

**BREVET DE TECHNICIEN SUPÉRIEUR**  
**SERVICES INFORMATIQUES AUX ORGANISATIONS**

**SESSION 2012**

**Sous-épreuve E 22 Algorithmique appliquée**

- Il est formellement interdit d'utiliser toute connexion à un réseau interne ou externe au centre d'examen, quel qu'en soit le procédé.
- L'usage d'une calculatrice est autorisée.
- Cette épreuve comporte 2 parties :
  - une première partie qui dure 30 minutes à l'issue de laquelle vous devez fournir une production écrite répondant au sujet.
  - une deuxième partie qui dure également 30 minutes. Vous travaillerez sur un des ordinateurs d'examen pour coder les algorithmes papiers sous Python.  
Vous enregistrerez votre travail sur une clé USB fournie par votre professeur.  
Vous imprimerez votre travail et le signerez.
- Vous n'oubliez pas de rendre :
  - Le sujet.
  - L'algorithme papier où figureront votre nom et prénom.
  - L'impression de votre programme signée

Nom :

Prénom :

**Problème :**

On considère une liste de noms. Le but de l'exercice est de tirer une lettre au hasard dans l'alphabet et d'afficher les noms de la liste commençant par cette lettre.

L'algorithme et la traduction Python du programme principal sont :

**Variables :**

- listenoms,alphabet : tableau de chaînes de caractères
- nom,lettre : chaîne de caractères
- c :entier

**Début**

```
listenoms=["Alain","Antoine", "Bernard", "Colin", "Christine",
"François", "Guy","Gérard", "Léa","Léon","Louis",
"Nathalie","Serge","Sylvie","Sylvain","Vincent"]
```

```
alphabet=["A","B","C","D","E","F","G","H","I","J","K",
"L","M","N","O","P","Q","R",
"S","T","U","V","W","X","Y","Z"]
```

```
# Choix de la lettre au hasard
```

```
lettre ← lettre_au_hasard()
```

```
# Liste des noms commençant par la lettre choisie précédemment
```

```
noms← noms(listenoms,lettre)
```

```
# Affichage du résultat
```

```
c ← longueur(noms)
```

```
Si c=0 alors
```

```
    écrire : Il n'y a pas de noms commençant par la lettre ,
```

```
    lettre
```

```
FSi
```

```
Sinon si c=1 alors
```

```
    écrire : Il y a 1 nom commençant par la lettre , lettre
```

```
    écrire : Ce nom est, noms
```

```
FSinonsi
```

```
Sinon :
```

```
    écrire : Il y a ,c, noms commençant par la lettre , lettre
```

```
    écrire : Ces noms sont , noms
```

```
FinSinon
```

**Fin**

```
# Fonctions
```

```
import random # permet d'utiliser l'instruction
                  random.randint(a,b)
```

```
# Programme
```

```
listenoms=["Alain","Antoine", "Bernard", "Colin",
"Christine","François", "Guy","Gérard", "Léa","Léon",
"Louis", "Nathalie","Serge","Sylvie","Sylvain"]
```

```
alphabet=["A","B","C","D","E","F","G","H","I",
"J","K","L","M","N","O","P","Q","R","S","T",
"U","V","W","X","Y","Z"]
```

```
# Choix de la lettre au hasard
```

```
lettre = lettre_au_hasard()
```

```
# Liste des nom commençant par la lettre choisie
                  précédemment
```

```
noms=noms(listenoms,lettre)
```

```
# Affichage du résultat
```

```
c=len(noms)
```

```
if c==0 :
```

```
    print("Il n'y a pas de noms commençant
          par la lettre ", lettre)
```

```
elif c==1 :
```

```
    print("Il y a 1 nom commençant par la lettre ",
          lettre)
```

```
    print("Ce nom est", noms)
```

```
else :
```

```
    print("Il y a ",c," noms commençant ,
          par la lettre ",lettre)
```

```
    print("Ces noms sont ", noms)
```

**Partie A :****8 points**

A traiter sur feuille et à rendre avant d'accéder aux machines.  
Durée maximum : 30 minutes

**Exercice 1**

Écrire une fonction `lettre_au_hasard()` , sans paramètre, et qui retourne une lettre choisie au hasard de la liste `alphabet` .

*Aide : on pourra utiliser l'instruction `alea(0,25)` qui restitue de manière aléatoire un entier entre 0 et 25.*

*Exemple : Si le nombre choisi au hasard est 2, `lettre_au_hasard()` renvoie la lettre C.*

**Exercice 2**

Écrire une fonction `noms(liste,lettre)` dont les paramètres sont une liste de noms, `liste` et une lettre, `lettre`. Cette fonction créera la liste des noms de `liste` commençant par la lettre `lettre` et retournera cette liste.

