

**Exercices spécifiques sur les listes**

**Exercice 1B.1 :**

Parmi les carrés parfaits à cinq chiffres, sauvegarder dans une liste ceux possédant une séquence de trois chiffres consécutifs dans l'ordre croissant ?

**Exercice 1B.2 :**

Sauvegarder dans une liste tous les carrés parfaits dont les chiffres sont dans un ordre strictement décroissant.

**Exercice 1B.3 :**

Sauvegarder dans une liste les trois plus petits carrés parfaits possédant 4 fois le chiffre 2 ?

**Exercice 1B.4 :**

Sauvegarder dans une liste les trois plus petits carrés parfaits comportant une séquence de trois chiffres 7 consécutifs ?

**Exercice 1B.5 :**

Sauvegarder dans une liste les dix premiers carrés parfaits dont les chiffres sont symétriques (ex : 12321 ou 247742) ?

**Exercice 6 :**

Ecrire un programme qui trouvera tous les nombres entre 500 et 1200 (les deux inclus), divisibles par 7 mais non multiples de 5 et de 3, dont la somme des chiffres vaut 17. Les nombres obtenus doivent être imprimés dans une liste que l'on affichera.

**Exercice 1B.7 :**

Sauvegarder dans une liste les nombres à neuf chiffres, composés de trois triplets de trois chiffres chacun, tels que :

- le nombre correspondant au premier triplet est égal au tiers du nombre correspondant au dernier triplet
- le nombre correspondant au triplet du milieu est obtenu en soustrayant le premier triplet du troisième.

Chaque chiffre de 1 à 9 est utilisé une et une seule fois.

**CORRIGE – Notre Dame de La Merci – Montpellier – M. Quet**

**Exercice 1B.1 :**

*Parmi les carrés parfaits à cinq chiffres, sauvegarder dans une liste ceux possédant une séquence de trois chiffres consécutifs dans l'ordre croissant ?*

```
def liste_des_chiffres(n):
    u = []
    for i in range(5):
        u.append(n//10**(4-i))
        n -= u[i]*10**(4-i)
    return(u)

def test_valeurs(a,b,c,d,e):
    z = 0
    if (b==a+1 and c==b+1) or (c==b+1 and d==c+1) or (d==c+1 and e==d+1):
        z = 1
    return z

for i in range(100, 317):
    v = []
    test = 0
    j = i**2
    v = liste_des_chiffres(j)
    test = test_valeurs(v[0],v[1],v[2],v[3],v[4])
    if test == 1:
        print(i,j)
```

On obtient :

```
106 11236
111 12321
116 13456
153 23409
186 34596
281 78961
```

**NB :** La commande `list(str(n))` crée une liste alphanumérique des chiffres d'un nombre :

```
def test_valeurs(a,b,c,d,e):
    z = 0
    if (b==a+1 and c==b+1) or (c==b+1 and d==c+1) or (d==c+1 and e==d+1):
        z = 1
    return z

for i in range(100, 317):
    v = []
    test = 0
    j = i**2
    v = list(str(j))
    test = test_valeurs(int(v[0]),int(v[1]),int(v[2]),int(v[3]),int(v[4]))
    if test == 1:
        print(i,j)
```

**Exercice 1B.2 :**

Sauvegarder dans une liste tous les carrés parfaits dont les chiffres sont dans un ordre strictement décroissant.

```

liste = []
for i in range(4, 10000):
    v = []
    j = i**2
    v = list(str(j))
    v.reverse()
    w = v[:]
    w.sort()
    z = 0
    for k in range(0, len(w)-1):
        if w[k] == w[k+1]:
            z = 1
    if v == w and z == 0:
        liste.append(j)
        print(i, j)
print(liste)

```

On obtient :

```

8 64
9 81
29 841
31 961
[64, 81, 841, 961]

```



**Exercice 1B.3 :**

Sauvegarder dans une liste les trois plus petits carrés parfaits possédant 4 fois le chiffre 2 ?

```

z = 0
i = 33
liste = []
while z < 3:
    v = []
    j = i**2
    v = list(str(j))
    if v.count('2') >= 4:
        print(i, j)
        liste.append(j)
        z += 1
    i += 1
print(liste)

