TP0 - 3) Hello World para *Jupyter+SageMath*

Criação de Corpos Finitos Primos

Neste exercício 3-a) vamos criar corpos finitos distintos **Fp** para quatro valores diferente de **p*. *Estes valores de* *p** são todos valores primos e são da forma `2^k - 1` , onde cada *k** *é um valor primo também. Para tal criaremos os corpos finitos primos com os seguintes valores de* ***p****: *31,127,8191,131071* através da função do SageMath GF(p).

In [3]:
```
p1 = 31
corpoFinito1 = GF(p1)
print(corpoFinito1)

p2 = 127
corpoFinito2 = GF(p2)
print(corpoFinito2)

p3 = 8191
corpoFinito3 = GF(p3)
print(corpoFinito3)

p4 = 131071
corpoFinito4 = GF(p4)
print(corpoFinito4)
```

```
Finite Field of size 31
Finite Field of size 127
Finite Field of size 8191
Finite Field of size 131071
```

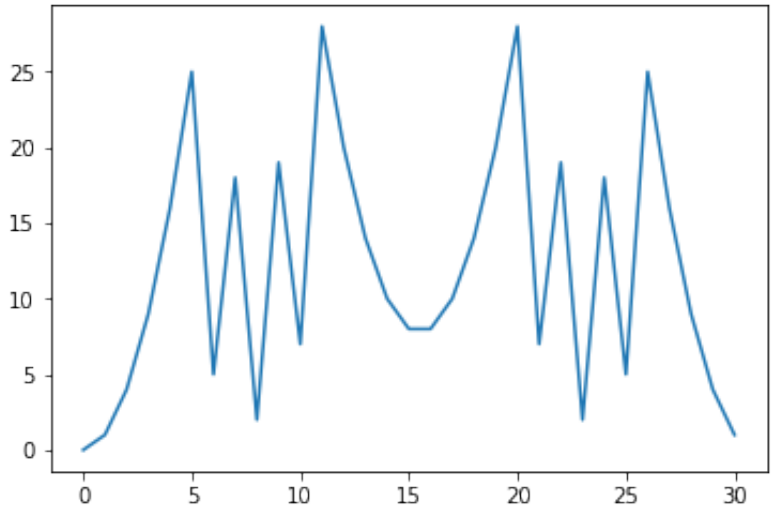
Plot da função x -> x^2 em cada Corpo Finito

Iremos agora executar um *plot* da função `x -> x^2` para cada corpo finito criado em cima. Com este *plot*, ferramenta do SageMath, irá ser criado um gráfico. Para o gráfico ser desenhado no nosso notebook será necessário a execução da função `plot.show()` .

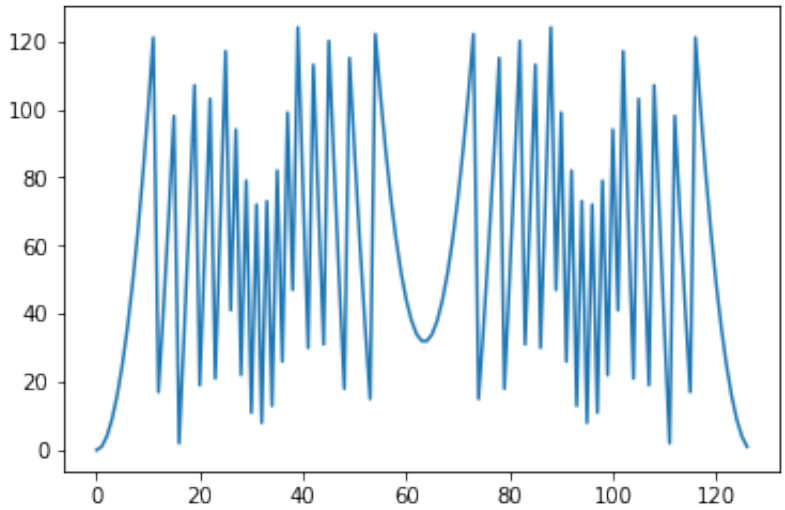
Para este exercício a função `plot` terá de receber uma lista de argumentos(que serão os valores do corpo finito) tendo por isso o seguinte formato: `plot([FUNCTION for x in LISTA])` . Também de realçar que a função pedida é uma função quadrática e como tal é necessário indicar ao SageMath que o x é elevado a 2. Para isso a expressão correta no SageMath é: `x**n` , onde n é o expoente. No nosso caso teremos `x**2` .

