Lycée TOUCHARD WASHINGTON

Langage C

TD n°5

SNir 1
2021-2022

Les tableaux

1- Gestion des notes

L'algorithme pour la gestion des notes d'une classe étudié lors de l'exercice 1 du TD5 d'algorithmique peut se résumé ainsi

Environnement :	**En entrée :** Le clavier **En Sortie :** L'écran

Schéma algorithmique

```
Début
        Pour indice allant de 0 à NB_ELEVES – 1
                lire : notes[indice]
        FinPour

        cpt ← 0
        min ← notes[0]
        max ←notes[0]
        somme ← 0

        Pour indice allant de 0 à NB_ELEVES – 1
            écrire : « L'élève numéro », indice, « a eu la note », notes[indices], « /20 »
            Si notes[indice] ≥ 10
                Alors cpt ← cpt +1
            FinSi
            Si min > notes[indices]
                Alors min ← notes[indice]
            FinSi
            Si max < notes[indices]
                Alors max ← notes[indice]
            FinSi
            somme = somme + notes[indice]
        FinPour

        écrire : cpt, « élèves ont obtenu une notes  ≥ 10 »
        écrire : « La moyenne de la classe est : », somme / NB_ELEVES
        écrire : « La note la plus faible est : », min
        écrire : « Lanote la plus élevée est : », max

Fin
```

Lexique des variables

indice	entier
notes[NB_ELEVES]	réel
cpt	entier
min	réel
max	réel
somme	réel

Lexique des constantes

NB_ELEVES	10

1. En langage C sous NetBeans, réalisez un projet nommé **GestionNotes**. Dans le programme principal **main.c**, codez l'algorithme ci-dessus, la boucle réalisant la saisie des notes sera remplacée par une affectation des notes lors de la déclaration du tableau.
2. Modifiez le programme pour que le numéro d'élève commence à 1 et non pas à 0.

2- Algorithme de tri par minima successifs

Rappel de la méthode :

Dans un premier temps, le plus petit élément du tableau est recherché puis permuté avec la valeur de la première case. Le traitement est repris avec le reste du tableau, le plus petit élément est permuté avec la deuxième case et ainsi de suite.

1er tour		Pour indice allant de 1 à NB_ELEMENTS -1			
5	7	2	1	4	
5	7	2	1	4	Permutation
2	7	5	1	4	Permutation
1	7	5	2	4	
2ème tour		Pour indice allant de 2 à NB_ELEMENTS -1			
1	7	5	2	4	Permutation
1	5	7	2	4	Permutation
1	2	7	5	4	

Le tableau est ainsi trié au fur et à mesure. La case comparée aux autres s'incrémente à chaque tour jusqu'à la valeur NB_ELEMENT -2

4ème tour		Pour indice allant de 3 à NB_ELEMENTS -1		
1	2	4	7	5
1	2	4	5	7

Ce qui permet d'obtenir les étapes suivantes :

indices	0	1			N
Au départ	**5**	**7**	**2**	**1**	**4**
1er tour	1	7	5	2	4
2eme tour	1	2	7	5	4
3eme tour	1	2	4	7	5
4eme tour	1	2	4	5	7

1. A partir de ces explications et de l'algorithme que vous avez réalisé pour l'exercice 3 du TD5 d'algorithmique, codez dans un nouveau projet nommé **TriMinima** le programme permettant de trier un tableau d'entier avec la méthode des minima successifs. Ce tableau sera initialisé lors de la déclaration du tableau.
2. La fonction ***int rand()*** de la librairie ***stdlib.h*** retourne un entier compris entre 0 et RAND_MAX obtenu de manière pseudo aléatoire, c'est toujours la même séquence. Pour la rendre plus aléatoire, il est nécessaire d'utiliser une deuxième fonction ***srand*** comme le montre l'exemple ci-dessous :

```
int main()
{
        srand(time(NULL));      // Initialisation de la donnée seed
        printf("%d", rand());   // rand  renvoie un nombre aléatoire entre 0 et RAND_MAX
        return 0;
}
```

Modifiez votre programme pour remplir le tableau de 20 cases avec des valeurs aléatoires comprise entre 0 et 100