**ASSIGNMENT – 2 Array Implementation of list ADT**

**#function\_prototypes.h**

//Structure Protoypes

struct List;

struct Student;

//Typedef of structures

typedef struct List List;

typedef struct Student Student;

//Student Functions

Student getStudent();

void putStudent(const Student);

//List Functions

List\* CreateList();

int isFull(const List \* const);

int isEmpty(const List \* const);

void addEnd(List \* const,const Student);

void addStart(List \* const,const Student);

void addRegno(List \* const,const Student,const int);

void searchName(const List \* const,char \* const);

void deleteRecord(List \* const,const int );

void putList(const List \* const);

void putPrevNext(const List \* const, const int);

**#struct\_student.h**

struct Student{

int regno;

char name[30];

float marks[5];

} ;

struct List{

int capacity;

int size;

Student \* array;

};

Student getStudent(){

Student s;

printf("Enter the name: ");

scanf("%[^\n]",s.name);

printf("Enter the regno: ");

scanf("%d",&s.regno);

printf("Enter the marks in 5 subjects: ");

for(int i = 0 ; i < 5 ; i++)

scanf("%f",&s.marks[i]);

getchar();

return s;

}

void putStudent(const Student S){

printf("Name : %s\n",S.name);

printf("RegNo : %d\n",S.regno);

printf("Marks :");

for(int i = 0 ; i < 5 ; i++)

printf("%.2f ",S.marks[i]);

printf("\n");

}

List\* createList(const int c){

List \* L = (List\*)malloc(sizeof(List));

if(!L){

printf("Memory Allocation Error!");

exit(0);

}

L -> size = 0;

L -> capacity = c;

L -> array = (Student\*)malloc(sizeof(Student) \* c);

if (!L -> array){

printf("Memory Allocation Error!");

exit(0);

}

return L;

}

int isFull(const List \* const L){

return L -> size == L -> capacity;

}

int isEmpty(const List \* const L){

return L -> size == 0;

}

void addEnd(List \* const L,const Student S){

if(isFull(L)){

printf("List Full!");

return;

}

L -> array[L -> size] = S;

L -> size++;

}

void addStart(List \* const L,const Student S){

if(isFull(L)){

printf("List Full!");

return;

}

for(int i = L -> size ; i > 0 ; i--)

L -> array[i] = L -> array[i-1];

L -> array[0] = S;

L -> size++;

}

void addRegno(List \* const L,const Student S,const int rno){

if(isFull(L)){

printf("List Full!");

return;

}

int isFound = 0, index;

//Searching for register number in the list

for(int i = 0 ; i < L -> size; i++)

if(L -> array[i].regno == rno){

index = i;

isFound = 1;

break;

}

if(!isFound){

printf("Given register number does not exist in the list!\n");

return;

}

//Moving all elements to create vacancy to add element

for(int i = L -> size ; i > index ; i--)

L -> array[i] = L -> array[i-1];

L -> array[index+1] = S;

L -> size++;

}

void searchName(const List \* const L,char \* const str){

Student s;

int isFound = 0;

for(int i = 0 ; i < L -> size ; i++)

if(strcmp(L -> array[i].name,str) == 0){

isFound = 1;

s = L -> array[i];

}

if(!isFound){

printf("No such name found!\n");

return;

}

putStudent(s);

}

void deleteRecord(List \* const L,const int regno){

int isFound = 0,index;

for(int i = 0 ; i < L -> size ; i++)

if(L -> array[i].regno == regno){

isFound = 1;

index = i;

break;

}

if(!isFound){

printf("No such record found!\n");

return;

}

for(int i = index ; i < (L -> size - 1) ; i++)

L -> array[i] = L ->array[i+1];

L -> size--;

}

void putList(const List \* const L){

if(isEmpty(L)){

printf("List Empty!\n");

return;

}

for(int i = 0 ; i < L -> size ; i++){

putStudent(L -> array[i]);

printf("\n");

}

}

void putPrevNext(const List \* const L , const int regno){

int isFound = 0,index;

for(int i = 0 ; i < L -> size ; i++)

if(L -> array[i].regno == regno){

isFound = 1 ;

index = i;

break;

}

if(!isFound){

printf("No such record found!\n");

return;

}

if(index != 0){

printf("Previous Student Data\n");

putStudent(L -> array[index-1]);

}

if(index != L -> size -1 ){

printf("Next Student Data\n");

putStudent(L -> array[index + 1]);

}

}

**#StudentList.c //Main program**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include "FunctionPrototypes.h"

#include "struct\_student.h"

int main(void){

int size,

stop = 0,

opt = -1,

regno;

char name[30];

printf("Enter the size of the list: ");

scanf("%d",&size);

List \* L = createList(size);

while(!stop){

system("clear");

printf("1 - Add at end\n2 - Add at start\n3 - Add after regno\n4 - Search by name\n5 - Delete a given record\n");

printf("6 - Display all elements\n7 - Previous and next\n8 - exit\nEnter your choice: ");

scanf("%d",&opt);

getchar();

switch(opt){

case 1: addEnd(L,getStudent()); break;

case 2: addStart(L,getStudent()); break;

case 3: printf("Enter the register number to search: ");

scanf("%d",&regno);

getchar();

addRegno(L,getStudent(),regno); break;

case 4: printf("Enter the name to search: ");

scanf("%[^\n]",name);

getchar();

searchName(L,name); break;

case 5: printf("Enter the register number to search: ");

scanf("%d",&regno);

getchar();

deleteRecord(L,regno); break;

case 6: putList(L); break;

case 7: printf("Enter the register number to search: ");

scanf("%d",&regno);

getchar();

putPrevNext(L,regno); break;

case 8: stop = 1; break;

default:printf("Invalid Input!\n"); break;

}

printf("Press any key to continue......");

getchar();

}

free(L -> array);

free(L);

return 0;

}

**OUTPUT:**