Practica Matplotlib

Nombre: Jesus Alejandro Valencia Valadez

In [4]:
```
import pandas as pd
```

In [5]:
```
df = pd.read_csv("pokemon.csv")
df.head()
```

Out[5]:

	abilities	against_bug	against_dark	against_dragon	against_electric	against_fairy	against
0	['Overgrow', 'Chlorophyll']	1.0	1.0	1.0	0.5	0.5	
1	['Overgrow', 'Chlorophyll']	1.0	1.0	1.0	0.5	0.5	
2	['Overgrow', 'Chlorophyll']	1.0	1.0	1.0	0.5	0.5	
3	['Blaze', 'Solar Power']	0.5	1.0	1.0	1.0	0.5	
4	['Blaze', 'Solar Power']	0.5	1.0	1.0	1.0	0.5	

5 rows × 41 columns

In [6]:
```
columnas = list(df.columns) # Traer todas las columnas a una variable, df con r
columnas
```

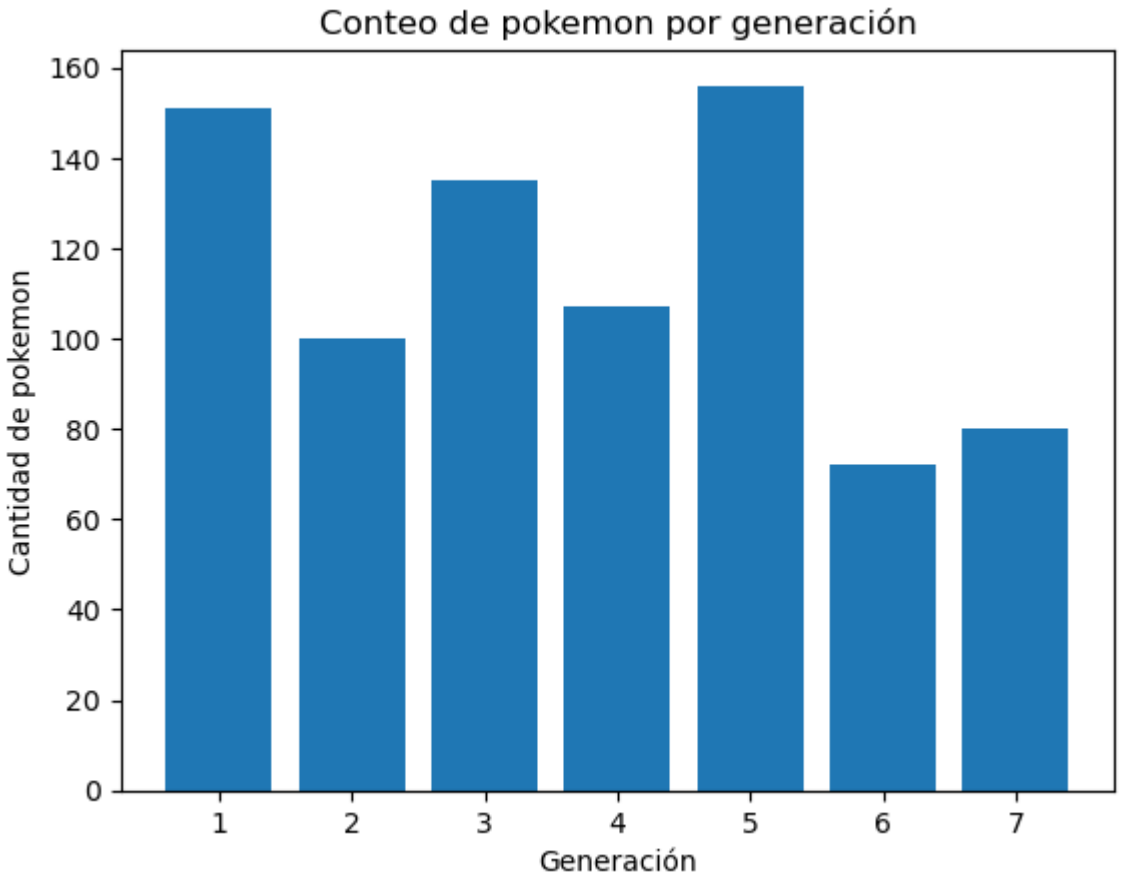
Out[6]:
```
['abilities',
 'against_bug',
 'against_dark',
 'against_dragon',
 'against_electric',
 'against_fairy',
 'against_fight',
 'against_fire',
 'against_flying',
 'against_ghost',
 'against_grass',
 'against_ground',
 'against_ice',
 'against_normal',
 'against_poison',
 'against_psychic',
 'against_rock',
 'against_steel',
 'against_water',
 'attack',
 'base_egg_steps',
 'base_happiness',
 'base_total',
 'capture_rate',
 'classfication',
 'defense',
 'experience_growth',
 'height_m',
 'hp',
 'japanese_name',
 'name',
 'percentage_male',
 'pokedex_number',
 'sp_attack',
 'sp_defense',
 'speed',
 'type1',
 'type2',
 'weight_kg',
 'generation',
 'is_legendary']
```

