S3

Module: I132
Date : 13/02/2016
Heure : 16H00-18H00
Locaux: Forum, Amphi A, B et E
Resp. : B.Ouchao & F.Amounas

T[2] → x[10]

Examen : Algorithmique et Programmation II

Durée: 2H

Exercice 1.

1. Donner la déclaration d'un pointeur sur un tableau de 12 chaînes de caractères.
2. Donner la déclaration d'un tableau dont chacun de ses 15 éléments est une structure à trois champs : un entier noté *nombre*, un pointeur d'entier *ptrint* et un tableau de 15 caractères. On suppose que le tableau soit initialisé.

   Donnez l'expression permettant de placer dans le champ ptrint du 3ème élément du tableau, l'adresse du champ nombre du 11ème élément.
3. Que calcule cet algorithme ?

```
Algorithme cal;
Variable i,j,r:entier
debut
 lire(i,j);
r←1
tantque (j≠0) faire
   si (j mod 2=0) alors
        i←i*i ;
j←j/2 ;
  sinon
r←r*i ;
j←j-1;
  finsi
fintantque
ecrire(r)
fin
```

~~Alg puissance~~

Alg : puissance
Variables i, j, r : entier
Si (j=0)
retourne 1
Sinon
r ← 1
Pour ...

Proposer un autre algorithme plus simple pour parvenir au même résultat.

4. Indiquer le rôle de la fonction g:

```
void g ( char *t1, char *t2) {
  char *pt1,*pt2 ;
    pt1=t1;
    pt2=t2;
    while (*pt2 != '\0') {
       *pt1 = *pt2 ;
        pt1++ ;
        pt2++ ;
     }
   *pt1 = '\0' ;
 }
```

Récursive ; condition d'arrêt
possède

tableau t[15]
t[2] -> ptrint = &t[10]

Exercice 2.

Définir une fonction « ***suite_de_zero*** » qui prend en paramètre un vecteur V de *nb* éléments et qui retourne la position i tel que V[i] est le début de la plus longue suite de zéros dans V.

Exemple :

V :

'1'	'0'	'1'	'0'	'0'	'0'	'1'	'1'	'0'	'0'	'0'	'0'	'0'	'1'	'1'	'0'	'1'	'0'	'0'

→ La plus longue suite de zéros se trouve à la position 8.

<u>NB</u> : S'il y a plusieurs suites de zéros de longueur maximale, la fonction retournera la position de la première suite rencontrée.

Exercice 3.

Définir une fonction « *VerifieMatrice* » qui détermine si une matrice carrée d'ordre N contient une ligne et une colonne identiques. La fonction rend 3 résultats : Vrai ou Faux, l'indice d'une ligne et l'indice de sa colonne identique. Ces derniers indices contiendront -1 s'il n'existe aucune ligne et colonne identiques.

Exemple:

W	K	R	U
A	P	E	H
W	A	W	M
M	H	U	K

Vrai, 2, 0

D	E	N	J	F	Z
Y	E	V	L	R	X
C	T	P	G	D	F
V	A	T	P	F	Q
D	Y	E	M	B	C
M	K	Q	E	H	A

Vrai, 3, 2

G	S	D	H
F	A	U	B
H	X	N	W
R	L	M	A

Faux, -1, -1

Exercice 4.

On considère un réseau social de type Twitter et N individus appartenant à ce réseau. On considère le tableau des relations entre ces personnes. Il s'agit d'un tableau bidimensionnel.

On convient que, si l'on appelle A ce tableau :

- A[i][j] = 1 si j est abonné aux messages de i ;
- A[i][j] = 0 dans le cas contraire.

On notera que ce réseau social n'est pas symétrique, c'est-à-dire que A[i][j] = 1 n'implique pas nécessairement A[j][i] = 1. On considère par ailleurs que A[i][i] = 0.

1. Écrivez une fonction « ***ratio*** » qui prend pour paramètre un tableau des relations, A de N×N et qui retourne un tableau monodimensionnel R de taille N contenant des nombres réels tel que R[i] représente le rapport entre le nombre de personnes auxquelles i est abonné divisé par le nombre de personnes qui sont abonnées à i.

A[i][j] = 1

2. Par la suite, on cherche à établir le niveau de liaison entre les individus.

- Si j est abonné à i, on dit que le niveau de liaison entre i et j est de 1.
- Si j n'est pas abonné à i mais qu'il existe une personne k telle que k est abonné à i et que j est abonné à k, le niveau de liaison de i à j est 2.

Plus généralement, considérons toutes les chaînes d'abonnement entre i et j: $(i, a_1, a_2, \ldots, a_n, j)$ a_1 est abonné à i, a_2 à a_1, a_{k+1} à a_k, …, j à a_n. Soit m(i,j) le nombre minimal de personnes dans toutes les chaînes d'abonnement liant i et j. Le niveau de liaison entre i et j est m(i,j) +1.

- Si i et j ne peuvent pas être reliés par une chaîne d'abonnement, on conviendra, dans ce cas, que le niveau de liaison entre i et j est 0.

Nous allons fabriquer par itération successive le tableau des niveaux de liaison. Une itération consiste à appeler une fonction « *calculer_niveau* ». Cette fonction a pour argument un niveau de liaison p et un tableau A (N×N) qui contient tous les niveaux de liaison de 1 à p et des 0 pour les niveaux supérieurs à p. Après l'exécution de calculer_niveau, le tableau A est complété par le niveau de liaison p+1.

Exemple :

0	1	0	0
0	0	1	0
0	0	0	1
0	0	0	0

⇨

0	1	2	0
0	0	1	2
0	0	0	1
0	0	0	0

1→ 0 (c.-à-d. 1 est abonné à 0)
2→ 1
3→ 2

Après exécution de la fonction *calculer_niveau* pour p=1.

Écrivez la fonction « *calculer_niveau* ».

Bonne Chance