SECB3203 PROGRAMMING FOR BIOINFORMATICS – Lab 1 (3%)
Due: 14 Nov (Friday) 11.59 PM MYT
Note:

- **Filename: L1_Name.pdf** (e.g., *L1_Ali.pdf*)

PART A - AWS Cloud Foundation / Microsoft Learn (1%)

Draw and Organize Your AWS Architecture Diagrams (1%):

Scenario: Let's consider a scenario where you are operating an e-commerce website and are looking to transition a portion of your IT infrastructure to the AWS cloud. You currently manage an on-premises data center that hosts your customer database, and you intend to migrate this database and related services to AWS. AWS services employed support the migration of your e-commerce infrastructure to the cloud, providing scalability, robust availability, and advanced security features. The AWS-hosted website seamlessly interacts with your customer database, resulting in a dependable and user-friendly e-commerce platform. Make sure you need to include the AWS services from the categories of **AWS Cloud Security, Networking and Content Delivery, Compute, Storage, and Database**.

Reference: https://cloudockit.medium.com/5-tips-for-drawing-organizing-your-aws-architecture-diagrams-1bf1e9d84fd1

PART B - Coding in Python (2%)

1. **Search the Gene database with general text queries will produce the most relevant results through NCBI (using the *biopython* package) and the partial codes provided.**

   Reference: https://www.ncbi.nlm.nih.gov/gene/

   Print screen your complete codes and output only.

Instructions:

1. Please change the search_term to any disease or more specific cancer type.
2. Repeat the step 1 twice and summarize the findings you obtained.
3. Tabulate your answers for gene symbol, other aliases, and ID only.

```
!pip install biopython

from Bio import Entrez

# Define your search term
search_term = "cancer"  # Modify this to your desired
query

# Perform the search
handle = Entrez.esearch(db="gene", term=search_term,
retmax=3)  # You can adjust retmax for the number of
results you want
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)

# Print gene information for all search results
for i, gene_info in enumerate(gene_information):
    print(f"Gene Information for Result {i + 1}:")
```

```
    print(gene_info)
    print("=" * 50 + "\n")
```

2. **To load the Iris dataset, display its size before and after performing stratified ten-fold cross-validation in Python, you can use the scikit-learn library.** Scikit-learn provides a convenient way to work with datasets, perform cross-validation, and assess their sizes.

   Given: The dataset is iris (*from sklearn.datasets import load_iris*).

   Return: The size of training and test sets based on stratified ten-fold cross-validation (*from sklearn.model_selection import StratifiedKFold*)

   Requirement: *pip install scikit-learn*

   Reference:

   https://scikit-learn.org/stable/tutorial/index.html

   https://towardsdatascience.com/how-to-train-test-split-kfold-vs-stratifiedkfold-281767b93869

   Print screen your complete codes and output.