Realizar una visualización por ejercicio que respondan cada cuestionamiento:

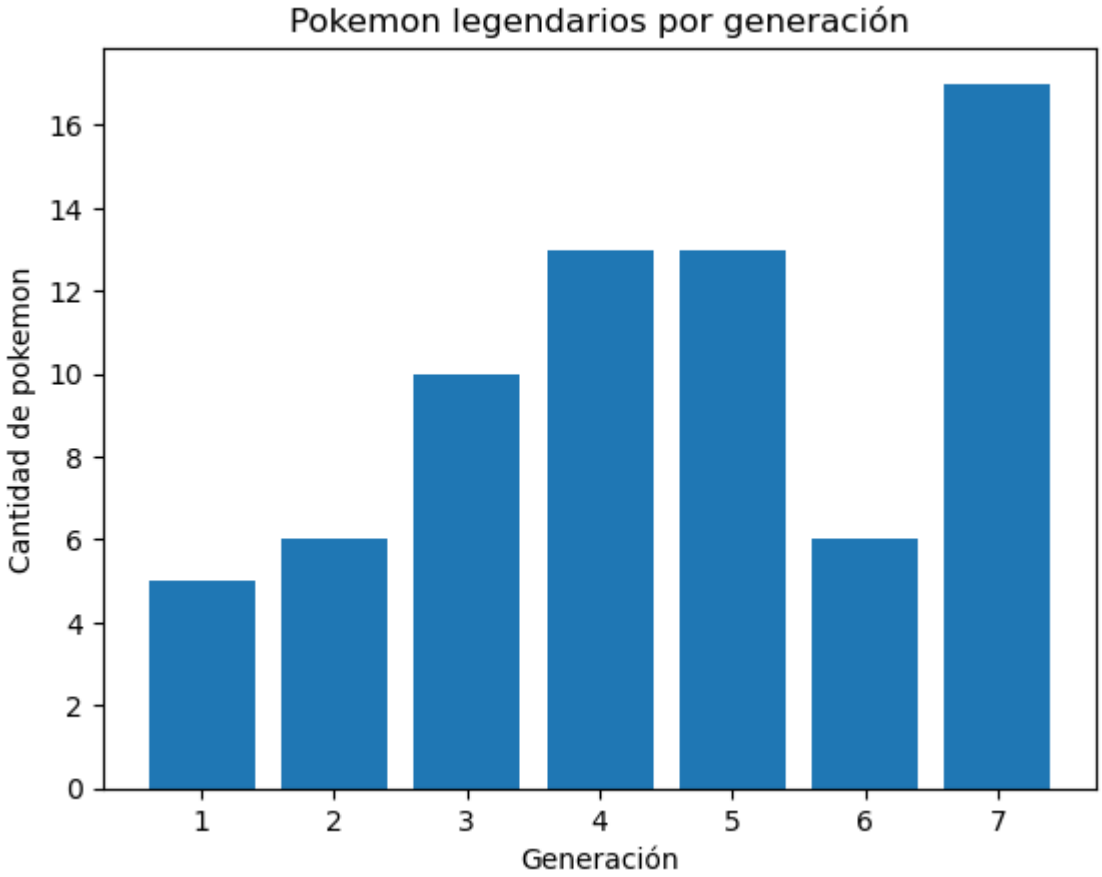
1. ¿Cuántos nuevos pokemon hay por generación?
2. ¿Cuántos pokemon legendarios hay por generación?
3. Visualización por peso y altura.
4. ¿Cuál es el tipo más común de pokemon?
5. ¿Cuáles son las combinaciones de tipo más comunes?
6. ¿Qué pokemon son los mejores en términos de sus stats? (attack, defense, sp_attack, sp_defense, speed, hp)
7. ¿Cómo cambiaron los stats en promedio conforme avanzaban las generaciones?