Exemple :

```
listenoms=["Alain","Antoine","Bernard","Colin","Christine","François","Guy","Gérard","Léa","Léon","Louis",  
"Nathalie","Serge","Sylvie","Sylvain","Vincent"]
```

- `noms(listenoms,"A")` renvoie la liste `["Alain","Antoine"]`
- `noms(listenoms,"W")` renvoie la liste `[]` (la liste vide).

**Partie B :****8 points**

A traiter sur un ordinateur en utilisant le langage Python.  
Vous enregistrerez votre travail sous votre nom.  
Durée maximum : 30 minutes

**Exercice 3**

Coder sous Python la fonction `lettre_au_hasard()` de l'exercice 1 puis la tester.

On utilisera l'instruction `random.randint(0,25)` qui restitue de manière aléatoire un entier entre 0 et 25

**Exercice 4**

Coder sous Python la fonction `noms(liste,lettre)` de l'exercice 2 puis la tester.

**Exercice 5**

Télécharger le programme principal. Faites un copier coller sur votre page contenant les fonctions puis le tester.

**BREVET DE TECHNICIEN SUPÉRIEUR**  
**SERVICES INFORMATIQUES AUX ORGANISATIONS**

**Aide mémoire**

- `a // b` calcule le quotient entier de la division de `a` par `b`.  
Exemple : `13 // 5` donne 2
- `a % b` calcule le reste de la division de `a` par `b`.  
Exemple : `13 % 5` donne 3
- `a / b` calcule le quotient de `a` par `b`.  
Exemple : `13 / 5` donne 2.6
- `a * b` calcule le produit de `a` par `b`.  
Exemple : `3 * 5` donne 15
- `a ** b` calcule `a` à la puissance `b`.  
Exemple : `2 ** 3` donne 8
- Vrai et Faux sont les booléens `True` et `False`.
- `liste[i]` restitue l'élément de la liste de rang `i`.  
Exemple : `Liste=[10,25,33,4,8]` `Liste[1]` renvoie 25.
- Une matrice est une liste de listes.  
Exemple :
 

– <code>M=[[1, 2, 3, 4, 5], [6, 7, 8, 9, 10], [11, 12, 13, 14, 15]]</code>	<code>==&gt;</code>	<table style="border-collapse: collapse;"> <tr><td style="padding-right: 10px;">1</td><td style="padding-right: 10px;">2</td><td style="padding-right: 10px;">3</td><td style="padding-right: 10px;">4</td><td style="padding-right: 10px;">5</td></tr> <tr><td style="padding-right: 10px;">6</td><td style="padding-right: 10px;">7</td><td style="padding-right: 10px;">8</td><td style="padding-right: 10px;">9</td><td style="padding-right: 10px;">10</td></tr> <tr><td style="padding-right: 10px;">11</td><td style="padding-right: 10px;">12</td><td style="padding-right: 10px;">13</td><td style="padding-right: 10px;">14</td><td style="padding-right: 10px;">15</td></tr> </table>	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	2	3	4	5													
6	7	8	9	10													
11	12	13	14	15													
- `M[0]` est la première ligne de la matrice : `[1,2,3,4,5]`
- `M[1][2]` est la valeur de la deuxième ligne, troisième colonne soit 8.
- `len(liste)` restitue la longueur de la liste.  
Exemple : `Liste=[10,25,33,4,8]`, `Len(Liste)` renvoie 5.
- `Liste.append(a)` ajoute l'élément `a` à la liste `Liste`  
Exemple : `Liste=[10,25,33,4,8]`, `Liste.append(100)` renvoie `[10,25,33,4,8,100]`
- `Liste.insert(i,a)` ajoute l'élément `a` au rang `i` de la liste `Liste`  
Exemple `Liste=[10,25,33,4,8]`, `Liste.insert(0,100)` renvoie `[100,10,25,33,4,8]`
- `range(a,b)` est la liste des entiers compris entre `a` et `b`, `b` non compris.  
Exemple :
  - `range(0,5)` renvoie la liste `[0,1,2,3,4]`
  - `for i in range(0,5) :`  

.....

 correspond à : Pour `i` allant de 0 à 4
- Chaîne de caractères. Comme pour les listes :
  - `len(ch)` donne la longueur de la chaîne de caractères `ch`.
  - `ch[0]` est le premier caractère de la chaîne `ch`
 Exemple : `ch="AMI"`, `len(ch)` renvoie 3. `ch[0]` renvoie A. `ch+"S"` renvoie "AMIS"
- Syntaxe pour l'écriture d'une fonction :  
Exemple :
 

<pre>&gt;&gt;&gt; def somme_produit(a,b):       somme=a+b       produit=a*b       return somme,produit</pre>	donne	<pre>&gt;&gt;&gt; somme_produit(3,4) (7, 12) &gt;&gt;&gt; a,b=somme_produit(3,4) &gt;&gt;&gt; a 7 &gt;&gt;&gt; b 12</pre>
--	-------	---