1. If we add a signed integer and an unsigned integer, will the result be signed or unsigned? Justify your answer with an example program in C/CPP?

Answer is unsigned integer

unsigned integer + signed integer = unsigned integer.

```
#include<iostream.h>
#include<conio.h>
int main() {
  unsigned int a=10000;
  int b=2147483647 ;
  cout<<a+b;
  getch();
  return 0;
}
```

2. VIT-AP has a very good server facility to serve students. All the servers are developed in Data Centre (DC). DC gets suffered daily due to internal cyber-attacks which leads to the increased downtime of the server. Write your own security design principles to protect SDC from future attacks.

https://www.websitemagazine.com/blog/7-tips-for-minimizing-website-downtime

https://www.vxchnge.com/blog/minimize-server-downtime-best-practices

3. Summarize endianness? Examine the Endianness of Intel Xeon processor.

Endianness refers to the sequential order in which bytes are arranged into larger numerical values when stored in memory or when transmitted over digital links.

If one computer reads bytes from left to right, and other computer reads from right to left, we're going to have issues when we need to communicate.

Endianness is represented two ways **Big-endian** (**BE**) and **Little-endian** (**LE**).

- **BE** stores the **big-end** first. When reading multiple bytes, the first byte (or the lowest memory address) is the biggest - so it makes the most sense to people who read left to right.
- **LE** stores the **little-end** first. When reading multiple bytes, the first byte (or the lowest memory address) is the littlest - so it makes most sense to people who read right to left.

```
#include <bits/stdc++.h>
using namespace std;
int main() {
	unsigned int i = 1;
	char *c = (char*)&i;
	if (*c)
		cout<<"Little endian";
	else
		cout<<"Big endian";
	return 0;
}
```

No, All Intel/AMD/Modern (maximum) CPUs are little endian. IA32 bit & IA64bit Xeon processor are also little endian.

4. Design a risk mitigation framework for a new e-commerce site "eVIT.com". Identify the business & technical risk and prioritize the potential risk.

http://www.unibulmerchantservices.com/ecommerce-risk-management-guide/

(refer the pdf I sent about RMF)

5. As the matter of the fact VIT-AP has a very good infrastructure. As a security engineer you are asked to build a secure network design for VITAP. Hence plan and construct a defence in depth design principle for VITAP infrastructure. Note: The design approach which you are going to construct should focus on the both, system and network.

http://etutorials.org/Networking/Cisco
https://www.learncisco.net/courses/iins/common-security-threats/

6. State the default system path of 16/32/64-bit binaries in windows.
https://stackoverflow.com/questions/949959/
Compare and contrast attack vector and attack surface. Categorize attack surface in detail.

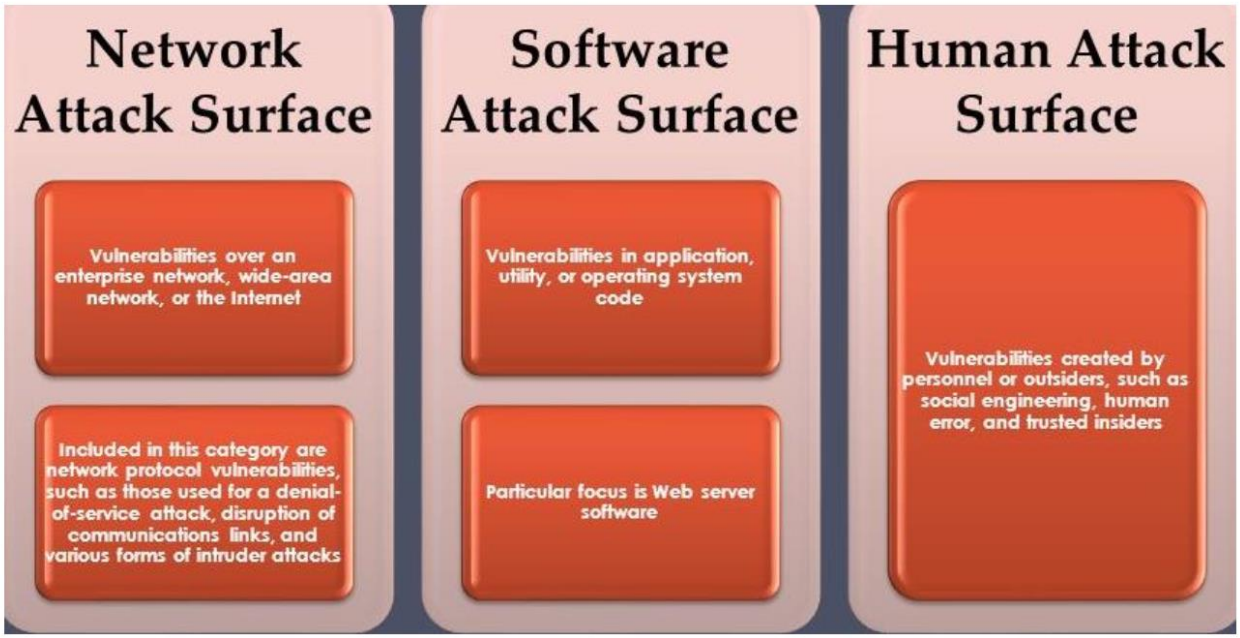
Attack vector

- An **Attack vector** is a path or means by which a hacker (or cracker) can gain access to a computer or network server in order to deliver a payload or malicious outcome.
- **Attack vectors** enable hackers to exploit system vulnerabilities, including the human element.
- Common attack vectors include malware, email attachments, web pages, pop-ups, instant messages, text messages and social engineering.

Attack Surface

- The **attack surface** of a software environment is the sum of the different points where an unauthorized user can try to enter data to or extract data from an environment.
- The **physical attack surface** includes everything related to hardware and physical devices; here we're talking about routers, switches, desktop computers, notebooks, tablets and mobile phones, TVs, printers, USB ports, surveillance cameras, etc.
- Once an attacker has accessed a computing device physically, the intruder will look for **digital attack surfaces** left vulnerable by poor coding, default security setting or poorly-maintained software that has not been updated or patched.
- This digital attack surface, includes software applications, networks, ports, operating system services, web and desktop applications and more. In other words, everything running on the digital side of any company.

Attack surface categories



7. Design any five abuse cases of your own to test and validate "eVIT.com/login.*".

5. Abuse Case

- Misuse and abuse cases describe how users can misuse or exploit weak controls in software features to attack an application.
- A direct attack against business functionalities, which may bring in revenue or provide a positive user experience, can have a tangible business impact.
- Abuse cases can be an effective way to drive security requirements to properly protect these critical business use cases.

Abuse Case1

- A user misuses the shopping cart by adding a large quantity of products without the intent to purchase

Abuse Case2

- Denial of service attack with anonymous accounts

Abuse Case3

- Automated denial of service attacks using botnet or testing tools