

TP 12

TP de synthèse (procédures, fonctions, tableaux, structures, fichiers)

Ce sujet porte sur la manipulation de deux tableaux de données : un tableau 1D de couples *planète/diamètre*, et une matrice carrée (tableau 2D) de distances entre ces planètes. Le schéma qui suit montre un exemple avec trois planètes.

Planètes		Distances				
		0	1	2		
0	Mercure 4 900	0	50	92		
1	Vénus 12 000	50	0	42		
2	Terre 12 900	92	42	0		

Par exemple, la distance entre *Mercure* (planète d'indice 0) et la *Terre* (planète d'indice 2) est de 92 millions de km.

le diamètre est donné en km, les distances en millions de km

Le tableau des distances est symétrique, donc $td[i][j]$ est égal à $td[j][i]$, et bien sûr, $td[i][i]$ est égal à 0.

Définition des structures de données utilisées :

```
#define MAX 10

typedef char chaine20[21];

typedef struct{
    chaine20 nomPlanete;
    int diametre;
} tPlanete;

typedef tPlanete tabPlanetes[MAX];

typedef int tabDistances[MAX][MAX];
```

Remarque importante

Vous devrez tester dans un main toutes les procédures et fonctions demandées ci-dessous.

Question 1

Écrivez une procédure `void initDistances(tabDistances td)` qui initialise tous les éléments du tableau `td` à 0.

Question 2

Écrivez une procédure `void remplirPlanetes(tabPlanetes tp, int * nbPlanetes)` qui remplit le tableau `tp` à l'aide du fichier binaire "Planetes.data" accessible sur Moodle. La procédure fournit en sortie le nombre de planètes ajoutées au tableau (paramètre `nbPlanetes`).

Question 3

Écrivez une procédure

```
void remplirDistances(tabPlanetes tp, tabDistances td, int nbPlanetes)
```

qui lit au clavier et enregistre dans `td` la distance entre tous les couples de planètes du tableau `tp`.

Exemple avec 4 planètes (en gras les saisies de l'utilisateur) :

```
Distance entre Mercure et Venus : 50
Distance entre Mercure et Terre : 92
Distance entre Mercure et Mars: 170
Distance entre Venus et Terre : 42
Distance entre Venus et Mars : 120
Distance entre Terre et Mars : 78
```

Question 4

Écrivez une procédure `void afficherPlanetes(tabPlanetes tp, int nbPlanetes)` qui affiche les éléments du tableau `tp` comme suit :

Mercure	4900	km de diametre
Venus	12000	km de diametre
Terre	12800	km de diametre
Mars	6800	km de diametre

Question 5

Écrivez une procédure :

```
void afficherDistances(tabDistances td, tabPlanetes tp, int nbPlanetes)
```

qui affiche les éléments du tableau `td` comme ci-dessous :

	Mercure	Venus	Terre	Mars
Mercure	0	50	92	170
Venus	50	0	42	120
Terre	92	42	0	78
Mars	170	120	78	0

Question 6

Écrivez une fonction `int plusGrandeDistance(tabDistances td, int nbPlanetes)` qui retourne la plus grande valeur du tableau `td`.

Question 7

Écrivez une fonction

```
int recherchePlanete(tabPlanetes tp, chaine20 nom, int nbPlanetes)
```

qui retourne l'indice, dans `tp`, de la ville dont le nom est passé en paramètre, ou bien -1 si la ville n'est pas dans `tp`.

Question 8

Écrivez une fonction `void distanceMoyenne(tabDistances td, tabPlanetes tp, int nbPlanetes)` qui lit au clavier le nom d'une planète et qui affiche sa distance moyenne avec les autres planètes. Sur l'exemple avec 4 planètes, si l'utilisateur saisit Venus la procédure devra afficher :

Distance moyenne de Venus avec les 3 autres planètes : 70,67 millions de km

Si le nom de planète saisi est absent du tableau `tp`, la procédure doit signaler l'erreur et redemander la saisie.