

EJERCICIO 13

a) Goldbach conjeturó que todo entero compuesto e impar es la suma de un primo y el doble de un cuadrado. Así, por ejemplo, $9 = 7 + 2 \cdot 1^2$, $15 = 7 + 2 \cdot 2^2$, $21 = 3 + 2 \cdot 3^2$, etc. Esta conjetura resultó ser falsa. Determina el menor entero que no cumple lo conjeturado por Goldbach.

```
for i in srange (3,1000000):
    flag=1
    if is_prime(i):
        continue
    else:
        if (i%2==1):
            flag=0
            for j in srange (1,i):
                if is_prime(j):
                    for k in srange (1,i):
                        if i==j+2*k^2:
                            flag=1

    if flag==0:
        print i
        break
```

Con este programa se puede ver que número entero no lo cumple. No he podido sacarlo ya que el tiempo de ejecución es muy largo.

b) Existen enteros, por ejemplo 145, que son iguales a la suma de los factoriales de sus dígitos. Determina todos los enteros con esta propiedad:

```
for i in srange (1,100000):
    L=[]
    L=i.digits()
    suma=0
    for j in srange (0,len(L)):
        suma=suma + factorial(L[j])
    if suma==i:
        print i
```

En este caso el siguiente numero después de 145 es 40585 y si lo comprobamos a mano, vemos que este número también lo cumple.

d) Sea (a, b, c) una tripleta de enteros positivos tal que existe un triángulo rectángulo con la longitud de los lados igual a los enteros de la tripleta. Podemos decir, por ejemplo, que una tal tripleta es rectangular. Sea p el perímetro de un tal triángulo. Para $p \leq 1000$ determina el perímetro p m para el que existe el mayor número de tripletas rectangulares distintas con ese perímetro:

```
L=[]
for i in srange (0,1000):
    L.append(0)
for i in srange (1,1000):
    a=i
```

```

for j in srange (1,1000):
    b=j
    c=a^2+b^2
    if is_square(c):
        c=sqrt(c)
        if (a+b+c)<=1000:
            L[a+b+c-1]=L[a+b+c-1]+1
maximo=L[0]
indice=0
for i in srange (0,1000):
    if L[i]>maximo:
        maximo=L[i]
        print maximo
        indice=i
print indice

```

Este programa guarda en una lista el número de veces que aparece el perímetro con esa condición e imprime el valor del perímetro que más aparece, que en este caso es 839.

EJERCICIO 8

```

def ciclos(d):
    D=list()
    for i in srange (0,d):
        num=i
        L=list()
        while 1:
            num=cuadrados_mod(num,d)
            if L.count(num)==1:
                index=L.index(num)
                for j in srange (0,index-1):
                    L.remove(L[j])
                if D.count(L)==1:
                    break
                D.append(L)
                break
            L.append(num)
    return D

```

Esta función nos devuelve una lista con las listas correspondientes a los ciclos pedidos en el ejercicio (aunque los devuelve repetidos).