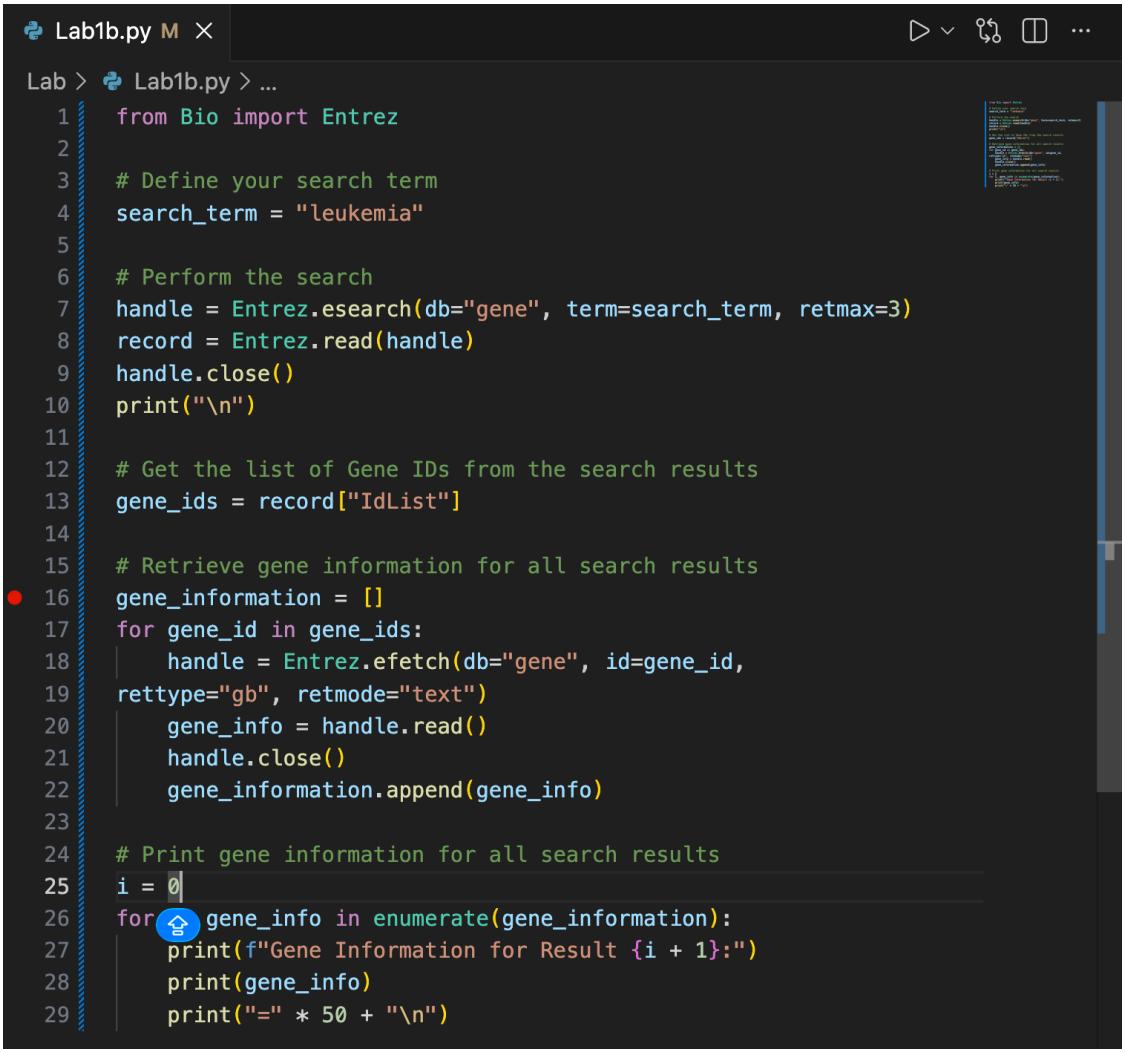
PART B

1.

a. I have changed the **`search_term`** to "**`leukemia`**" and then to "**`cystic fibrosis`**" as the topic of interest.