8. ¿Un pokemon es más fuerte entre más difícil es de capturar?

In [10]:
```
import matplotlib.pyplot as plt
# Ejercicio 1. ¿Cuántos nuevos pokemon hay por generación?
df2 = df[ ["name","generation"] ].groupby("generation").count() # Cuento todas

plt.bar(df2.index, df2["name"])
plt.title("Conteo de pokemon por generación")
plt.xlabel("Generación")
plt.ylabel("Cantidad de pokemon")
plt.show()
```



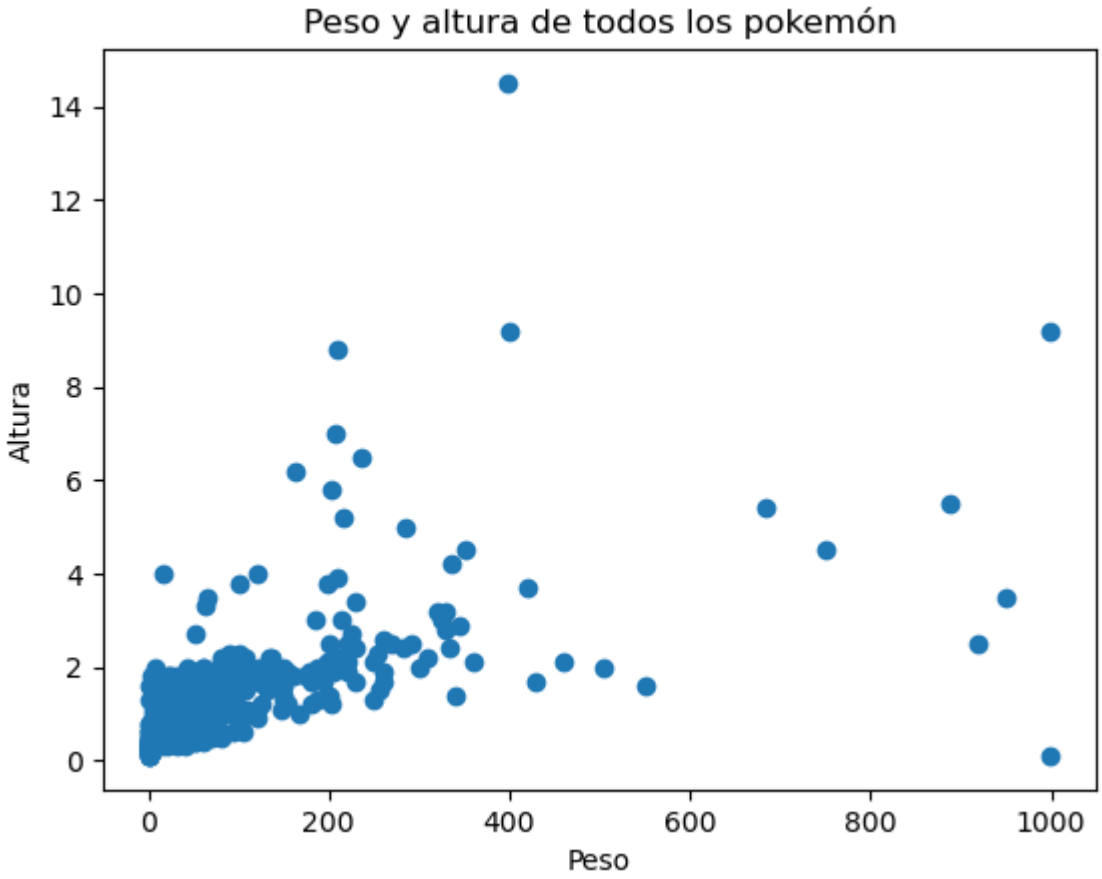
In [5]:
```
# Ejercicio 2. ¿Cuántos pokemon legendarios hay por generación?
df3 = df[ df["is_legendary"] == 1][ ["name", "generation", "is_legendary"] ].g

plt.bar(df3.index, df3["name"])
plt.title("Pokemon legendarios por generación")
plt.xlabel("Generación")
plt.ylabel("Cantidad de pokemon")
plt.show()
df3
```



