



Ecole Polytechnique Sousse
Département Informatique
SECTION : GÉNIE INFORMATIQUE, NIVEAU : 3^{ème} ANNÉE
AU : 2019-2020

Programmation C

Travaux Pratiques N°3

Exercice 1

1. Tri par sélection (par minimum)

Principe :

Le principe de cette méthode est simple. Elle consiste à :

- Chercher l'indice du plus petit élément du tableau T13[0..N13] et permuter l'élément correspondant avec l'élément d'indice 0 ;
- Chercher l'indice du plus petit élément du tableau T13[1..N13] et permuter l'élément correspondant avec l'élément d'indice 1 ;
-
- Chercher l'indice du plus petit élément du tableau T13[N13-2..N13-1] et permuter l'élément correspondant avec l'élément d'indice N13-2 ;

Exemple :

Tableau initial	60	50	20	40	10	30
Après la 1 ^{ère} itération	10	50	20	40	60	30
Après la 2 ^{ème} itération	10	20	50	40	60	30
Après la 3 ^{ème} itération	10	20	30	40	60	50
Après la 4 ^{ème} itération	10	20	30	40	60	50
Après la 5 ^{ème} itération	10	20	30	40	50	60

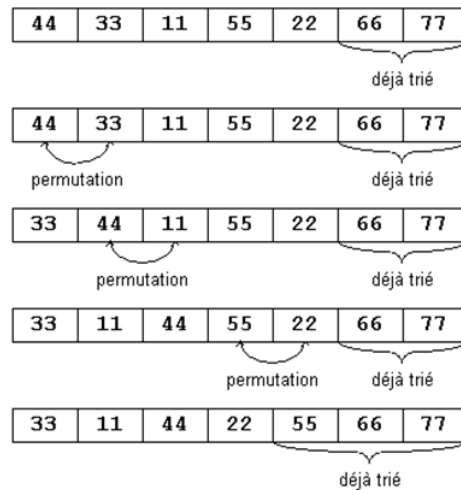
Écrire une fonction TRI_SELECTION à deux paramètres T13, N13 permettant de trier le tableau T13 par ordre croissant.

2. Tri à bulles

Principe :

En recommençant chaque fois au début du tableau, on effectue à plusieurs reprises le traitement suivant : On propage, par permutations successives, le plus grand élément du tableau vers la fin du tableau (comme une bulle qui remonte à la surface d'un liquide).

Exemple :



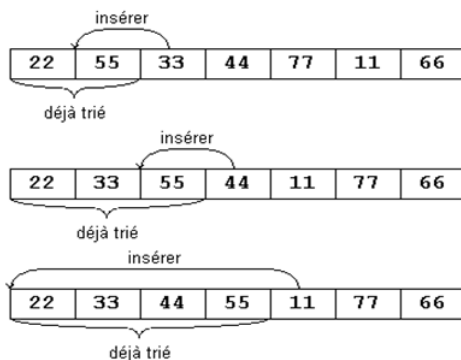
Écrire une fonction TRI_BULLES à deux paramètres T14, N14 permettant de trier le tableau T14 par ordre croissant.

3. Tri par insertion

Principe :

Trier le tableau de gauche à droite en insérant à chaque fois l'élément I+1 dans le tableau (déjà trié) des I premiers éléments.

Exemple :



Écrire une fonction TRI_INSERTION à deux paramètres T15, N15 permettant de trier le tableau T15 par ordre croissant.

4. Tri par dénombrement

Principe :

Le tri par dénombrement prend comme hypothèse que l'on trie des entiers et que l'on connaît l'intervalle $1..k$ ($k > 0$) dans lequel sont choisis les n entiers à trier. Certaines valeurs peuvent figurer plusieurs fois dans la liste des données à trier ; ces valeurs figureront alors avec le même nombre d'occurrences après le tri.

Le principe de ce tri est de compter pour chaque entier compris entre 1 et k, son nombre d'occurrences parmi les entiers à trier, puis on en déduit le résultat du tri.

Écrire les fonctions suivantes que vous jugez nécessaires pour trier le tableau T16 de taille N16 par ordre croissant.

Exemple :

Pour $k = 8$ et $N16 = 9$

T16

6	4	2	8	4	2	3	6	4
---	---	---	---	---	---	---	---	---

On crée le tableau E de nombres d'occurrences associés à chaque indice.

E

0	2	1	3	0	2	0	1
---	---	---	---	---	---	---	---

Le tableau final trié est :

T16

2	2	3	4	4	4	6	6	8
---	---	---	---	---	---	---	---	---

5. Le but de cet exercice est de trier le tableau T17 de taille N17 dans un autre tableau B par ordre décroissant en utilisant le principe suivant :

Principe :

- chercher le maximum de T17
- placer ce maximum dans B
- remplacer le maximum par -1 dans T17
- refaire les étapes 1, 2 et 3 jusqu'à ce que le tableau T17 soit entièrement composé de -1.

Écrire les fonctions que vous jugez nécessaires pour résoudre ce tri.

6. Tri particulier

Le but de ce problème est d'écrire un programme C qui effectue le tri d'un tableau T18 d'entiers en mettant en premier les entiers positifs (ou nuls) dans l'ordre décroissant, puis ensuite les entiers négatifs mais dans l'ordre croissant.

Écrire une fonction TRI_PARTICULIER permettant de trier T18 de taille N18 en appliquant le principe ci-dessus.

Exemple :

Pour $N=8$

T

-4	3	-6	1	5	0	-2	6
----	---	----	---	---	---	----	---

Après tri

T

6	5	3	1	0	-6	-4	-2
---	---	---	---	---	----	----	----

Exercice 2

1. Recherche séquentielle :

Écrire Une fonction RECH_SEQUENTIELLE à 2 paramètres T19, N19 permettant de (d') :

- Saisir la valeur val,
- Chercher la première position de la valeur val dans le tableau T19.

2. Recherche dichotomique :

Principe :

Le principe est de décomposer le tableau T20 en deux sous tableaux. Trois cas peuvent se produire :

- Si $val = T20[milieu]$ alors val est trouvé et la recherche est terminée.
- Si $val < T20[milieu]$ alors on va chercher val dans la partie gauche du tableau T20.
- Si $val > T20[milieu]$ alors on va chercher val dans la partie droite du tableau 20.

On poursuit la recherche tant que $T20[milieu]$ est différent de val est tant que la dimension de sous tableau reste valide.

Écrire une fonction RECH_DICHOTOMIQUE permettant de chercher la première position de la valeur "val" (avec "val" saisie à partir du clavier) dans le tableau T20 de éléments triés par ordre croissant

N.B :

- T13 contient des entiers quelconques et de taille N13.
- T14 contient des entiers quelconques et de taille N14.
- T15 contient des entiers quelconques et de taille N15.
- T16 contient des entiers avec ($1 \leq T16[i] \leq k$ et $k > 0$) et de taille N16.
- T17 contient des entiers strictement positifs et distincts et de taille N17.
- T18 contient des entiers quelconques et de taille N18.
- T19 contient des entiers quelconques et de taille N19.
- T20 contient des entiers quelconques et de taille N20.