```c
//  Created by Ritwik Chandra Pandey on 24/01/21.
//  183215
//  UCS-3 Assignment
//  Array Implementation of List


#include <stdio.h>
int MAX = 10;
int CURRENT = 0;

int isEmpty(){
    if(CURRENT == 0){
        return 1;
    }
    else return 0;
}

int isFull(){
    if(CURRENT == MAX){
        return 1;
    }
    else return 0;
}

void insertToEOL(int *Listptr){
    if(isFull()){
        printf("List is full, cannot insert\n");
    }else{
        int element;
        printf("Enter element to insert\n");
        scanf("%d",&element);
```

```
        int index = CURRENT;
        Listptr[index] = element;
        CURRENT = CURRENT + 1;
    }

}

void deleteFromEOL (int *List){
    if(isEmpty()){
        printf("List is empty, cannot delete\n");
    }else{
        int element;
        int index = CURRENT - 1;
        element = List[index];
        CURRENT = CURRENT - 1;
        printf("The deleted element is %d from postition %d",element,index+1);

    }
}

void insertToFOL(int *List){
    if(isFull()){
        printf("List is full\n");
    }else{
        int element,index;
        index = CURRENT-1;
        while(index>=0){
            List[index+1]=List[index];
            index--;
        }
        printf("Enter the element to insert\n");
        scanf("%d",&element);
        List[0] = element;
        CURRENT+=1;
```

```
    }
}

void deleteFromFOL(int *List){
    if(isEmpty()){
        printf("List is empty\n");
    }else{
        int element, index;
        element = List[0];
        index = 0;
        while(index<CURRENT){
            List[index]=List[index+1];
            index++;
        }
        printf("The element deleted from position 1 is %d",element);
        CURRENT-=1;
    }

}

void insertToNthPos(int *List){
    if(isFull()){
        printf("The list is already full, insertion is not possible\n");

    }else{
        int element, index,pos;

        index = 0;
        if(isEmpty()){
            printf("List the empty, thus the only available position is 1");
            printf("Enter element to insert : \n");
            scanf("%d",&element);
            printf("Element has been put at position 1");
            List[index]=element;
            CURRENT+=1;
```

```
        }else{
            do{
                printf("Please specify the position between 1 and %d (both included)
\n",CURRENT);
                scanf("%d",&pos);}
            while(pos<=0 || pos>CURRENT);
            index=CURRENT-1;
            printf("Enter element to insert : \n");
            scanf("%d",&element);


            while(index>=pos-1){
                List[index+1]=List[index];
                index--;
            }
            List[pos-1] = element;
            CURRENT+=1;
        }


    }

}
void deleteFromNthPos(int *List){
    int element, pos, index;
    if(isEmpty()){
        printf("Deletion is not possible as the list is already empty\n");

    }else{
        do{
            printf("Please specify the position between 1 and %d(both included)\n",CURRENT);
            scanf("%d",&pos);
        }while(pos<=0 || pos>CURRENT);
        element=List[pos-1];
        index=pos-1;
```

```
        while(index<CURRENT){
            List[index]=List[index+1];
            index++;
        }
        printf("The element just deleted from position %d is %d",pos,element);
        CURRENT -= 1;

    }
}

void searchInList(int *List){
    int element,i,found=0;
    if(isEmpty()){
        printf("List is empty\n");
    }else{
        printf("Enter the element to search\n");
        scanf("%d",&element);
        for(i=0;i<CURRENT;i++){
            if(List[i]==element){
                found = 1;
                break;
            }
        }
        if(found)
            printf("Element %d available in position %d\n",element,i+1);
        else
            printf("Element %d not available in List\n",element);
    }
}

void printList(int List[]){
    int i=0;
    if(isEmpty()){
        printf("List is empty\n");
    }else{
```

```
        for(i=0;i<CURRENT;i++){
            printf("%d -> ",List[i]);
        }
        printf("END\n");
    }
    printf("\tMAX = %d\n",MAX);
    printf("\tCURRENT = %d\n",CURRENT);
}

int main(){
    int List[MAX];
    int select = 0,r;
    printf("\t\tARRAY IMPLEMENTATION OF LIST\n\n");
    do{
        printf("\t1.IsEmpty\n\t2.IsFull\n\t3.Insert element to End Of List\n\t4.Delete from End
Of List\n\t5.Insert element in Front Of List\n\t6.Delete element from Front of List\n\t7.Insert
element to nth Position of List\n\t8.Delete element from nth position of List\n\t9.Search
through the List\n\t10.Print List\n\t11.Exit\n");


    printf("\tPlease Enter Your Choice\n");
    scanf("%d",&select);
    switch(select)
    {
        case 1:
            r=isEmpty();
            r==0?printf("List is Not Empty\n"):printf("List is Empty\n");
            printf("------------------------------------------------------------\n");
            break;
        case 2:
            r=isFull();
            r==0?printf("List is Not Full\n"):printf("List is Full\n");
            printf("------------------------------------------------------------\n");
            break;
        case 3:
```

```
            insertToEOL(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 4:
            deleteFromEOL(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 5:
            insertToFOL(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 6:
            deleteFromFOL(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 7:
            insertToNthPos(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 8:
            deleteFromNthPos(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 9:
            searchInList(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 10:
            printList(List);
            printf("-----------------------------------------------------------\n");
            break;
        case 11:
            break;

        default:
```

```
            printf(“\t\n\nYou have not entered the right choice\n\n");

    }
}while(select!=11);

}
```