

Mamouni My Ismail  
Professeur agrégé de mathématiques  
Enseignant en classes de MP  
CPGE My Youssef  
Rabat, Maroc  
mamouni.myismail@gmail.com  
myismail.chez.com

Corrigé abrégé

Epreuve Informatique

Concours Centrale Sup Elec, 2004

> **restart:**

On commence par déclarer les écoles, le nombre de postes ouverts pour chaque école

> **Ecoles** := [E1, E2, E3]; **N** := [1, 2, 1]; **candidats** := [C1,  
> C2, C3, C4];

$$Ecoles := [E1, E2, E3]$$
$$N := [1, 2, 1]$$
$$candidats := [C1, C2, C3, C4]$$

Puis le classement de chaque école pour les candidats

> **E1** := [C2, C4]; **E2** := [C4, C3, C1, C2]; **E3** := [C1, C2, C3];

$$E1 := [C2, C4]$$
$$E2 := [C4, C3, C1, C2]$$
$$E3 := [C1, C2, C3]$$

Les voeux de chaque candidat classés dans l'ordre de préférence, la 1ère case représente un numéro pour identifier le candidat

> **C1** := [1, 3, 2]; **C2** := [2, 3, 2, 1]; **C3** := [3, 2, 3]; **C4** := [4,  
> 1, 2];

$$C1 := [1, 3, 2]$$
$$C2 := [2, 3, 2, 1]$$
$$C3 := [3, 2, 3]$$

$$C_4 := [4, 1, 2]$$

Pour chaque école, on va chercher les candidats à éliminer, en commençant bien sûr par le dernier classé pour cette même école. Pour cela on dénombre les candidats les mieux classés pour lesquels cette école représente le 1<sup>er</sup> choix, si ce nombre dépasse le nombre de poste ouverts par la dite école, alors notre candidat est éliminé.

```
> choix_ecole:=proc(i) local
> Ecole,j,Nbr,k,candidat,a,b,NewEcole;
> Ecole:=op(i,Ecoles):NewEcole:=NULL:
> for j from nops(Ecole) to 1 by -1 do
>   Nbr:=0;
>   for k from j-1 to 1 by -1 do
>     candidat:=op(k,Ecole):
>     if op(2,candidat)=i then Nbr:=Nbr+1:
>     else
>     fi:
>   od:
>   if Nbr>=op(i,N) then print('Le candidat
>   éliminé est numéro'=op(1,op(j,Ecole)));
>   else print('Le candidat non éliminé
>   '=op(1,op(j,Ecole)));NewEcole:=NewEcole,C[op(1,op(j,Ecole))]:
>   fi:
>   od:
>   print('Le choix de l'école est
>   '= [NewEcole]);RETURN([NewEcole]);
> end:
On applique notre petit programme pour le 1er école
> E[1]:=choix_ecole(1);
```

*Le candidat non élimin = 4*

*Le candidat non élimin = 2*

*Le choix de l'école est = [C<sub>4</sub>, C<sub>2</sub>]*

$$E_1 := [C_4, C_2]$$

On applique notre petit programme pour le 2<sup>ème</sup> école

```
> E[2]:=choix_ecole(2);
```

*Le candidat non élimin = 2*

*Le candidat non élimin = 1*

*Le candidat non élimin = 3*

*Le candidat non elimin = 4*

*Le choix de l'cole est = [C<sub>2</sub>, C<sub>1</sub>, C<sub>3</sub>, C<sub>4</sub>]*

$E_2 := [C_2, C_1, C_3, C_4]$

On applique notre petit programme pour le 3<sup>ème</sup> école

```
> E[3]:=choix_ecole(3);
```

*Le candidat elimin est numro = 3*

*Le candidat elimin est numro = 2*

*Le candidat non elimin = 1*

*Le choix de l'cole est = [C<sub>1</sub>]*

$E_3 := [C_1]$

Maintenant chaque candidat va éliminer toute école qu'elle juge inutile, c'est à dire pour la quelle il est sûr d'avoir mieux. Comment ? il commence par la dernière et il vérifie si une école mieux classée va le prendre dans sa liste principale.

```
> for i from 1 to nops(Ecoles) do
> liste_principale[i]:=[seq(op(j,E[i]),j=1..N[i
> ])];
> od;
```

*liste\_principale<sub>1</sub> := [C<sub>4</sub>]*

*liste\_principale<sub>2</sub> := [C<sub>2</sub>, C<sub>1</sub>]*

*liste\_principale<sub>3</sub> := [C<sub>1</sub>]*

On s'inspirant de cet exemple on inverse les role écoles-candidats pour trouver les écoles à éliminer pour chaque candidat....A vous de le faire