1.

1° Donner un algorithme qui trouve le plus petit et le plus grand élément d'un tableau contenant n entiers naturels en effectuant un nombre de comparaisons de l'ordre de $\frac{3n}{2}$ (prouver votre programme et sa complexité, bien sûr !).
Indication : considérer les éléments du tableau 2 par 2 et commencer par comparer 2 éléments successifs.

2° Un étudiant a proposé la solution suivante :

```
let minmax  v =
  let min = ref 0 and max = ref 0 in
  for i=1 to vect_length(v)-1 do
     if v.(i) < v.(!min) then min:= i
     else if v.(i) > v.(!max) then max:=i done; (!min,!max);;
```

Ainsi le nombre de comparaisons effectuées pour chaque i est soit 1 soit 2. L'étudiant se demande si sa solution bien que fausse dans le cas le pire ne serait pas correcte en moyenne.
Le but de cet exercice est de montrer qu'il n'en est rien. Pour ce faire on retient le modèle des permutations et au lieu d'examiner les tableaux à trier on se penche sur la permutation associée, c'est à dire que l'on suppose que l'algorithme est exécuté sur un tableau α contenant tous les entiers naturels entre 1 et n, et que toutes les permutations de ces nombres sont équiprobables.

a. Pour toute permutation α on on appelle minimum local un entier j tel que :

$$\forall\ i \in [\![1, j[\![,\ \alpha(i) > \alpha(j)$$

On convient que 1 est toujours un minimum local de α. Exprimer le nombre de comparaisons effectuées par l'algorithme précédent en fonction du nombre k de minima locaux de la permutation α.

b. On note par $u_{n,k}$ le nombre de permutations sur n éléments qui présentent k minima locaux. Montrez que :

$$u_{n,0} = 0 \qquad u_{n,n} = 1$$

$$\forall\ k \in [\![1, n[\![\quad u_{n,k} = (n-1)u_{n-1,k} + u_{n-1,k-1}$$

c. On considère la famille de polynômes

$$P_n(x) = \sum_{k=1}^{\infty} u_{n,k} x^k$$

Donnez une expression de $P_n(x)$ comme un produit de monômes du premier degré. Montrez que la dérivée $P'_n(x)$ satisfait

$$P'_n(x) = P_n(x)\left(\frac{1}{x} + \frac{1}{x+1} + \ldots + \frac{1}{x+n-1}\right)$$

d. Donnez une expression du nombre moyen M_n de minima locaux dans les permutations sur n éléments.

e. Quel est le comportement à l'infini du nombre moyen de comparaisons dans le programme de l'étudiant ; le comparer à $\frac{3n}{2}$.

2.

Soit $x_1, x_2, \ldots, x_n$ une suite finie de nombre entiers non tous nécessairement distincts. La « multiplicité » d'une valeur x dans la suite est égale au nombre de fois où x apparaît dans la suite. Un entier z est dit « valeur majoritaire » si sa multiplicité est supérieure ou égale à $E(n/2) + 1$.

1$^{\mathbf{o}}$ Montrer que si x_i est différent de x_j et si l'on supprime x_i et x_j de la suite, la valeur majoritaire de la suite initiale, s'il en existe une, est aussi valeur majoritaire de la suite obtenue après suppression.

2$^{\mathbf{o}}$ Montrer qu'en examinant successivement les éléments de la suite dans l'ordre $x_1, x_2, \ldots, x_n$, on peut « mettre à jour » deux variables C et M ayant la propriété suivante :

lorsque l'on considère x_i, C contient une valeur qui est la seule candidate possible à être valeur majoritaire parmi $x_1, x_2, \ldots, x_{i-1}$ et M contient le nombre de fois où la valeur C est apparue jusqu'alors, si l'on exclue les fois où C a été éliminé.

3$^{\mathbf{o}}$ En déduire un algorithme qui, en deux « parcours » de la suite $x_1, x_2, \ldots, x_n$ détermine si la suite possède ou non une valeur majoritaire et donne cette valeur majoritaire quand elle existe.