Out[5]:

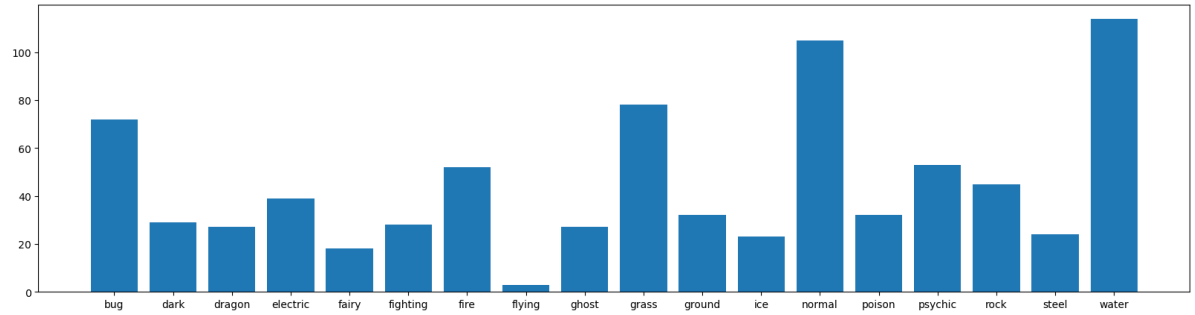
	name	is_legendary
generation		
1	5	5
2	6	6
3	10	10
4	13	13
5	13	13
6	6	6
7	17	17

In [6]:
```
# Ejercicio 3. Visualización por peso y altura.
df4 = df[ ["weight_kg", "height_m"] ]

plt.scatter(df4["weight_kg"], df4["height_m"])
plt.title("Peso y altura de todos los pokemón")
plt.xlabel("Peso")
plt.ylabel("Altura")
plt.show()
```



Ejercicio 4. ¿Cuál es el tipo más común de pokemon?

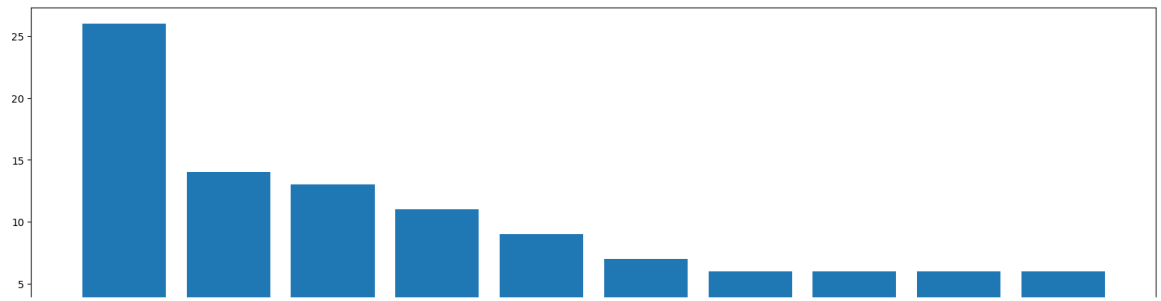
In [7]:
```
df5 = df[ ["name","type1","generation"] ].groupby("type1").count() # Contar
df5

plt.figure(figsize=(20,5))
plt.bar(df5.index, df5["name"])
plt.show()
```



Ejercicio 5. ¿Cuáles son las combinaciones de tipo más comunes?

