Ex. No. 2 **List using Array**

Date:

**Aim**

To perform various operations on List ADT using array implementation.

**Algorithm**

1. Start

2. Create a list of n elements

3. Display list operations as a menu

4. Accept user choice

5. If choice = 1 then

Get position of element to be deleted

Move elements one position upwards thereon.

Decrement length of the list

Else if choice = 2

Get position of element to be inserted.

Increment length of the list

Move elements one position downwards thereon

Store the new element in corresponding position

Else if choice = 3

Traverse the list and inspect each element

Report position if it exists.

6. Stop

**Program**

|  |
| --- |
|  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

/\* List operation using Arrays \*/

#include <stdio.h>

#include <stdlib.h>

void create();

void insert();

void search();

void deletion();

void display();

int i, e, n, pos;

static int b[50];

void main()

{

int ch;

char g = 'y';

create();

do

{

printf("\n List Operations");

printf("\n 1.Deletion\n 2.Insert\n 3.Search\n4.Exit\n");

printf("Enter your choice: ");

scanf("%d", &ch);

switch(ch)

{

case 1:

printf("\n");

deletion();

break;

case 2:

printf("\n");

insert();

break;

case 3:

printf("\n");

//printf("\nEnter the element to be searched: ");

search();

break;

case 4:

exit(0);

default:

printf("\nEnter the correct choice:");

}

printf("\n");

printf("Do you want to continue: ");

fflush(stdin);

scanf("\n %c",&g);

} while(g=='y' || g=='Y');

}

void create()

{

printf("\n Enter the number of elements:");

scanf("%d",&n);

printf("\n Enter list elements: ");

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

scanf("%d", &b[i]);

}

void deletion()

{

printf("\n enter the position you want to delete: ");

scanf("%d", &pos);

if(pos >= n)

printf("\n Invalid location");

else

{

for(i=pos+1; i<n; i++)

b[i-1] = b[i];

n--;

printf("List elements after deletion");

display();

}

}

void search()

{

int flag = 0;

printf("\nEnter the element to be searched: ");

scanf("%d", &e);

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

{

if(b[i] == e)

{

flag = 1;

printf("Element is in the %d position", i);

break;

}

}

if(flag == 0)

printf("Value %d is not in the list", e);

}

void insert()

{

printf("\n Enter the position you need to insert: ");

scanf("%d", &pos);

if(pos >= n)

printf("\n Invalid location");

else

{

++n;

for(i=n; i>pos; i--)

b[i] = b[i-1];

printf("\n Enter the element to insert: ");

scanf("%d", &e);

b[pos] = e;

}

printf("\n List after insertion:");

display();

}

void display()

{

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

printf("\n %d", b[i]);

}

**Output**

**Result**

Thus various operations was successfully executed on list using array implementation.