

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
UNIVERSITE M'Hamed BOUGARA de BOUMERDES. FACULTE DES SCIENCES, DEPARTEMENT D'INFORMATIQUE

Matr Nom Prenom Gr... V1

Rattrapage, Durée : 1h

Module : Algo. & Structures de Données 1

Barème approximatif (10+10)

Filière: 1^{ière} Ing.Info- S1-2023/2024, Date : 08/06/2024

Vous n'avez droit qu'à un seul sujet. Calculatrice interdite. L'effaceur aussi, sinon **-2 pts**. Si erreur, barrer proprement.

Exercice 1 (10pts) Les réponses de cet exercice suppose le langage C avec headers <stdio.h> et <string.h> inclus.

Afin de représenter les informations d'une section d'étudiants composée de 10 groupes exactement, on utilise 3 matrices **matNom**, **matMoy** et **matRate**. Les éléments de ces matrices sont des chaînes de caractères, des réels et des caractères, respectivement. Les éléments d'indice (i,j) de ces matrices sont associés à un même étudiant, et les colonnes j sont associés à un même groupe de cette section. Les informations mises dans ces matrices sont respectivement, le nom, la moyenne et la mention qui est un caractère parmi {'A', 'B', 'C', 'D'}. Un groupe comporte 20 à 25 étudiants. L'identification de la fin de chaque groupe est possible grâce à la mise de la constante "fin", -1 ou 'X', selon le type de la matrice, à la position qui suit immédiatement le dernier étudiant du groupe. Un nom d'étudiant ne peut dépasser 15 caractères alphabétiques.

Pour avoir le point, il faut cocher la (ou les) bonne(s) réponse(s) uniquement, s'il y en a (1pt x 10).

1. Dans la déclaration, on peut utiliser entre autres:

- ☐ `char matNom[25][10];`
- ☐ `float matMoy[10][25];`
- ☐ `char matRate[24][9];`
- ☐ `char matNom[25][10][15];`

2. `matNom[5][0]` :

- ☐ est un caractère
- ☒ est un tableau unidimensionnel
- ☒ est une chaîne de caractères
- ☐ n'existe pas

3. On peut utiliser ceci sans problème:

- ☐ `matNom[20][3]="fin";`
- ☐ `strcpy(matNom[20][3],"fin");`
- ☐ `strlen(matNom[10][10]);`
- ☒ `strcmp(matNom[10][9],"Malik");`

4. Si j est un entier < 20 alors `matMoy[j]` :

- ☐ est une matrice
- ☒ est une ligne d'une matrice
- ☐ est une colonne d'une matrice
- ☐ contient les moyennes du (j+1)-ième groupe

5. `matMoy[0][0]` :

- ☐ est une matrice
- ☒ est un réel
- ☐ est un tableau de réels
- ☐ est un tableau bidimensionnel

6. `matRate[5]`:

- ☐ est un caractère
- ☒ est un tableau unidimensionnel
- ☐ est une chaîne de caractères
- ☒ peut contenir les caractères 'A', répétés 10 fois.

7. On peut écrire sans aucun problème:

- ☐ `printf("%s",matNom[25][10]);`
- ☒ `scanf("%s",matNom[0][1]);`
- ☐ `printf("%c",&matNom[25][10]);`
- ☐ `scanf("%c",&matNom[0][1]);`

8. `matNom[5][0][3]` est :

- ☒ un caractère
- ☐ un tableau unidimensionnel
- ☐ une chaîne de caractères
- ☐ une écriture fausse

9. On peut utiliser ceci sans problème:

- ☒ `matNom[5][0][3]='a' ;`
- ☐ `matNom[5][0][0]="A" ;`
- ☒ `matNom[5][0][3]='c' ;`
- ☐ aucune de ces réponses

10. `matRate` est :

- ☐ un caractère
- ☒ un tableau de tableau de caractères
- ☒ une matrice de caractères
- ☐ une lettre alphabétique

Cocher 1 (ou +ieurs) case fausse → 0/1pt. Cocher 1 parmi 2 ou 2 parmi 3 justes → 0.5/1pt. Cocher 1 parmi 3 justes → 0.25/1pt

Exercice 2 (10pts) : **mat** est une matrice quelconque d'entiers (10 lignes et 10 colonnes max.). On appelle antidiagonale de **mat** toute séquence complète d'éléments parallèle ou confondue avec l'antidiagonale principale (voir exemple).

En supposant **mat** déjà remplie, écrire un algorithme ou un programme C qui inverse l'ordre de l'antidiagonale ayant la plus petite somme (s'il y en a plusieurs, inverser une seule suffit). L'affichage de la matrice résultat n'est pas demandé.

Exemple : une matrice 3 x 4 possède 6 antidiagonales. Les cases en gris sont des exemples de celles-ci.

1) La principale

2	-1	5	9
6	7	21	-7
13	5	6	6

2) une antidiagonale

2	-1	5	9
6	7	21	-7
13	5	6	6

3) encore une autre

2	-1	5	9
6	7	21	-7
13	5	6	6

4) la plus petite

2	-1	5	9
6	7	21	-7
13	5	6	6

5) le résultat est :

2	-1	5	9
6	7	21	6
13	5	-7	6

Remarque : Pour le C, il est interdit d'utiliser des headers autres que <stdio.h>

Réponse

```
int inversMinAntiDiag(int mat[][10], int n,
int m){
int iMinAnti, jMinAnti, iAstart, jAstart,
antiMinSum, i1, i2, j1, j2, iALast, jALast,
sum, i, j, tmp;
//Determiner la min som antidiag ---> 5pts
iAstart=0;
iMinAnti=iAstart;jMinAnti=0;
antiMinSum=mat[0][0];
for(jAstart=1;jAstart<m;jAstart++){
sum=0;
for(i=iAstart,j=jAstart;
(i<n)&&(j>-1);i++,j--){
sum+=mat[i][j];
}
if(sum<antiMinSum){
iMinAnti=iAstart;
jMinAnti=jAstart;
antiMinSum=sum;
}
}
jAstart=m-1;
for(iAstart=1;iAstart<n;iAstart++){
sum=0;
for(i=iAstart,j=jAstart;
(i<n)&&(j>-1);i++,j--){
sum+=mat[i][j];
}
if(sum<antiMinSum){
iMinAnti=iAstart;
jMinAnti=jAstart;
antiMinSum=sum;
}
}
printf("\n Min antidiag starts at i=%d
j=%d",iMinAnti,jMinAnti);//Enplus
//Inverser la min som antidiag ---> 5pts
iALast=jMinAnti; jALast=iMinAnti;
if(iALast>n-1){
jALast+=iALast-n+1; iALast=n-1;
}else if(jALast>0){
iALast+=jALast; jALast=0;
}
for(i1=iMinAnti,j1=jMinAnti,i2=iALast,
j2=jALast;j2<j1; i1++,j1--,i2--,j2++){
tmp=mat[i1][j1];
mat[i1][j1]=mat[i2][j2];
mat[i2][j2]=tmp;
}
return 0;
}
```

/*Il n'est pas demandé d'utiliser les actions paramétrées
La partie déclaration n'est pas comptabilisée */