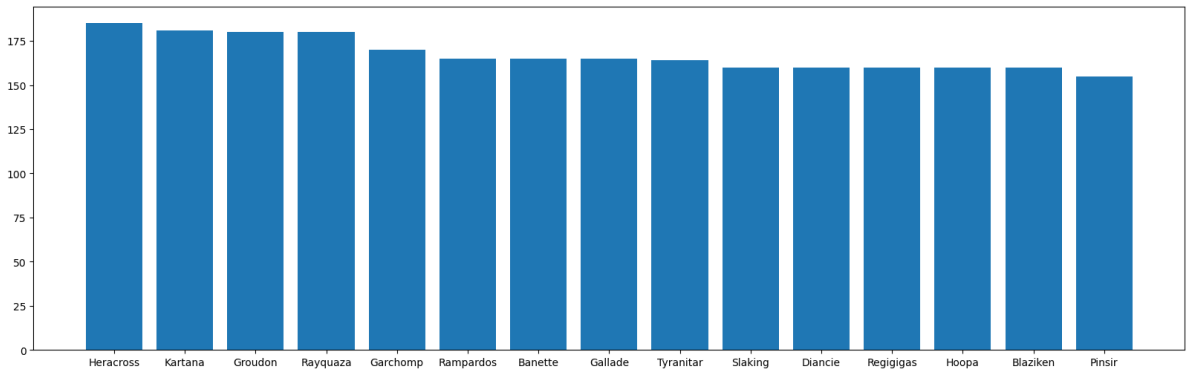
In [8]:
```
df6 = df[ ~df["type2"].isna() ]

df6.loc[:, "type"] = df6.loc[:, "type1"] + "-" + df6.loc[:, "type2"]
df7 = df6[["name", "type"]].groupby("type").count().sort_values("name", ascend

plt.figure(figsize=(20,6))
plt.bar(df7.index[:10], df7["name"][:10])
plt.show()
```

```
C:\Users\USER\AppData\Local\Temp\ipykernel_6076\3751828165.py:3: SettingWit
hCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-doc
s/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://p
andas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-vi
ew-versus-a-copy)
  df6.loc[:, "type"] = df6.loc[:, "type1"] + "-" + df6.loc[:, "type2"]
```



Ejercicio 6. ¿Qué pokemon son los mejores en términos de sus stats? (attack, defense, sp_attack, sp_defense, speed, hp)

