques et géométriques (6pts)

On considère des suites finies $(a_i)_{1 \leqslant i \leqslant n} = a_1, a_2, \ldots, a_n$. $(a_i)_{1 \leqslant i \leqslant n}$ est dite arithmétique de raison c si $\forall\ 1 \leqslant i \leqslant n-1$, $a_{i+1} - a_i = c$. $(a_i)_{1 \leqslant i \leqslant n}$ est dite géométrique de raison c si $\forall\ 1 \leqslant i \leqslant n-1$, $\frac{a_{i+1}}{a_i} = c$.

1.1 (2,5pts)

Écrire une fonction *arithm* qui prend comme argument un entier naturel n, un entier naturel i, ainsi qu'un paramètre par référence r et qui renvoie True si la suite de valeurs de la i-ième ligne, de la colonne 1 à la colonne n, est arithmétique, et donne à r la valeur de la raison, et renvoie False sinon.

1.2 (1pt)

Indiquer sans recopier toute votre précédente fonction le morceau de code à changer pour obtenir une fonction *geom* qui prend comme argument un entier naturel n, un entier naturel i, ainsi qu'un paramètre par référence r et qui renvoie True si la suite de valeurs de la i-ième ligne, de la colonne 1 à la colonne n, est arithmétique, et donne à r la valeur de la raison, et renvoie False sinon. Dans la suite, on pourra faire appel à la fonction *geom* comme si elle avait été écrite.

1.3 (2,5pts)

On suppose que des suites de taille n sont placées sur les m premières lignes de la feuille Excel. Écrire une procédure qui prend n et m en arguments et qui parcourt une à une les m lignes, et qui affiche pour chacune d'elle un message d'un des trois types suivants

- "la suite sur la ligne i est arithmétique de raison r"
- "la suite sur la ligne i est géométrique de raison r"
- "la suite sur la ligne i n'est ni arithmétique, ni géométrique"

2 Multiplication de nombres binaires (7pts)

On souhaite écrire une macro qui affiche la multiplication de deux nombres entiers positifs binaires. Dans la suite, on suppose qu'on a déclaré une variable globale n et qu'on lui a assigné une valeur quelconque. Les deux nombres seront rentrés sur les deux premières lignes de la Feuille Excel de la colonne $n+1$ à la colonne $2n$ (un par ligne). Le résultat sera contenu sur une ligne i de la colonne 1 à la colonne $2n$. De la ligne 3 à la ligne $i-1$, on trouve les nombres à sommer pour trouver le produit. Voici un exemple d'affichage avec la multiplication des nombres binaires 1011 et 110 (avec n=4 et i=6).

				1	0	1	1
			x		1	1	0
					1	1	0
+				1	1	0	0
+		1	1	0	0	0	0
	1	0	0	0	0	1	0

2.1 (1pt)

Écrire une fonction *compte1* sans aucun paramètre qui renvoie le nombre de 1 de l'entier naturel sur la première ligne. Ce nombre correspond au nombre de termes à sommer (et avec nos notations à $i-3$).

2.2 (2,5pts)

Écrire une macro *termesSommes* qui affiche sur les lignes 3 à $i-1$, les nombres à sommer pour calculer la multiplication. Sur chacune de ces lignes on retrouve le nombre de la deuxième ligne suivi d'une certain nombre de 0, dépendant de la position dans le premier nombre du "1" qu'on multiplie au deuxième nombre.

2.3 (2,5pts)

Écrire une procédure *somme* qui prend un entier i en paramètre et qui fait la somme des nombres binaires écrits de la ligne 3 à la ligne $i-1$ (de la colonne 1 à la colonne $2n$). Le résultat sera inscrit sur le i-ième ligne de la colonne 1 à la colonne $2n$.

2.4 (1pt)

Écrire une macro *multiplication* qui utilise la fonction et les procédures définies précédemment et réalise l'affichage de la multiplication décrit en introduction.

3 Stocks (7pts)

Une entreprise entrepose n types de produit notés $p_1, p_2, \ldots, p_n$ en m points de stockage notés $s_1, s_2, \ldots, s_m$. On souhaite simuler des opérations sur ces stocks. Pour cela, on va travailler sur un tableau à 2 dimensions défini par `Dim stock(1 To n, 1 To m) As Integer`, où n et m sont des variables globales entières supposées déjà définies. *stock* représente l'état des stocks, de sorte qu'en i-ième ligne, j-ième colonne (`stock(i,j)`) on trouve le nombre de produits p_i au point de stockage s_j. Une commande sera représentée par un tableau monodimensionnel à n cases (on supposera que ses indices vont de 1 à n) tel que le nombre de produits p_i commandés est contenu dans la i-ième case du tableau.

3.1 (1pt)

Écrire une fonction *commandeStock(com As Variant,j As Integer) As Boolean* qui dit si le point de stockage s_j peut assurer la commande *com*.

3.2 (1pt)

Écrire une fonction *commande(com As Variant) As Integer* qui renvoie l'indice du premier point de stockage à pouvoir assurer la commande *com*. Cette fonction devra retourner la valeur -1 si aucun point de stockage ne peut assurer la commande.

3.3 (1pt)

Écrire une fonction *transfert(com As Variant,j1 As Integer, j2 As Integer) As Boolean* qui transfert la commande *com* du stock s_{j1} au stock s_{j2} si c'est possible et renvoie True. Si ce n'est pas possible, rien n'est transféré et la fonction renvoie False.

3.4 (1pt)

Écrire une fonction *maxProduit(i As Integer) As Integer* qui renvoie le nombre de produits de type p_i du stock contenant le plus grand nombre de produits de ce type.

3.5 (1pt)

Écrire une fonction *épuisé(j As Integer) As Integer* qui renvoie l'indice du premier type de produit épuisé dans le stock s_j, et renvoie -1 si aucun type de produit n'est épuisé.

3.6 (2pts)

Écrire une fonction *réapprovisionner(j As Integer) As Boolean* qui réapprovisionne le stock s_j de la façon suivante : tant qu'un type de produit p_i y est épuisé, la moitié des produits p_i (-1, en cas de nombre impair) d'un autre stock sont passés au stock s_j. La fonction renvoie True si le réapprovisionnement a pu s'achever convenablement et False sinon.