**Assignment 3**
**Malware Detection using ELF features**

**Setup**

For the first 2 questions, use the "**Hello_ELF**" file given in the "**ELF_data.zip**" and the link for the data is given below**.**

https://drive.google.com/file/d/1oL_PRwRcR250i36FpELGF3DFd8NKPJLv/view?usp=sharing

Since the dataset contains malware executables, don't try to open the samples in the dataset. Set up an Ubuntu virtual machine (higher than 16.04). There are 3 subparts to this assignment and the solution for the first 2 questions should be in a separate python file with the name **"Solution_x.py"** where x denotes the question number. You can use the "**pyelftools**" utility to perform the tasks.

**Question 1**

Write a python code to extract the below information

- Segments with different sizes in file and memory and then, print the sections mapped to that segment.
- Sections not loaded in the memory

**Question 2**

Write a python code to

- print the first 10 bytes in the **".text"** section starting from the entry point address.
- count the number of global, local and weak symbols in the "**.dynsym**" section of type **STT_FUNC**.

**Question 3**

Extract the fields from the Executable and Linkable Format (ELF) format and build your own malware detection system. Solution for this question should be in a folder named **"Solution_3"** and the individual python/notebook files should have a meaningful name.

Perform the below tasks to implement the malware detection model.

1. Use the dataset given in the folder **"ELF_Dataset"** in the file **"ELF_data.zip".** The dataset consists of 2 folders 'Malware' with 500 samples and 'Benignware' with 500 samples.

2. Extract static features from each benign and malicious executable. Assign the label as '1' for malware and '0' for benignware.

3. Do research to design good features that will help your machine learning system make accurate inferences. You can start by extracting as many features from the ELF as possible. Then, apply a standard feature selection algorithm of your choice to reduce the number of features.

3. Split the dataset into non-overlapping training and test sets, in which the training set consists of 70% of the data (an arbitrarily chosen proportion) and the test set consists of the remaining 30%.

4. Train the machine learning classifiers (your choice. You can try different models and select the best one) to recognize malware using the features you have extracted. Use the inbuilt functions from "***sklearn***" for implementing the models and calculating metrics.

5. Test your model and calculate the prediction accuracy, precision, recall, false-positive rate, F1-score and confusion matrix.