3) Create a structure with name student books with members: name, usn, books_id, issue date

o/p: no. of days remaining return for books
no of books still student can granted
any fine

Code ⇒

```
# include <stdio.h>
void main(){
stouct student books{
 char name[50];
 int usn;
 int bookId;
 int issuedate;
}stdb[50];
void main()
{
int n,m;
pf("Enter no.student ~~%d details~~", i++)
pf("Enter total no. of students");
sf("%d", &n);
for (int i=0; i<=n; i++){
pf("\n Enter the details of student s %d", i++)
pf("\n Name");
sf("%s", stdb[i].name);
pf("%usn:");
```

o/p ⇒ Enter no. of students = 1
Enter details of student 1
Name : Abhi
USN : 1234
Enter no. of books you want : 2
Enter book 1 Id : 101
Enter book 2 Id : 102
The no. of books you can get : 2
The total no. of days since you took books 10.
Enter student no. : 1
The days remaining to return : 20

4) Convert inches to feet and add up all.

code ⇒
```
#include <stdio.h>
struct Distance{
    int feet;
    float inch;
};
void main(){
struct Distance arr[] = {{10,3.7},{10.5}};
int n = sizeof(arr) / sizeof(arr[0]);
struct Distance sum = {0,0.0};
for (int i=0; i<n, i++;){
    sum.feet = sum.feet+ arr[i].feet;
    sum.inch = sum.inch + arr[i].inch; }
```

```
scanf("%d", stdb[i].usn);
pf("\n enter the no. of books you want");
sf("%d",&m);
if (m<5){
   for (int j=0; j<=m; j++) {
      pf("\n Enter book %d id", j+1);
      sf("%d", &stdb[j].bookId);
   }
}
else
   pf("You can only get upto 4 books");
pf("the no_of books you can get is %d", 4-m);
pf("the days you since your took books");
sf("%d", stdb[i].issuedate);
}
int o;
pf("\n Enter stud no.:");
sf("%d", &o);
if (o<=90){
   int day_re = 90-(stdb[o].issuedate)
   pf("\n%d day are remaining", days_re);
}
else if (o>90){
   int fee = m*50*(o-90) + m*500;
   pf("In the fine is : %d", fee);
```

```
while (sum.inch >= 12.0){
    sum.inch -= 12.0;
    sum.feet++;
}
pf("Feet sum = %.0f Inch sum = %.2f", sum.feet,
    sum.inch);
}
```

o/p :- Feet sum = 20 Inch sum = 9.2

Week - 4

08/01/2024

Implement linear queue using linear Ar

* pseudo code

```
int a[size]
front = -1;
rear = -1;
enqueue(x){
  if (isfull())
```

```
isfull(){
  if (rear == size-1
    return (-1);
  }
```