Lab 10 (14/11/2025)

Union, Bit Fields, Enumeration, Pointers in C

Q 1: Student Result Analyzer

Create a program using:

- enum to represent grade categories: FAIL, PASS, MERIT, DISTINCTION
- union to store either percentage (float) or total marks (int)
- bit-field structure to store pass/fail status (1 bit) and attendance shortage (1 bit)
- A function using pointers that receives the union and bit-field struct to determine the grade.

Sample Input:

Enter total marks out of 500: 420

Enter attendance shortage (0-No,1-Yes): 0

Sample Output:

Percentage = 84.00

Status: Pass

Attendance OK

Grade: DISTINCTION

Q 2: File Permission Simulator

Write a C program that simulates Linux-style file permissions using:

- enum for actions \rightarrow READ=4, WRITE=2, EXECUTE=1
- bit-fields to store file permission bits for user
- union to store either permission integer or symbolic string
- A pointer-based function to update permissions.

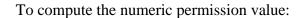
Hint:

Linux file permissions are normally represented using **three permission bits**:

Permission Binary Bit Decimal Value

| Read | 1 | 4 |
|---------|---|---|
| Write | 1 | 2 |
| Execute | 1 | 1 |

Numeric Permission Calculation Formula:



Permission=
$$(R\times4)+(W\times2)+(X\times1)$$

Where each of R, W, X is either 0 or 1.

Example

If a file has:

- Read = 1
- Write = 0
- Execute = 1

Then numeric value =

$$4+0+1=5$$

Symbolic representation = "r-x"

Sample Input:

Enter choice to modify permission:

- 1. Add Read
- 2. Add Write
- 3. Add Execute

Choice: 1

Sample Output:

Updated Permission Bits:

Numeric Permission: 4