**Question 1:**

**Write the program for deleting an element from the beginning and from any position.**

**Code (In C Programming Language)[USING ARRAY IMPLEMENTATION]:**

#include<stdio.h>

int findElement**(**int array**[],** int size**,** int keyToBeDeleted**){**

int i**;**

**for** **(**i **=** 0**;** i **<** size**;** i**++)**

**if** **(**array**[**i**]** **==** keyToBeDeleted**)**

**return** i**;**

**return** **-** 1**;**

**}**

int deleteElement**(**int array**[],** int size**,** int keyToBeDeleted**){**

int pos **=** findElement**(**array**,** size**,** keyToBeDeleted**);**

// If element is not found then it prints Element not found

**if** **(**pos **==** **-** 1**)** **{**

printf**(**"Element not found\n"**);**

**return** size**;**

**}**

// Otherwise it deletes the element & moves rest of the element by one position

int i**;**

**for** **(**i **=** pos**;** i **<** size **-** 1**;** i**++)**

array**[**i**]** **=** array**[**i **+** 1**];**

**return** size **-** 1**;**

**}**

int main**()** **{**

int array**[**100000**];**

int input**,** j**;**

printf**(**"Enter number of digit to be input: "**);**

scanf**(**"%d"**,&**input**);**

**for(**j **=** 0 **;** j **<** input **;** j**++){**

printf**(**"Enter the %d number: "**,**j**+**1**);**

scanf**(**"%d"**,&**array**[**j**]);**

**}**

printf**(**"\n\n"**);**

int i, keyToBeDeleted;

printf("Enter Number to be deleted: ");

scanf("%d",&keyToBeDeleted);

printf("Before Deletion: ");

for (i = 0; i < input; i++)

printf(" %d", array[i]);

printf("\n\n");

int size = deleteElement(array, input, keyToBeDeleted);

printf("After Deletion: ");

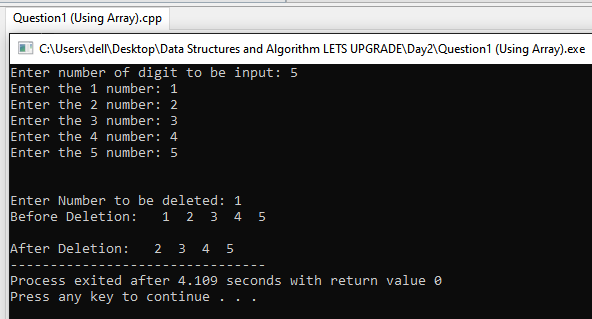
for (i = 0; i < size; i++)

printf(" %d",array[i]);

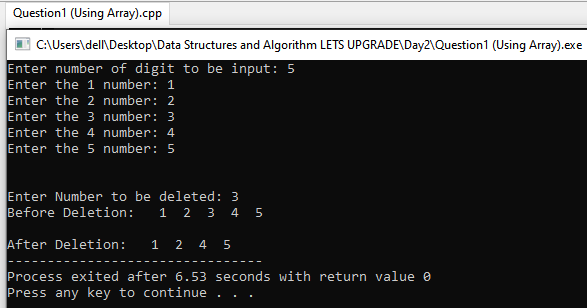
return 0;

}

**Sample Output of The Program (Delete from Beginning):**



**Sample Output of The Program (Delete From any Position):**



**Code (In C Programming Language)[USING SINGLE LINKED LIST IMPLEMENTATION]:**

#include <stdio.h>

#include <stdlib.h>

struct Node **{**

int item**;**

struct Node**\*** next**;**

**};**

void insertAtBeginning**(**struct Node**\*\*** ref**,** int data**)** **{**

struct Node**\*** new\_node **=** **(**struct Node**\*)**malloc**(sizeof(**struct Node**));**

new\_node**->**item **=** data**;**

new\_node**->**next **=** **(\***ref**);**

**(\***ref**)** **=** new\_node**;**

**}**

void insertAfter**(**struct Node**\*** node**,** int data**)** **{**

**if** **(**node **==** **NULL)** **{**

printf**(**"the given previous node cannot be NULL"**);**

**return;**

**}**

struct Node**\*** new\_node **=** **(**struct Node**\*)**malloc**(sizeof(**struct Node**));**

new\_node**->**item **=** data**;**

new\_node**->**next **=** node**->**next**;**

node**->**next **=** new\_node**;**

**}**

void insertAtEnd**(**struct Node**\*\*** ref**,** int data**)** **{**

struct Node**\*** new\_node **=** **(**struct Node**\*)**malloc**(sizeof(**struct Node**));**

