studentListADTImpl.h

```
typedef struct student
{
	int regNo;
	char name[30];
	float mark1,mark2,mark3,total;
	char result;
}student;

typedef struct listADT
{
	student s[200];
	// nos is number of elements in list
	int nos;
}listADT;

void initialize(listADT *l)
{
	l->nos = 0;
}

void showRecords(listADT *u)
{
	int i;
	for(i=0;i<u->nos;i++)
		dispRec(u->s[i]);

}
```

```
void dispRec(student s)
{
	printf("\nName: %s  \nRegno: %d \nTotal: %f \nResult: %c",s.name,s.regNo,s.total,s.result);
}

void acceptDetails(student *s)
{
	printf("\nEnter Student details: ");
	printf("\nName: ");
	scanf(" %[^\n]",s->name);
	printf("\n Regno: ");
	scanf("%d",&(s->regNo));
	printf("\n Mark 1: ");
	scanf("%f",&(s->mark1));
	printf("\n Mark 2: ");
	scanf("%f",&(s->mark2));
	printf("\n Mark 3: ");
	scanf("%f",&(s->mark3));

}

void insertFront(listADT *l,student s)
{
	int i;
	l->nos+=1;
	for(i=l->nos+1;i>0;i--)
		l->s[i]=l->s[i-1];
	l->s[i]=s;
}

void insertEnd(listADT *l ,student s)
```

```
{

        l->s[l->nos]=s;
        l->nos++;
}

void insertRegno(listADT *l,student s,int regNo)
{
        int flag=0,i,j;
        for(i=0;i<l->nos;i++)
                if(l->s[i].regNo==regNo)
                {
                        for(j=l->nos;j>i+1;j--)
                                l->s[j]=l->s[j-1];
                        l->s[j]=s;
                        l->nos++;
                        flag++;
                }
        if(flag==0)
                printf("Not found");
}

student* searchRegno(listADT *l,int regNo)
{
        int i,j;
        student *s=NULL;
        for(i=0;i<l->nos;i++)
        {
                if(l->s[i].regNo==regNo)
                        s=&(l->s[i]);
        }
```

```
	return s;
}

listADT* searchName(listADT *l,char name[])
{
	int i,j;
	listADT *temp;
	temp=malloc(sizeof(struct listADT));
	initialize(temp);
	for(i=0;i<l->nos;i++)
		if(strcmp(l->s[i].name,name)==0)
		{
			temp->s[temp->nos]=l->s[i];
			temp->nos+=1;
		}
	if(temp->nos)
		return temp;
	else
		return NULL;
}

void delete(listADT *l,int regNo)
{
	int i,j,flag=0;
	for(i=0;i<l->nos;i++)
		if(l->s[i].regNo==regNo)
		{
			printf("\n The deleted record: ");
			dispRec(l->s[i]);
			for(j=i;j<l->nos-1;j++)
				l->s[j]=l->s[j+1];
```

```
                l->nos-=1;
                flag=1;
            }
        if(flag==0)
            printf("\nRecord not found...");
}

void computeResult(listADT *l)
{
        int i,j;
        for( i=0;i<l->nos;i++)
        {
            l->s[i].total=l->s[i].mark1+l->s[i].mark2+l->s[i].mark3;
            if(l->s[i].mark1<50 || l->s[i].mark2<50 || l->s[i].mark3<50)
                l->s[i].result='F';
            else
            {
                if(l->s[i].mark1>80 && l->s[i].mark2>80 && l->s[i].mark3>80)
                    l->s[i].result='D';
                else
                    l->s[i].result='P';
            }
        }
        for(i=0;i<l->nos;i++)
            printf("\nName: %s \n Regno: %d \n Total: %f \n Result: %c",l->s[i].name,l->s[i].regNo,l->s[i].total,l->s[i].result);
}

listADT* listResult(listADT *l)
{
        listADT *temp;
```

```
	temp=malloc(sizeof(struct listADT));
	initialize(temp);
	for(int i=0;i<l->nos;i++)
	{
		if(!(l->s[i].result=='F'))
		{
			temp->s[temp->nos]=l->s[i];
			temp->nos+=1;
		}
	}
	if(temp->nos)
		return temp;
	else
		return NULL;
}

int listClass(listADT *l)
{
	int count=0,i;
	for(i=0;i<l->nos;i++)
	{
		if(l->s[i].result=='D')
		{
			count+=1;
		}
	}
	return count;
}
```

