
TD2 : Procédure et fonctions paramétrées

Questions de révision générale :

- 1) Quel avantage d'avoir des paramètres pour les fonctions et procédures ? Tirer conclusion des solutions de la série 1
- 2) Que veut dire un paramètre formel et un paramètre effectif ?
- 3) Quelle est la différence entre un paramètre passé par valeur et un paramètre passé par adresse ? Expliquer à l'aide d'exemple et donner la syntaxe C et C++.
- 4) Pourquoi les tableaux et les matrices doivent "de préférence" être passés par adresse ?

Exercice 1 : (Procédure de recherche dans un tableau)

- 1) Déclarer un tableau T d'entier dans la taille maximal est donnée par une constante Max;
- 2) Ecrire une procédure **Chercher_Tableau** permettant de chercher une valeur **V** en retournant son indice **ind** si **V** existe, et en retournant la valeur -1 si le contraire. (Au début on commence par la méthode la plus simple et intuitive : **Recherche séquentielle**).
 - Penser à résoudre la même solution avec une fonction;
- 3) Ecrire une procédure **Chercher_Tableau_Binaire** (ou une fonction) qui exploite la technique de la recherche binaire (dichotomique).
 - Proposer une solution itérative.
 - Proposer une solution récursive.

Exercice 2 : (Procédure de recherche plus sophistiquée)

La dichotomie étagée : La dichotomie étagée est l'une des techniques de recherche dans les tableaux triés. Elle consiste à subdiviser le tableau en un ensemble d'étages (k étage, et on parle de dichotomie étagée k). Ensuite de localiser la valeur recherchée dans quelle zone se trouve-t-elle. On refait la même subdivision pour la zone trouvée (en k étages). Et de refaire la même chose pour la zone localisée. On arrête la subdivision quand la zone localisée est indivisible par k, où on fait une recherche séquentielle de la valeur.

Exemple : On veut chercher la valeur **3** dans le tableau T de 15 éléments :

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	2,5	2,75	3	8	12	23	45	78	90	111	130	145	178

- Pour un $k=5$. Donc on doit subdiviser le tableau : $15/5=3$. Donc chaque étage doit contenir 3 éléments. Les cinq zones seront : [1,3], [4,6], [7,9], [10,12], [13,15] On doit comparer 3 avec les bornes de ces étages pour savoir elle est où : $3 > T[1]$ et $3 > T[3]$, donc 3 n'est pas dans le premier étage. $T[4] < 3 < T[5]$, donc 3 peut exister dans ce deuxième étage ... on arrête la comparaison avec les autres bornes.
- On essaye de subdiviser cet étage : on voit que cet étage contient trois éléments : T[4], t[5], T[6] c'est indivisible par 5 donc on fait une recherche séquentielle pour trouver que $3=T[5]$.

Questions :

1. C'est quoi l'avantage de cette technique de recherche ?
2. Faites une analyse de ce problème:
 - a) Proposer une décomposition fonctionnelle de ce problème: les fonctions (tâches) que le programme doit assurer. Représenter ceci sous forme de diagrammes de boîtes (une fonction sous formes de boîte, par exemple).
 - b) Y'a-t-il des choix à prendre par le programmeur dans la résolution de ce problème ? Donnez des exemples. Et faites vos choix.
3. Enrichir la décomposition faite dans la question (2.a), et proposer les algorithmes et les structure de données convenables pour résoudre le problème de la dichotomie étagée.
4. Exécuter cette solution sur des exemples et voir si votre solution est correcte pour ces exemples.
5. Programmer en C++ votre solution.