struct Node**\*** last **=** **\***ref**;**

new\_node**->**item **=** data**;**

new\_node**->**next **=** **NULL;**

**if** **(\***ref **==** **NULL)** **{**

**\***ref **=** new\_node**;**

**return;**

**}**

**while** **(**last**->**next **!=** **NULL)**

last **=** last**->**next**;**

last**->**next **=** new\_node**;**

**return;**

**}**

void deleteNode**(**struct Node**\*\*** ref**,** int key**)** **{**

struct Node **\***temp **=** **\***ref**,** **\***prev**;**

**if** **(**temp **!=** **NULL** **&&** temp**->**item **==** key**)** **{**

**\***ref **=** temp**->**next**;**

free**(**temp**);**

**return;**

**}**

**while** **(**temp **!=** **NULL** **&&** temp**->**item **!=** key**)** **{**

prev **=** temp**;**

temp **=** temp**->**next**;**

**}**

**if** **(**temp **==** **NULL)** **return;**

prev**->**next **=** temp**->**next**;**

free**(**temp**);**

**}**

void printList**(**struct Node**\*** node**)** **{**

**while** **(**node **!=** **NULL)** **{**

printf**(**"%d -> "**,** node**->**item**);**

node **=** node**->**next**;**

**}**

**}**

int main**()** **{**

struct Node**\*** head **=** **NULL;**

insertAtEnd**(&**head**,** 1**);**

insertAtBeginning**(&**head**,** 2**);**

insertAtBeginning**(&**head**,** 3**);**

insertAtEnd**(&**head**,** 4**);**

insertAfter**(**head**->**next**,** 5**);**

printf**(**"Linked list: "**);**

printList**(**head**);**

int hapus**;**

printf**(**"\n\nEnter number you want to deleted: "**);**

scanf**(**"%d"**,&**hapus**);**

printf**(**"\nAfter deleting an element: "**);**

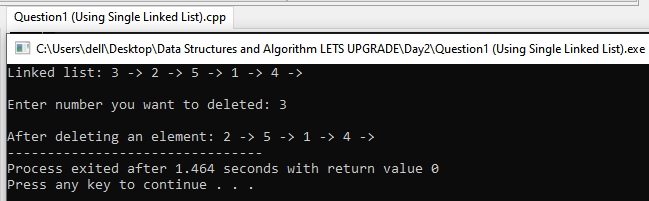
deleteNode**(&**head**,** hapus**);**

printList**(**head**);**

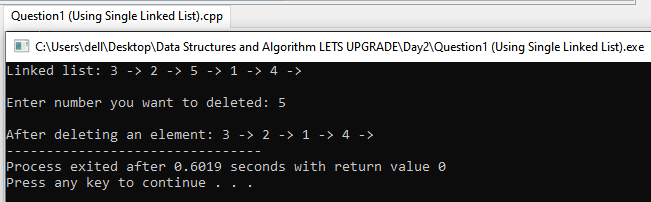
**return** 0**;**

**}**

**Sample Output of The Program (Delete from Beginning):**



**Sample Output of The Program (Delete From any Position):**



**Question 2:**

**Write the program for printing the array after rotating it k times towards left, where k would be taken as user input.**

**Code (In C Programming Language):**

#include <stdio.h>

void leftRotate**(**int arr**[],** int k**,** int n**){**

int mod **=** k **%** n**;**

printf**(**"Output After %d Left Rotation is: "**,**k**);**

**for** **(**int i **=** 0**;** i **<** n**;** i**++){**

printf**(**"%d "**,**arr**[(**mod **+** i**)** **%** n**]);**

**}**

printf**(**"\n"**);**

**}**

int main**()**

**{**

int array**[**100000**];**

int input**,**j**;**

printf**(**"Enter number of digit to be input: "**);**

scanf**(**"%d"**,&**input**);**

**for(**j **=** 0 **;** j **<** input **;** j**++){**

printf**(**"Enter the %d number: "**,**j**+**1**);**

scanf**(**"%d"**,&**array**[**j**]);**

**}**

int k**;**

printf**(**"Input How many time you want to Left Rotate the Array? : "**);**

scanf**(**"%d"**,&**k**);**

printf**(**"Before Left Rotation (Original): "**);**

**for** **(**int i **=** 0**;** i **<** input**;** i**++){**

printf**(**"%d "**,**array**[**i**]);**

**}**

printf**(**"\n"**);**

leftRotate**(**array**,** k**,** input**);**

**return** 0**;**

**}**

**Sample Output of The Program (After 3 Left Rotation):**