studentListADTPrototype.h

```
void showRecords(listADT *u);
void dispRec(student s);
void initialize(listADT *l);
void acceptDetails(student *s);
void insertFront(listADT *l,student s);
void insertEnd(listADT *l,student s);
void insertRegno(listADT *l,student s,int );
student* searchRegno(listADT *l,int );
listADT* searchName(listADT *l,char name[]);
void delete(listADT *l,int );
void computeResult(listADT *l);
listADT* listResult(listADT *l);
int listClass(listADT *l);
```

studentListADT.c

```
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
#include"studentlistADTImpl.h"
#include"studentlistADTPrototype.h"

void main()
{
	listADT l,*u;
	initialize(&l);
	char name[80];
	int x=1,ch,t,regNo;
	student *s,g;
	do
	{
		printf("\nSTUDENT lIST OPERATIONS:\n1.insertFront\n2.insertEnd\n3.insertRegno\n4.searchRegno\n5.searchName\n6.deleteRecord\n7.computeResult\n8.listResult\n9.listClass\n10.Exit");
		printf("\nPlease choose one: ");
		scanf("%d",&ch);

		switch(ch)
		{
			case 1:
				acceptDetails(&g);
				insertFront(&l,g);
				break;
			case 2:
				acceptDetails(&g);
```

```
                insertEnd(&l,g);
                break;
        case 3:
                acceptDetails(&g);
                printf("\nEnter the Register no: ");
                scanf("%d",&regNo);
                insertRegno(&l,g,regNo);
                break;
        case 4:
                printf("\nEnter the Register no: ");
                scanf("%d",&regNo);
                s=searchRegno(&l,regNo);
                if(s==NULL)
                        printf("\nRecord not found");
                else
                        dispRec(*s);
                break;
        case 5:
                printf("\nEnter Name: ");
                scanf(" %[^\n]",name);
                u=searchName(&l,name);
                if(u==NULL)
                        printf("\nRecord not found");
                else
                        showRecords(u);
                break;
        case 6:
                printf("\nEnter the Regno: ");
                scanf("%d",&regNo);
                delete(&l,regNo);
                break;
```

```
                        case 7:
                                computeResult(&l);
                                break;
                        case 8:
                                u=listResult(&l);
                                if(u==NULL)
                                        printf("\n\nNo student has passed");
                                else
                                        showRecords(u);

                                break;
                        case 9:
                                t=listClass(&l);
                                printf("\n %d students have got first class ",t);
                                break;
                        default:
                                x=0;
                                break;
                }


        }while(x);
}

/*      OUTPUT:
       ----------

 C:\Users\srina\Desktop>gcc studentListADT.c -o a
In file included from studentListADT.c:4:0:
studentlistADTImpl.h: In function 'showRecords':
studentlistADTImpl.h:25:3: warning: implicit declaration of function 'dispRec' [-Wimplicit-function-
declaration]
```

```
  dispRec(u->s[i]);
  ^
studentlistADTImpl.h: At top level:
studentlistADTImpl.h:29:6: warning: conflicting types for 'dispRec'
 void dispRec(student s)
      ^
studentlistADTImpl.h:25:3: note: previous implicit declaration of 'dispRec' was here
  dispRec(u->s[i]);
  ^

C:\Users\srina\Desktop>a

STUDENT lIST OPERATIONS:
1.insertFront
2.insertEnd
3.insertRegno
4.searchRegno
5.searchName
6.deleteRecord
7.computeResult
8.listResult
9.listClass
10.Exit
Please choose one: 1

Enter Student details:
Name: ram

 Regno: 123

 Mark 1: 99
```

Mark 2: 99

Mark 3: 99

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 2

Enter Student details:

Name: shyam

Regno: 234

Mark 1: 89

Mark 2: 78

Mark 3: 67

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 3

Enter Student details:

Name: mohan

Regno: 345

Mark 1: 55

Mark 2: 45

Mark 3: 34

Enter the Register no: 123

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 7

Name: ram

Regno: 123

Total: 297.000000

Result: D

Name: mohan

Regno: 345

Total: 134.000000

Result: F

Name: shyam

Regno: 234

Total: 234.000000

Result: P

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 4

Enter the Register no: 123

Name: ram

Regno: 123

Total: 297.000000

Result: D

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 5

Enter Name: ram

Name: ram

Regno: 123

Total: 297.000000

Result: D

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 6

Enter the Regno: 123

The deleted record:

Name: ram

Regno: 123

Total: 297.000000

Result: D

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 8

Name: shyam

Regno: 234

Total: 234.000000

Result: P

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 9

0 students have got first class

STUDENT lIST OPERATIONS:

1.insertFront

2.insertEnd

3.insertRegno

4.searchRegno

5.searchName

6.deleteRecord

7.computeResult

8.listResult

9.listClass

10.Exit

Please choose one: 10

C:\Users\srina\Desktop>

*/