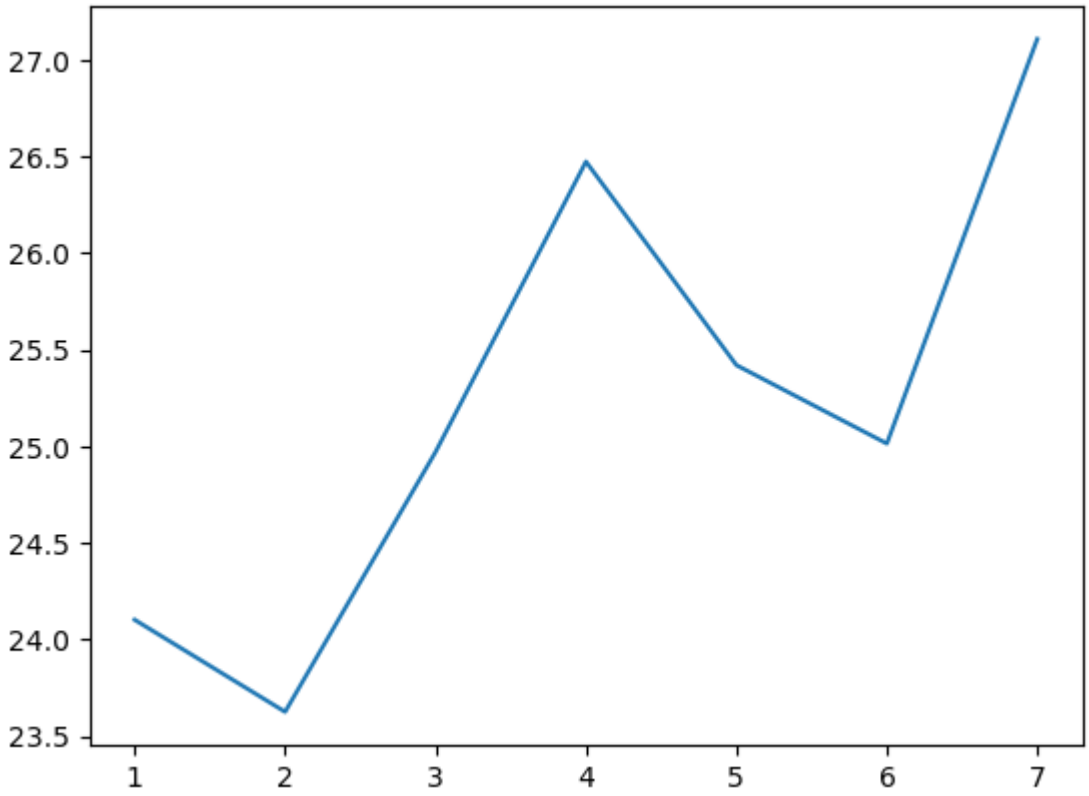
In [9]:
```
df8 = df[ ["name", "attack"]].sort_values("attack", ascending = False)
plt.figure(figsize=(20,6))


plt.bar(df8["name"][:15], df8["attack"][:15])
plt.show()
```



Ejercicio 7

In [10]:
```
def stats_prom(fila):
    return (fila["attack"] + fila["defense"]) / 6
df["stats_prom"] = df.apply(stats_prom, axis=1)
```

In [11]:
```
df2 = df[ ["stats_prom", "generation"]].groupby("generation").mean()
plt.plot( df2.index, df2["stats_prom"])
plt.show()
```



Ejercicio 8.

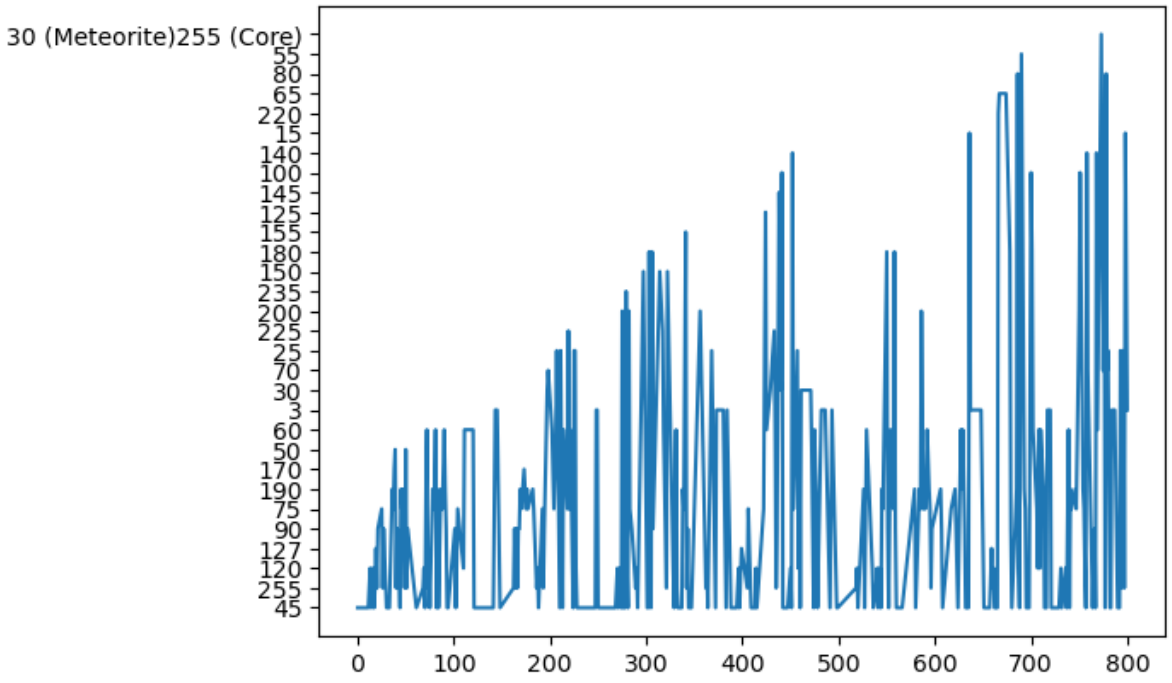
In [12]:
```
df["capture_rate"].values
```

