Program 1: Network traffic monitoring is important to detect attacks in the network. This program will take a pcap file as input and extract the following attributes from each packet:

A: Different port numbers with their occurrence frequencies

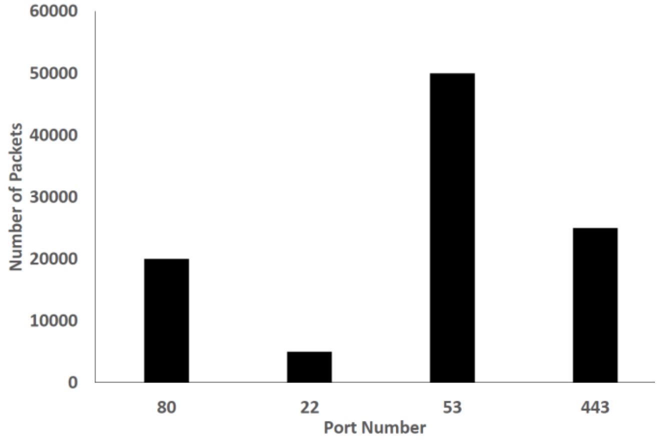
B: Different IP addresses with their occurrence frequencies

C: Different MAC addresses with their occurrence frequencies

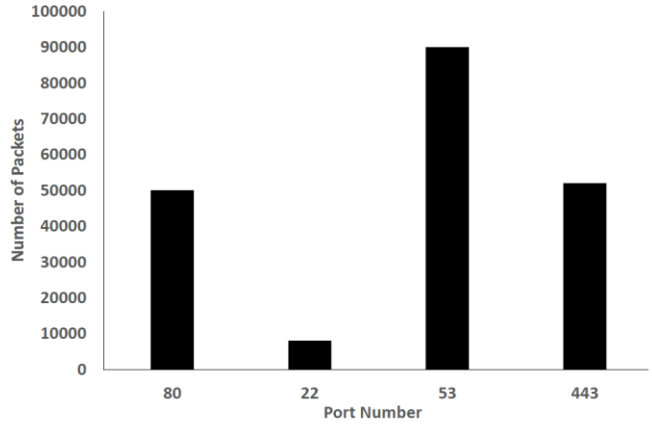
Plot the three graphs for each of the above attributes. After each iteration of 10,000 packets, the graph should modify with the cumulative values.

Example Input: test.pcap (having 1,00,000 packets)

Example Output: Plot for first 10,000 packets for port numbers i.e. A (Hold this graph on screen for 10 seconds)



After processing next 1,00,000 packets (Hold this graph on screen for 10 seconds):



And so on till 1,00,000 packets.

The example is shown for port numbers only. You need to plot three graphs at a time i.e. for all three attributes. Only one program MUST do all the thing.

Note: You can use internet, books whatever you need. If code is found plagiarised with other student, then it will deduct the marks equivalent to plagiarism (if 90% plagiarism found between any two students then 9 marks will be dedusted out of 10 marks for both of them i.e. even if full program is correct then also you will get 1 mark). Remember code plagiarism checking rules are different from text plagiarism checking rules.

Program 2: Zero-day attacks are those attacks which are launched for the very first time on the network. One of the methods to detect zero-day attacks is by payload analysis of network packets. In this method, first payload of attack free network traffic is extracted and then 'n-grams' are generated from that payload. These n-grams are stored as attack free n-grams. When a test packet comes, n-grams are extracted from the payload of test packet (in the same way, extraction happened for attack free packets) and matched with attack free n-grams (which we created from attack free packets). For that packet there may be 'm' n-grams generated from its payload among which 'k' n-grams do not match with attack free n-grams (say malicious n-grams). Maliciousness of test packet is calculated using (k/m * 100). If the maliciousness is more than X% then report that packet is malicious.

Example Input: AttackFree.pcap, MixPackets.pcap, value of 'n' for n-gram = 3, X = 20%

Working:

```
Empty Storage (Say a text file)

For each packet in AttackFree.pcap do:

    Extract payload P (Example Payload P = Hello World)

    Extract 3-gram = Hel, ell, llo, o<space>W, <space>Wo, orl, rld

    Store these 3-grams in Storage (can be a text file). Remove duplicate n-grams from Storage

Finish all packets
```

Attack free n-grams are now stored from AttackFree.pcap. Now evaluate each packet in MixPackets.pcap and find how many packets are normal and how many are malicious

Working:

```
For each packet in MixPackets.pcap do:

    Extract payload P (Example Payload P = Hello India)

    Extract 3-gram = Hel, ell, llo, o<space>I, <space>In, ndi, dia

    For each n-gram of that packet do:

        If n-gram exist in Storage

            Normal ngram Count ++

        Else

            Anomalous ngram Count ++

    Calculate Maliciousness = ((Anomalous ngram Count)/(Anomalous ngram Count + Normal ngram Count))*100

    If Maliciousness < X

    Print Maliciousness and Print Packet is Normal

    Else

    Print Maliciousness and Print Packet is Anomalous
```

Note: You can use internet, books whatever you need. If code is found plagiarised with other student, then it will deduct the marks equivalent to plagiarism (if 90% plagiarism found between any two students then 9 marks will be dedusted out of 10 marks for both of them i.e. even if full program is correct then also you will get 1 mark). Remember code plagiarism checking rules are different from text plagiarism checking rules.

Student Name	Email ID	Roll Number	Assigned Program
Srishtee Kriti	sk4189@bennett.edu.in	E17CSE011	1
Monisha gali	gm1611@bennett.edu.in	E17CSE017	2
Boddeda jaya sai avinash	ba5314@bennett.edu.in	E17CSE018	1
MADHU KIRAN AKKIREDDY	mr1283@bennett.edu.in	E17CSE023	2
Shanmukha Sai Sumanth Yenneti	SY9438@bennett.edu.in	E17CSE024	1
Lekhana	gl8316@bennett.edu.in	E17CSE026	2
Nitheshwar C R	nr3702@bennett.edu.in	E17CSE045	1
Amit Kumar	ak1809@bennett.edu.in	E17CSE048	2
G.N.S.Abhiram	ga6662@bennett.edu.in	E17CSE050	1
Shreyas Papinwar	Sp3298@bennett.edu.in	E17CSE067	2
Pratheek yatham	yr7884@bennett.edu.in	E17CSE068	1
Harsh Kataria	hk9663@bennett.edu.in	E17CSE071	2
Kusumanth Gali	gr6331@bennett.edu.in	E17CSE080	1
Sourabh Chawala	Sc6465@bennett.edu.in	E17CSE083	2
Sai Prakash	jp2986@bennett.edu.in	E17CSE104	1
Akshita Mehta	am7799@bennett.edu.in	E17CSE117	2
Souvik Mishra	sm5939@bennett.edu.in	E17CSE127	1
Palacharla Bhaskar Praveen	PP3400@bennett.edu.in	E17CSE131	2
Varun Sharma	vs8950@bennett.edu.in	E17CSE137	1
Zubin choudhary	zc7326@bennett.edu.in	E17CSE150	2
Naman Bansal	nb3319@bennett.edu.in	E17CSE179	1
T.GIRIDHAR	tg1449@bennett.edu.in	E17CSE185	2
Bharat Ahuja	ba8365@bennett.edu.in	E17CSE189	1