

Exercice :

Afin de répondre aux besoins des écoles primaires d'une région de la Tunisie. Le ministère de l'Éducation charge une commission de visiter ces écoles, de chercher leurs besoins en Matériels (Tableaux à craie et tables) et de déterminer le(s) école(s) ayant le besoin le plus élevé.

Pour automatiser les tâches de cette commission, on se propose d'écrire un programme permettant :

- de remplir un premier tableau **TE** par les **N** codes des écoles primaires avec $2 \leq N \leq 250$. Le code d'une école est une chaîne de 6 chiffres
- de remplir un deuxième tableau **TB** par les besoins matériels des écoles en tableaux à craie et en tables sous le format suivant : **Nombre_de_tableaux#Nombre_de_tables** sachant que chaque besoin ne doit pas dépasser **999**. Exemples: "852#999" . "0#105" . "52#0"
- de calculer dans un troisième tableau **TBU**, le budget nécessaire pour chaque école calculé comme suit : $p1 * \text{Nombre_de_tableaux} + p2 * \text{Nombre_de_tables}$ sachant que **p1** et **p2** sont respectivement les prix d'un tableau à craie et d'une table et qui sont deux constantes dont les valeurs en dinars sont 80 et 160
- d'afficher toutes les écoles ayant le besoin budgétaire le plus élevé en matériels

Exemple : Pour **N=8** et les deux tableaux **TE** et **TB** suivants :

TE :

310600	035100	199600	001800	421200	600700	059700	776600
1	2	3	4	5	6	7	8

TB :

10#8	100#16	17#0	32#50	8#0	0#0	5#5	2#10
1	2	3	4	5	6	7	8

Le tableau **TBU** sera :

TBU :

2080	10560	1360	10560	640	0	1200	1760
1	2	3	4	5	6	7	8

En effet dont le code est "310600" a besoin de **10** tableaux à craie et de **8** tables ("10#8"). Le budget est calculé comme suit: $10 * 80 + 8 * 160 = 2080$

le programme affiche : Le besoin budgétaire le plus élevé est 105690 D et les écoles concernées sont : 035100-001800

SOLUTION EN PYTHON :

```
from numpy import*
```

```
def saisie ():
    valide=False
    while valide==False:
        n=int(input("donner n"))
        valide=2<=n<=250
    return n
```

```
def remplir(n,te):
    for i in range (n):
        valide=False
        while valide==False:
            te[i]=input("t["+str(i)+"]")
            valide=len(te[i])==6
```

```
def remplir2(n,TB,TBU):
    p1=80
    p2=160
    for i in range(n):
        valide=False
        while valide==False:
            nb1=int(input("donner nb1"))
            nb2=int(input("donner nb2"))
            valide=0<=nb1<=999 and 0<=nb2<=999
        TB[i]=str(nb1)+"#"+str(nb2)
        TBU[i]=p1*nb1+p2*nb2
```

```
def maximum (n,TBU):
    max=TBU[0]
    for i in range(n):
        if TBU[i]>max:
            max=TBU[i]
    return max
```

```
def afficher (n,max,te,TBU):  
    ch=""  
    for i in range(n):  
        if TBU[i]==max:  
            ch=ch+te[i]+"-"  
    print("le besoin budgétaire le plus élevé",max,"et les écoles concédés  
sont",ch )
```

```
#p.p
```

```
te=array([str]*250)  
TB=array([str]*250)  
TBU=array([int]*250)  
n=saisie()  
remplir(n,te)  
remplir2(n,TB,TBU)  
max=maximum (n,TBU)  
afficher(n,max,te,TBU)
```