

Programmation 2

L1 2011-2012

Travaux Pratiques 4 - Tris naïfs et tableaux

Nous allons dans ce tp écrire différentes fonctions pour trier des tableaux d'entiers.

Pour pouvoir tester ces fonctions on remplira des tableaux avec le générateur aléatoire standard du C. La fonction standard `long int random(void)` renvoie une valeur entière positive pseudo-aléatoire à chaque appel. Pour pouvoir l'utiliser on doit inclure le fichier `<stdlib.h>`. La ligne `#include<stdlib.h>` doit être présente avant le premier appel.

Exercice 1.

1. Écrire et exécuter le programme suivant :

```
#include<stdlib.h>
#include<stdio.h>
#define MAX 10000

int main(void){
    int i,val;
    for(i=0;i<10;i++){
        val=random()%MAX;
        printf("%d \n",val);
    }
    return 0;
}
```

2. Que fait le programme ?
3. Exécuter plusieurs fois le programme. Les valeurs affichées ont-elles changés ?

Pour pouvoir changer la suite des valeurs renvoyées par `random` on utilise la fonction `void srand(unsigned int seed)`. La suite des valeurs renvoyées par `random` dépend entièrement du paramètre `seed` (*graine*) de `srand`.

Pour avoir une suite différente à chaque exécution, il faut fournir une valeur différente à la fonction `srand()`.

On peut utiliser dans ce but la fonction `time_t time(time_t *t)` qui fournit le temps écoulé en secondes depuis le 1er janvier 1970. Penser à

inclure le fichier `<time.h>` avant toute utilisation de `time`. On peut appeler cette fonction sous la forme :

```
graine=time(NULL);
```

Exercice 2. Générateur

Écrire une fonction `void RemplirTableau(int tableau[],int taille,int intervalle)` qui remplit le tableau `tableau` avec des valeurs pseudo-aléatoires comprises entre 0 et `intervalle`.

Exercice 3.

Écrire une fonction `int Minimum(int tableau[],int taille)` qui renvoie l'indice de la case du tableau qui contient la plus petite valeur.

Exercice 4. Implémentation des algorithmes de tris

1. Ecrire les fonctions permettant le tri d'un tableau d'entiers. Chaque fonction doit prendre en paramètre un tableau `int tableau[]` et sa taille `int taille`.
 - (a) tri à bulles
 - (b) tri par sélection
 - (c) tri par insertion
2. Rajouter deux paramètres à vos fonctions pour calculer
 - Le nombre de comparaison
 - Le nombre de déplacement
3. Comparez avec la valeur théorique.

Exercice 5.

Écrire une fonction récursive `int rechercherDich(int taille,int tableau[],int valeur)` qui renvoie l'indice de la case de `tableau` qui contient `valeur`. On suppose que le tableau est trié. Si la valeur ne se trouve pas dans le tableau la fonction doit renvoyer -1.