

SN/UCAD/ESP/DGI/DIC1/Langage C

Série d'exercices (tableaux, structures)

Exercice 1

On se propose de déterminer la moyenne olympique d'une suite de N notes. Celles-ci sont données sous la forme d'un tableau. Il convient de supprimer de ces notes la plus élevée (ou l'une des plus élevées en cas de *ex-aequo*) et la plus faible, et de calculer la moyenne des $N - 2$ restantes.

Écrire une fonction dont le prototype est le suivant :

```
double moy_olymp(int notes[], int N);
```

Exercice 2

On désire manipuler des polynômes de degré arbitraire (inférieur à une valeur maximale prédéterminée, par exemple 31).

- imaginer une représentation, basée sur un tableau, pour de tels polynômes ;
- implanter une procédure d'évaluation de la fonction polynomiale en un point donné ;
- réaliser la procédure d'addition de deux polynômes.

Exercice 3

On se propose de construire un histogramme décrivant la répartition des valeurs des éléments d'un vecteur tel que celui-ci :

```
{ 2, 3, 7, 6, 9, 11, 12, 15, 18, 17, 14, 13, 12, 7, 8, 7, 5, 3, 2, 1, 1, 0, 2, 8, 11, 13, 12, 11, 6, 3};
```

Le programme affichera l'histogramme dans la fenêtre du terminal, en utilisant des blancs et des étoiles, comme dans l'exemple de la figure.

Écrire une fonction dont le prototype est :

```
void histog(int v[], int nb)
```

Modifier le programme afin de pouvoir choisir la hauteur de l'histogramme. Le prototype devient :

```
void histog(int v[], int nb, int H)
```



Exercice 4

Écrire une fonction qui réalise une permutation aléatoire d'un tableau d'entiers (on peut utiliser la fonction `rand`). Si possible, cette fonction ne doit pas créer de copie du tableau, mais le permuer « en place ».

Le prototype de cette fonction doit être :

```
void permute_tableau (int tab[], int N);
```

Exercice 5

FIG. 4.2 – Histogramme

On veut maintenant analyser le générateur de nombres aléatoires fourni par le système, `rand()`. On tirera 10000 valeurs aléatoires, comprises entre 0 et 50, dont on étudiera la répartition au moyen du générateur d'histogrammes de l'exercice 3.

Exercice 6: Chaînage par tableau

On a relevé sur un bordereau de saisie, les résultats obtenus par des étudiants à un examen. Chaque ligne du bordereau donne le numéro et le nom d'un étudiant ainsi que la note correspondante. Le nombre d'étudiants NBETU est supposé inférieur à 100.

On veut classer les étudiants par ordre de mérite au moyen d'un tableau VETU. Pour cela on crée un vecteur SUIVANT, d'entiers, de même taille que VETU et une variable DEB qui contiendra l'indice de la case de VETU contenant les informations sur l'étudiant le plus méritant.

1. Écrire une procédure qui parcourt le vecteur VETU et chaîne entre eux les éléments VETU[i], où $i \in [1, \text{NBETU}]$, à l'aide du tableau SUIVANT de telle sorte que :
 - VETU[i] \geq VETU[SUIVANT[i]]
 - et SUIVANT[i] = 0 si l'élément d'indice i correspond à l'étudiant ayant la plus petite note.
2. En déduire un programme qui saisit les informations sur les étudiants d'une classe et affiche par ordre de mérite les résultats.