Out[12]:
```
array(['45', '45', '45', '45', '45', '45', '45', '45', '45', '255', '120',
       '45', '255', '120', '45', '255', '120', '45', '255', '127', '255',
       '90', '255', '90', '190', '75', '255', '90', '235', '120', '45',
       '235', '120', '45', '150', '25', '190', '75', '170', '50', '255',
       '90', '255', '120', '45', '190', '75', '190', '75', '255', '50',
       '255', '90', '190', '75', '190', '75', '190', '75', '255', '120',
       '45', '200', '100', '50', '180', '90', '45', '255', '120', '45',
       '190', '60', '255', '120', '45', '190', '60', '190', '75', '190',
       '60', '45', '190', '45', '190', '75', '190', '75', '190', '60',
       '190', '90', '45', '45', '190', '75', '225', '60', '190', '60',
       '90', '45', '190', '75', '45', '45', '45', '190', '60', '120',
       '60', '30', '45', '45', '225', '75', '225', '60', '225', '60',
       '45', '45', '45', '45', '45', '45', '45', '255', '45', '45', '35',
       '45', '45', '45', '45', '45', '45', '45', '45', '45', '45', '25',
       '3', '3', '3', '45', '45', '45', '3', '45', '45', '45', '45', '45',
       '45', '45', '45', '45', '45', '255', '90', '255', '90', '255',
       '90', '255', '90', '90', '190', '75', '190', '150', '170', '190',
       '75', '190', '75', '235', '120', '45', '45', '190', '75', '65',
       '45', '255', '120', '45', '45', '235', '120', '75', '255', '90',
       '45', '45', '30', '70', '45', '225', '45', '60', '190', '75',
       '190', '60', '25', '190', '75', '45', '25', '190', '45', '60',
       '120', '60', '190', '75', '225', '75', '60', '190', '75', '45',
       '25', '25', '120', '45', '45', '120', '60', '45', '45', '45', '75',
       '45', '45', '45', '45', '45', '30', '3', '3', '3', '45', '45',
       '45', '3', '3', '45', '45', '45', '45', '45', '45', '45', '45',
       '45', '45', '255', '127', '255', '90', '255', '120', '45', '120',
       '45', '255', '120', '45', '255', '120', '45', '200', '45', '190',
       '45', '235', '120', '45', '200', '75', '255', '90', '255', '120',
       '45', '255', '120', '45', '190', '120', '45', '180', '200', '150',
       '255', '255', '60', '45', '45', '180', '90', '45', '180', '90',
       '120', '45', '200', '200', '150', '150', '150', '225', '75', '225',
       '60', '125', '60', '255', '150', '90', '255', '60', '255', '255',
       '120', '45', '190', '60', '255', '45', '90', '90', '45', '45',
       '190', '75', '205', '155', '255', '90', '45', '45', '45', '45',
       '255', '60', '45', '200', '225', '45', '190', '90', '200', '45',
       '30', '125', '190', '75', '255', '120', '45', '255', '60', '60',
       '25', '225', '45', '45', '45', '3', '3', '3', '3', '3', '3', '3',
       '3', '3', '3', '45', '3', '3', '45', '45', '45', '45', '45', '45',
       '45', '45', '45', '255', '120', '45', '255', '127', '255', '45',
       '235', '120', '45', '255', '75', '45', '45', '45', '45', '120',
