

Programmation 1 – Licence MPCI – TP 4

(sources : certains exercices proviennent de la [page de H. Garreta](#))

Exercice 1

RECHERCHE SÉQUENTIELLE. Ecrire une fonction dont le prototype est

```
int rang(int T[N], int X, int n);
```

qui, étant donné un tableau d'entiers T de taille N , dont seuls les n premiers éléments ont une valeur utile et un nombre entier X , renvoie le plus petit rang i (i est donc plus petit que $n-1$) tel que $T[i]=X$. Si X ne fait pas partie des n premiers éléments de T , alors la fonction renvoie -1 .

Ecrire un programme (i.e. `main`) qui utilise cette fonction.

Exercice 2

RECHERCHE DICHOTOMIQUE DANS UN TABLEAU ORDONNÉ. Ecrire une fonction dont le prototype est

```
int rang_dicho(int T[N], int X, int n);
```

qui, étant donné tableau d'entiers deux à deux distincts et triés par ordre croissant T de taille N , tel que seuls les n premiers éléments de T ont une valeur utile et un nombre X ,

- renvoie le rang i tel que $T[i]=X$ si celui existe;
- renvoie le plus petit rang i tel que $T[i]>X$.

La fonction `rang_dicho` de doit pas modifier T et doit mettre en œuvre le principe de la dichotomie, qui est rappelé ci-après.

Dichotomie pour la recherche d'un élément. Considérer deux indices i et j tels que le sous-tableau $[T[i] \dots T[j]]$ soit seul susceptible de contenir X (initialement $i=0$ et $j=n-1$). En comparant X et l'élément du milieu du sous-tableau, déterminer celle des deux moitiés du sous-tableau qui est susceptible de contenir X . Recommencer cette opération jusqu'à déterminer une unique position du tableau.

Exercice 3

A VOS MÉNINGES !

1. Ecrire une fonction dont le prototype est

```
float puissance(float x, int p);
```

qui calcule x^n en se reposant sur la relation de récurrence suivante :

$$x^n = \begin{cases} 1 & \text{si } n = 0 \\ x^{n/2}x^{n/2} & \text{si } n \text{ est pair} \\ x \times x^{(n-1)/2}x^{(n-1)/2} & \text{sinon} \end{cases}$$

2. Ecrire une fonction

```
float zero(int n, float v, float a, float b, float epsilon);
```

qui, en utilisant le principe de dichotomie vu en terminale, renvoie une valeur x comprise entre a et b telle que $|x^n - v| \leq \text{epsilon}$ si elle existe et qui renvoie $a-1$ sinon.

3. A l'aide de ce qui précède écrire une fonction

```
float racine(float v, int n);
```

qui renvoie la racine n -ième de v (avec une précision que vous aurez choisie).