TEMA 3

Ejercicio 1:

Dada la lista $L = [3, 5, 6, 8, 10, 12]$, se pide:

(a) Averigua la posicion del numero 8:

```
L=[3,5,6,8,10,12]
print L.index(8)+1
```

Out: 4

(b) Cambiar el valor 8 por 9:

```
L1=list(L)
L1[3]=9
print L1
```

Out: $[3, 5, 6, 9, 10, 12]$

(c) Intercambiar 5 y 9:

```
L2=list(L1)
a=L2[3]
L2[3]=L2[1]
L2[1]=a
print L2
```

Out: $[3, 9, 6, 5, 10, 12]$

(d) Intercambiar cada valor por el que ocupa su posicion simetrica:

```
L3=list(L)
n=len(L3)
for i in srange (0,n/2):
    a=L3[i]
    L3[i]=L3[n-i-1]
    L3[n-i-1]=a
print L3
```

Out: $[12, 10, 8, 6, 5, 3]$

(e) Crear la lista resultante para concatenar a la original:

```
print L+L3
```

Out: $[3, 5, 6, 8, 10, 12, 12, 10, 8, 6, 5, 3]$

Ejercicio 2:

La orden prime range(100,1001) genera la lista de los numeros primos entre 100 y 1000.
Se pide, siendo primos=prime range(100,1001):

(a) Averiguar el primo que ocupa la posicion central:

```
primos=prim_range(100,1001)
i=floor(len(primos)/2)
print primos[i]
```

Out: 509

(b) Averiguar la posicion de 331 y 631:

```
print primos.index(331)+1
print primos.index(631)+1
```

Out: 4290

(c) Extraer la sublista de primos entre 331 y 631:

```
print primos[primos.index(331):primos.index(631)+1]
```

Out: [331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409,
419, 421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499,
503, 509, 521, 523, 541, 547, 557, 563, 569, 571, 577, 587, 593, 599, 601,
607, 613, 617, 619, 631]

(d) Extraer una sublista de primos que olvide los dos centrales de cada cuatro:

```
primos2=list() for i in srange (0,len(primos)):
if i%4 == 0 :
    primos2.append(primos[i])
if i%4 == 3 :
    primos2.append(primos[i])
```

Out: [101, 109, 113, 137, 139, 157, 163, 179, 181, 197, 199, 227, 229, 241,
251, 269, 271, 283, 293, 313, 317, 347, 349, 367, 373, 389, 397, 419, 421,
439, 443, 461, 463, 487, 491, 509, 521, 547, 557, 571, 577, 599, 601, 617,
619, 643, 647, 661, 673, 691, 701, 727, 733, 751, 757, 773, 787, 811, 821,
829, 839, 859, 863, 883, 887, 919, 929, 947, 953, 977, 983]

(e) Extraer una sublista de primos que olvide el tercero de cada tres:

```
primos3=list()
for i in srange (0,len(primos)):
    if i%3 == 2 :
```

```
        continue
    primos3.append(primos[i])
print primos3
```

Out: [101, 103, 109, 113, 131, 137, 149, 151, 163, 167, 179, 181, 193, 197,
211, 223, 229, 233, 241, 251, 263, 269, 277, 281, 293, 307, 313, 317, 337,
347, 353, 359, 373, 379, 389, 397, 409, 419, 431, 433, 443, 449, 461, 463,
479, 487, 499, 503, 521, 523, 547, 557, 569, 571, 587, 593, 601, 607, 617,
619, 641, 643, 653, 659, 673, 677, 691, 701, 719, 727, 739, 743, 757, 761,
773, 787, 809, 811, 823, 827, 839, 853, 859, 863, 881, 883, 907, 911, 929,
937, 947, 953, 971, 977, 991, 997]

Ejercicio 3:

Considerese el numero 1000!:
-En cuantos ceros acaba?

```
L=factorial(1000).digits()
cont=0
for i in srange (0,len(L)):
    if L[i]==0:
        cont=cont+1
    if L[i]!=0:
        break
print cont
```

Out: 249

-Se encuentra el numero 666 entre sus subcadenas? En caso afirmativo, localizar (encontrar los ndices de) todas las apariciones.

```
for i in srange (0,len(L)):
if L[i]==6:
    if L[i+1]==6:
        if L[i+2]==6:
            print "Yes in:"
            print i,i+1,i+2
```

Out: Yes in:
2182 2183 2184
Yes in:
2442 2443 2444

-Encontrar la subcadena mas larga de doses consecutivos. Mostrarla con los dos dgitos que la rodean.

```
contf=0
```

```
indice=-1
for i in srange (0,len(L)):
    if L[i]==2:
        cont=0
        for j in srange (i,len(L)):
            if L[j]==2:
                cont=cont+1
            if L[j]!=2:
                if cont¿contf:
                    contf=cont
                    indice=i
                break
print contf,L[indice-1:indice+contf+1]
```

Ejercicio 4:

Si se aplica la funcion sum() a una lista numerica, nos devuelve la suma de todos sus elementos. En particular la composicion sum(k.digits()), para k un variable entera, nos devuelve la suma de sus dgitos (en base 10). Calcular, con la composicion sum(k.digits()), la suma de los dgitos del numero k=factorial(1000) y
calcular la misma suma sin utilizar el metodo .digits()

```
print sum(factorial(1000).digits())
suma=0
for i in srange (0,len(L)):
    suma=suma+L[i]
print suma
```

Out: 10539 10539

Ejercicio 6:

Sin utilizar los metodos .divisors() ni la funcion max(), elabora codigo que, a partir de dos numeros a y b, calcule:
-El conjunto de divisores de a
-El conjunto de divisores de b
-El conjunto de divisores comunes y elegir el mayor de ellos

```
def divisores(a):
    A=set()
    for i in srange (1,a+1):
mif (a%i==0):
            A.add(i)
```

```
    return A
def divisores_comunes(A,B):
    return AB
def maximo_conj(C):
    J=list(C)
    res=1
    for i in srange (0,len(J)):
        if J[i]¿res:
            res=J[i]
    return res
A=divisores(100)
B=divisores(350)
C=divisores_comunes(A,B)
print A
print B
print C
print maximo_conj(C)
```

Out: $set([1, 2, 4, 5, 100, 10, 50, 20, 25])$
set([1, 2, 35, 5, 70, 7, 10, 14, 175, 50, 25, 350])
set([1, 2, 5, 10, 50, 25])
50