b. After repeating step 1 twice, I found that different diseases (**`search_term`**) return different genes in different species. Each search focuses on genes related to that specific disease condition. Another interesting observation is the genes returned for **`search_term = "cystic fibrosis"`**: EGFR from humans do has some connection to leukemia (the first **`search_term`**).

```
Gene Information for Result 1:

1. LOC144531885
leukemia inhibitory factor receptor-like [Sander vitreus (walleye)]
Other Designations: leukemia inhibitory factor receptor-like
Chromosome: 16
Annotation: Chromosome 16 NC_135870.1 (28581425..28602849)
ID: 144531885


================================================

Gene Information for Result 2:

1. pbx4
pre-B-cell leukemia transcription factor 4 [Sander vitreus (walleye)]
Other Designations: pre-B-cell leukemia transcription factor 4
Chromosome: 15
Annotation: Chromosome 15 NC_135869.1 (16032391..16060199, complement)
ID: 144530797


================================================

Gene Information for Result 3:

1. pbxip1a
pre-B-cell leukemia homeobox interacting protein 1a [Sander vitreus (walleye)]
Other Designations: pre-B-cell leukemia transcription factor-interacting protein 1
Chromosome: 14
Annotation: Chromosome 14 NC_135868.1 (11971962..11982191)
ID: 144528519
```

```
Gene Information for Result 1:

1. cftr
CF transmembrane conductance regulator [Sander vitreus (walleye)]
Other Designations: cystic fibrosis transmembrane conductance regulator
Chromosome: 8
Annotation: Chromosome 8 NC_135862.1 (5473012..5522248, complement)
ID: 144521857


================================================

Gene Information for Result 2:

1. cftr
CF transmembrane conductance regulator [Mustelus asterias (starry smooth-hound)]
Other Designations: cystic fibrosis transmembrane conductance regulator
Chromosome: 9
Annotation: Chromosome 9 NC_135809.1 (27249470..27379241, complement)
ID: 144498565


================================================

Gene Information for Result 3:

1. EGFR
Official Symbol: EGFR and Name: epidermal growth factor receptor [Homo sapiens (human)]
Other Aliases: ERBB, ERBB1, ERRP, HER1, NISBD2, NNCIS, PIG61, mENA
Other Designations: epidermal growth factor receptor; EGFR vIII; avian erythroblastic leuk
emia viral (v-erb-b) oncogene homolog; cell growth inhibiting protein 40; cell proliferati
on-inducing protein 61; epidermal growth factor receptor tyrosine kinase domain; erb-b2 re
ceptor tyrosine kinase 1; proto-oncogene c-ErbB-1; receptor tyrosine-protein kinase erbB-1
Chromosome: 7; Location: 7p11.2
Annotation: Chromosome 7 NC_000007.14 (55019017..55211628)
MIM: 131550
ID: 1956
```

c. Codes:

Lab1b.py M

Lab > Lab1b.py > ...

```python
from Bio import Entrez
import pandas as pd
import re

# Define your search term
search_term = "cystic fibrosis"

# Perform the search
handle = Entrez.esearch(db="gene", term=search_term, retmax=3)
record = Entrez.read(handle)
handle.close()
print("\n")

# Get the list of Gene IDs from the search results
gene_ids = record["IdList"]

# Retrieve gene information for all search results
gene_information = []
for gene_id in gene_ids:
    handle = Entrez.efetch(db="gene", id=gene_id,
rettype="gb", retmode="text")
    gene_info = handle.read()
    handle.close()
    gene_information.append(gene_info)

# Extract gene symbol, other aliases, and ID
table_data = []
for gene_info in gene_information:
    # Extract gene symbol
    symbol_match = re.search(r'^\d+\.\s+(\S+)', gene_info, re.MULTILINE)
    symbol = symbol_match.group(1) if symbol_match else "N/A"

    # Extract other aliases
    aliases_match = re.search(r'Other Aliases:\s*(.+?)(?:\n|$)', gene_info)
    aliases = aliases_match.group(1).strip() if aliases_match else "N/A"

    # Extract ID
    id_match = re.search(r'ID:\s*(\d+)', gene_info)
    gene_id = id_match.group(1) if id_match else "N/A"

    table_data.append([symbol, aliases, gene_id])

# Create and display pandas DataFrame
df = pd.DataFrame(table_data, columns=["Gene Symbol", "Other Aliases", "ID"])
print(f"Gene search results for: {search_term}\n")
print(df.to_string(index=False))
```

Output:

```
Gene search results for: cystic fibrosis

Gene Symbol                                    Other Aliases        ID
       cftr                                              N/A 144521857
       cftr                                              N/A 144498565
       EGFR ERBB, ERBB1, ERRP, HER1, NISBD2, NNCIS, PIG61, mENA      1956
```

2. Codes:

Lab1(2).py U

Lab > Lab1(2).py > ...

```python
from sklearn.datasets import load_iris
from sklearn.model_selection import StratifiedKFold
import pandas as pd

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target

# Display dataset size before cross-validation
print(f"Total samples: {X.shape[0]}")
print(f"Number of features: {X.shape[1]}")
print(f"Number of classes: {len(set(y))}")
print(f"Dataset dimensions: {X.shape}\n")
print("=" * 70 + "\n")

# Create Stratified K-Fold with 10 splits
skf = StratifiedKFold(n_splits=10, shuffle=True, random_state=42)

# Perform stratified k-fold cross-validation
fold_results = []
for fold, (train_index, test_index) in enumerate(skf.split(X, y), 1):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

    train_size = X_train.shape[0]
    test_size = X_test.shape[0]

    fold_results.append({
        'Fold': fold,
        'Training Set Size': train_size,
        'Test Set Size': test_size,
        'Total': train_size + test_size
    })

# Create a pandas DataFrame
df = pd.DataFrame(fold_results)
print(df.to_string(index=False))
```

Output:

```
Total samples: 150
Number of features: 4
Number of classes: 3
Dataset dimensions: (150, 4)

======================================================================

 Fold  Training Set Size  Test Set Size  Total
    1                135             15    150
    2                135             15    150
    3                135             15    150
    4                135             15    150
    5                135             15    150
    6                135             15    150
    7                135             15    150
    8                135             15    150
    9                135             15    150
   10                135             15    150
```