

Cours de programmation séquentielle

Tableaux unidimensionnels

1 Buts

- Utilisation de tableaux unidimensionnels.
- Utilisation de fonctions.
- Utilisation de librairie externe.
- Compilation avec plusieurs fichiers.

2 Énoncé

Une façon de déclarer un tableau d'entiers `tab` est la suivante:

```
int32_t size = 10;  
int32_t tab[size];
```

Écrire un programme qui réalise les tâches suivantes.

0. Écrire un `Makefile` pour compiler votre projet.
1. Demander à l'utilisateur d'entrer la valeur de `size` pour la création du tableau.
2. Remplir le tableau `tab` de valeurs aléatoires plus petites qu'un entier `val_max` qui est beaucoup plus petit que `size` (disons 10 à 100 fois plus petit).
3. Trouver le plus petit élément de `tab`.
4. Placer le plus grand élément de `tab` en fin de tableau en procédant à un échange de place.
5. Rechercher dans `tab` un élément entré au clavier par l'utilisateur et retourner l'indice de la première occurrence.
6. Calculer la moyenne des éléments de `tab`.
7. Calculer la variance des éléments de `tab`. La variance `var` des éléments de `tab` est définie par la formule:

$$\text{var} = \frac{1}{\text{size}} \sum_{i=0}^{\text{size}-1} (\text{tab}[i] - \text{tab}_m)^2, \quad (1)$$

où tab_m désigne la moyenne des éléments de `tab` et `tab[i]` est le i -ème élément de `tab`.

8. Trier les éléments de `tab` par ordre croissant.
9. Trouver l'élément médian du tableau `tab`. Après avoir trié le tableau `tab`, l'élément médian est défini comme étant la valeur:
 - `tab[(size-1)/2]` si `size` est impair

— $(\text{tab}[(\text{size}-1)/2] + \text{tab}[\text{size}/2])/2.0$ si `size` est pair.

Exemple:

TABLE 1: L'élément médian de ce tableau vaut: 3.

1	5	8	2	6	5	3	1	0	3
---	---	---	---	---	---	---	---	---	---

10. Écrire une fonction pour chacun des points précédents.
11. Vérifier que le générateur de nombres aléatoires `rand()` est équitable. Pour cela, utiliser un tableau d'entiers `histo` pour comptabiliser le nombre de fois qu'une valeur est tirée. À noter que la valeur tirée correspond à l'indice du tableau et que c'est son contenu qui est incrémenté de 1. Utiliser les notions précédentes (moyenne, écart-type, minimum/maximum) pour dire si le générateur est équitable.
12. Visualiser le contenu du tableau `histo` en utilisant la bibliothèque SDL 2.0 (<https://wiki.libsdl.org/FrontPage>). Un exemple d'utilisation de la librairie vous est fourni sur Cyberlearn (`gfx_example.tar`). Vous pouvez également télécharger ce fichier en cliquant sur [ce lien](#).

2.1 Exercice supplémentaire:

Étendre les questions “11” et “12” à un tableau en deux dimensions d'entiers, `histo_2d`. Pour ce faire il s'agira de définir un tableau de tableaux,

```
int32_t height = 200;
int32_t width = 200;
int32_t histo_2d[width][height];
```

de générer des couples de nombres aléatoires, `x` et `y`, entre `0`, `height-1` et `0`, `width-1` à l'aide de la fonction `rand()`, et pour chaque tirage de `x`, `y`, incrémenter de un la valeur du tableau `histo_2d[x][y]`. Finalement, afficher le tableau en niveaux de gris à l'aide de la librairie SDL2 pour vérifier si le générateur `rand()` ne possède pas de corrélation en deux dimensions (s'il produit des paires de nombres homogènes dans l'espace).