In [17]:
```
import matplotlib.pyplot as plt
```

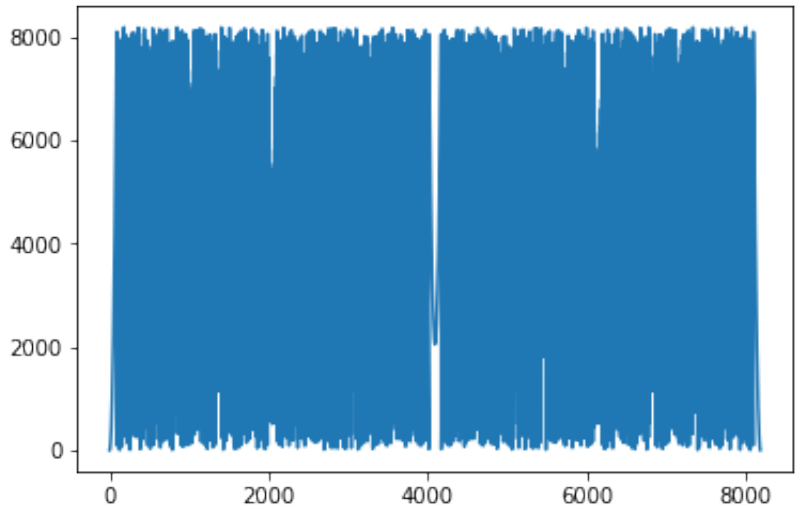
In [18]:
```
grafico1 = plt.plot([x**2 for x in corpoFinito1])
plt.show(grafico1)
```



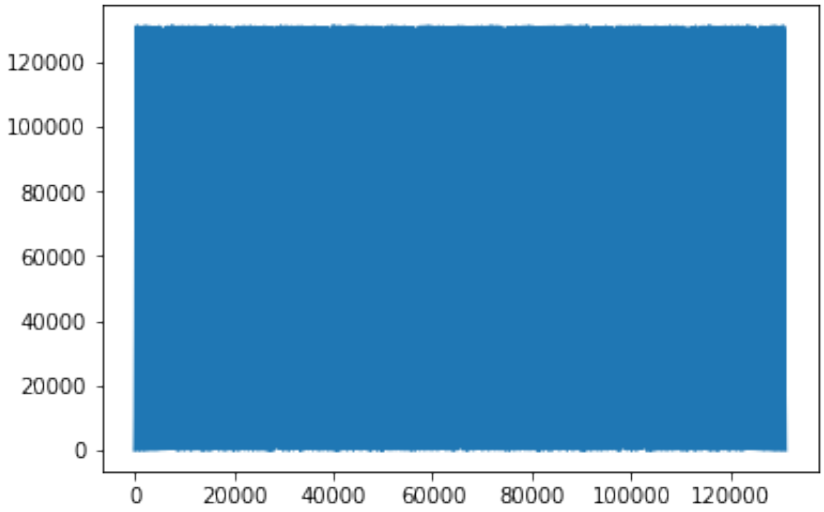
In [6]:
```
grafico2 = plt.plot([x**2 for x in corpoFinito2])
plt.show(grafico2)
```



In [8]:
```
grafico3 = plt.plot([x**2 for x in corpoFinito3])
plt.show(grafico3)
```



In [9]:
```
grafico4 = plt.plot([x**2 for x in corpoFinito4])
plt.show(grafico4)
```



Determinar elementos primitivos de Corpos Finitos

Neste exercício c) teremos de determinar elementos primitivos dos corpos finitos primos anteriormente definidos e verificar a seguinte preposição:

> *Para todo g primitivo de Fp e para todo exponente n, verifica-se que g^n = 1 se e só se n = 0 mod(p-1)*

Para tal iremos usufruir das funções do SageMath que nos fornece a função `primitive_element()` para encontrar um elemento primitivo de um corpo finito. Após isso iremos calcular o n através da função `mod(0,p-1)` e por fim iremos verificar se o primitivo g eleavado a n é igual a 1 (`g^n = 1`).

In [32]:
```
g1 = corpoFinito1.primitive_element()
print(g1)
n1 = mod(0,p1-1)
print(n1)
result1 = g1^n1 == 1
print(result1)
```

```
3
0
True
```

In [28]:
```
g2 = corpoFinito2.primitive_element()
print(g2)
n2 = mod(0,p2-1)
print(n2)
result2 = g2^n == 1
print(result2)
```

```
Corpo Finito 2:
3
0
True
```

In [29]:
```
g3 = corpoFinito3.primitive_element()
print(g3)
n3 = mod(0,p3-1)
print(n3)
result3 = g3^n3 == 1
print(result3)
```

```
Corpo Finito 3:
17
0
True
```

In [31]:
```
g4 = corpoFinito4.primitive_element()
print(g4)
n4 = mod(0,p4-1)
print(n4)
result4 = g4^n == 1
print(result4)
```

```
3
0
True
```