       '45', '45', '120', '45', '200', '190', '75', '190', '75', '190',
       '75', '45', '125', '60', '190', '60', '45', '30', '190', '75',
       '120', '225', '60', '255', '90', '255', '145', '130', '30', '100',
       '45', '45', '45', '50', '75', '45', '140', '60', '120', '45',
       '140', '75', '200', '190', '75', '25', '120', '60', '45', '30',
       '30', '30', '30', '30', '30', '30', '30', '45', '45', '30', '50',
       '30', '45', '60', '45', '75', '45', '3', '3', '3', '3', '3', '3',
       '3', '3', '3', '30', '3', '3', '45', '3', '3', '45', '45', '45',
       '45', '45', '45', '45', '45', '45', '255', '255', '255', '120',
       '45', '255', '90', '190', '75', '190', '75', '190', '75', '190',
       '75', '255', '120', '45', '190', '75', '255', '120', '45', '190',
       '45', '120', '60', '255', '180', '90', '45', '255', '120', '45',
       '45', '45', '255', '120', '45', '255', '120', '45', '190', '75',
       '190', '75', '25', '180', '90', '45', '120', '60', '255', '190',
       '75', '180', '90', '45', '190', '90', '45', '45', '45', '45',
       '190', '60', '75', '45', '255', '60', '200', '100', '50', '200',
       '100', '50', '190', '45', '255', '120', '45', '190', '75', '200',
```

```
'200', '75', '190', '75', '190', '60', '75', '190', '75', '255',
'90', '130', '60', '30', '190', '60', '30', '255', '90', '190',
'90', '45', '75', '60', '45', '120', '60', '25', '200', '75', '75',
'180', '45', '45', '190', '90', '120', '45', '45', '190', '60',
'190', '60', '90', '90', '45', '45', '45', '45', '15', '3', '3',
'3', '3', '3', '3', '3', '3', '3', '3', '3', '3', '45', '45', '45',
'45', '45', '45', '45', '45', '45', '255', '127', '255', '120',
'45', '255', '120', '45', '220', '65', '225', '120', '45', '200',
'45', '220', '65', '160', '190', '75', '180', '90', '45', '200',
'140', '200', '140', '190', '80', '120', '45', '225', '55', '225',
'55', '190', '75', '45', '45', '45', '45', '45', '100', '180',
'60', '45', '45', '45', '75', '120', '60', '120', '60', '190',
'55', '190', '45', '45', '45', '3', '3', '3', '3', '45', '45',
'45', '45', '45', '45', '45', '45', '45', '255', '120', '45',
'255', '127', '255', '120', '45', '225', '60', '45', '190', '75',
'190', '90', '60', '190', '75', '190', '60', '200', '100', '190',
'75', '190', '75', '120', '45', '140', '70', '235', '120', '45',
'60', '45', '45', '90', '45', '140', '60', '60', '3', '3',
'30 (Meteorite)255 (Core)', '45', '70', '180', '45', '80', '70',
'25', '45', '45', '45', '3', '3', '3', '3', '45', '45', '45', '45',
'45', '25', '255', '30', '25', '255', '15', '3', '3'], dtype=object)
```

In [16]:
```
df6 = df[ df["capture_rate"] == '30 (Meteorite)255 (Core)']["name"]
df.at[773,"capture_rate"] = 30
```

In [19]:
```
df.loc[773][["name","capture_rate"]]

plt.plot( df6.index, df6["capture_rate"])
plt.show()
```



In [ ]:
```

```