```

On obtient :

```

1415 2002225
1484 2202256
1485 2205225
[2002225, 2202256, 2205225]

```



**Exercice 1B.4 :**

Sauvegarder dans une liste les trois plus petits carrés parfaits comportant une séquence de trois chiffres 7 consécutifs ?

```

z = 0
i = 10
liste = []

```

```

while z < 3:
    v = []
    j = i**2
    v = list(str(j))
    for k in range(0,len(v)-2):
        if '7' in v[k] and '7' in v[k+1] and '7' in v[k+2]:
            print(i,j)
            liste.append(j)
            z += 1
    i += 1
print(liste)

```

On obtient :

```

527 277729
882 777924
1076 1157776
[277729, 777924, 1157776]

```



### Exercice 1B.5 :

Sauvegarder dans une liste les dix premiers carrés parfaits dont les chiffres sont symétriques (ex : 12321 ou 247742) ?

```

z = 0
i = 1
liste = []
while z < 10:
    v = []
    j = i**2
    v = list(str(j))
    w = v[:]
    w.reverse()
    if v == w:
        print(i,j)
        z += 1
        liste.append(j)
    i += 1
print(liste)

```

On obtient :

```

1 1
2 4
3 9
11 121
22 484
26 676
101 10201
111 12321
121 14641
202 40804
[1, 4, 9, 121, 484, 676, 10201, 12321, 14641, 40804]

```



**Exercice 1B.6 :**

Ecrire un programme qui trouvera tous les nombres entre 500 et 1200 (les deux inclus), divisibles par 7 mais non multiples de 5 et de 3, dont la somme des chiffres vaut 17. Les nombres obtenus doivent être imprimés dans une liste que l'on affichera.

```

liste = []
for i in range(101):
    j = 700 + 7*i
    if j%5 != 0 and j%3 != 0:
        u = list(str(j))
        somme = 0
        for k in range(len(u)):
            somme += int(u[k])
        if somme == 17:
            liste.append(j)
print(liste)

```

On obtient :

[728, 791, 854, 917, 1169, 1358]



**Exercice 1B.7 :**

Sauvegarder dans une liste les nombres à neuf chiffres, composés de trois triplets de trois chiffres chacun, tels que :

- le nombre correspondant au premier triplet est égal au tiers du nombre correspondant au dernier triplet
- le nombre correspondant au triplet du milieu est obtenu en soustrayant le premier triplet du troisième.

Chaque chiffre de 1 à 9 est utilisé une et une seule fois.

Soit  $\overline{abcdefghi}$  le nombre cherché. On a :

$$\begin{cases} \overline{abc} = \frac{1}{3} \overline{ghi} \\ \overline{def} = \overline{ghi} - \overline{abc} \end{cases} \Leftrightarrow \begin{cases} \overline{3abc} = \overline{ghi} \\ \overline{def} = \overline{ghi} - \overline{abc} \end{cases} \quad \text{Ainsi : } \begin{cases} a \in \{1; 2; 3\} \\ g + h + i = k \times 3, k \in \mathbb{N} \end{cases}$$

Le premier triplet doit être compris entre 123 et 333.

Voici un programme en python :

```

liste = []
for i in range(123,334):
    v = []
    v = list(str(i))
    if v[0]!=v[1] and v[0]!=v[2] and v[1]!=v[2] and '0' not in v:
        w = []
        j = 3 * i
        w = list(str(j))
        if w[0]!=w[1] and w[0]!=w[2] and w[1]!=w[2] and '0' not in w:
            k = j - i
            x = []
            x = list(str(k))
            if x[0]!=x[1] and x[0]!=x[2] and x[1]!=x[2] and '0' not in x:
                if x[0] in v or x[0] in w or x[1] in v or x[1] in w or x[2] in v or x[2] in w:
                    z = 0
                else :
                    print(1000000*i+1000*k+j)
                    liste.append(1000000*i+1000*k+j)
print(liste)

```



On obtient 4 solutions :

192384576

219438657

273546819

327654981

[192384576, 219438657, 273546819, 327654981]