**DSA LAB 1**

1. **Write a C program to insert and delete an element from a 1-D static array.**

Code

#include <stdio.h>

int search(int \*a, int size, int elem)

{

int i;

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

{

if (elem == \*(a + i))

{

return i;

}

}

return -1;

}

void delete (int \*a, int size, int elem)

{

int i, index = search(a, size, elem);

a += index;

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

{

\*a = \*(a + 1);

a++;

}

size--;

a -= size;

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

{

printf("%d ", \*(a + i));

}

}

void insert(int \*a, int size, int elem, int pos)

{

int i = 0;

a += size;

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

{

\*a = \*(a - 1);

a--;

}

\*a = elem;

a -= pos - 1;

size++;

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

{

printf("%d ", \*(a + i));

}

}

int main()

{

int a[100], size, i, elem, pos;

char ch;

printf("Enter size of array ");

scanf("%d", &size);

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

{

printf("Enter %d element ", i + 1);

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

}

printf("Enter i to insert/ d to delete ");

scanf(" %c", &ch);

if (ch == 'i')

{

printf("Enter element to insert ");

scanf("%d", &elem);

printf("Enter positon ");

scanf("%d", &pos);

insert(a, size, elem, pos);

}

else if (ch == 'd')

{

printf("Enter element to delete ");

scanf("%d", &elem);

delete (a, size, elem);

}

else

{

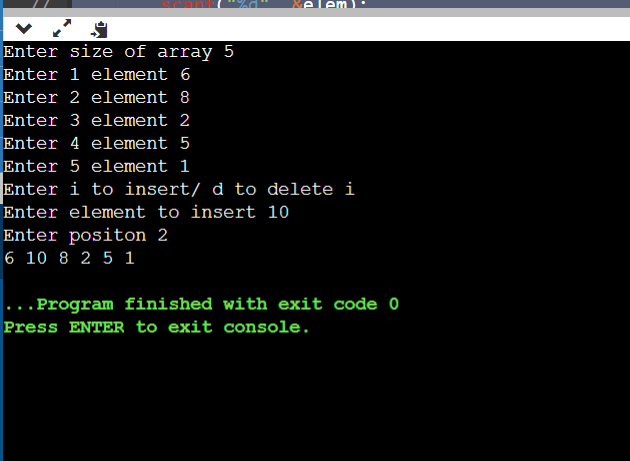
printf("Wrong choice ");

}

return 0;

}

Output



1. **Write a C program to reverse the elements present in a 1-D static array.**

**Code**

#include <stdio.h>

int main() {

int array[100], n, c, t, end;

scanf("%d", &n);

end = n - 1;

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

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

for (c = 0; c < n/2; c++) {

t = array[c];

array[c] = array[end];

array[end] = t;

end--;

}

printf("Reversed array elements are:\n");

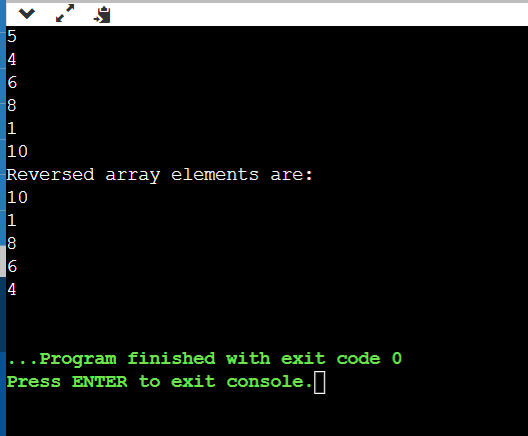
for (c = 0; c < n; c++)

printf("%d\n", array[c]);

return 0;

}

**Output**



1. **Write a C program to delete duplicates elements in a 1-D static array.**

#include<stdio.h>

int main() {

int arr[20], i, j, k, size;

printf("\nEnter array size : ");

scanf("%d", &size);

printf("\nAccept Numbers : ");

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

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

printf("\nArray with Unique list : ");

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

for (j = i + 1; j < size;) {

if (arr[j] == arr[i]) {

for (k = j; k < size; k++) {

arr[k] = arr[k + 1];

}

size--;

} else

j++;

}

}

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

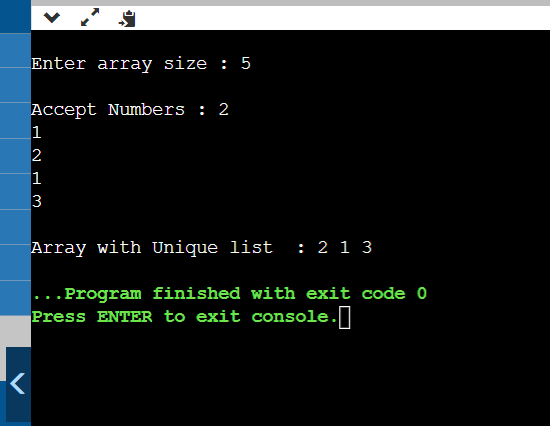
printf("%d ", arr[i]);

}

return (0);

}

**Output**



1. **Write a C program to find the largest and smallest element present in a 1-D static array.**

#include<stdio.h>

int main()

{

int a[50],i,n,large,small;

printf("How many elements:");

scanf("%d",&n);

printf("Enter the Array:");

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

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

large=small=a[0];

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

{

if(a[i]>large)

large=a[i];

if(a[i]<small)

small=a[i];

}

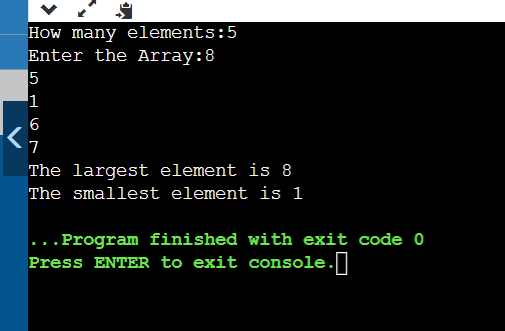
printf("The largest element is %d",large);

printf("\nThe smallest element is %d",small);

return 0;

}

**Output**



1. **Write a C program to find the second largest and second smallest element present in a 1-D static array.**

#include <stdio.h>

void main ()

{

int number[30];

int i, j, a, n, counter, average;

printf("Enter the value of N\n");

scanf("%d", &n);

printf("Enter the numbers \n");

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

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

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

{

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

{

if (number[i] < number[j])

{

a = number[i];

number[i] = number[j];

number[j] = a;

}

}

}

printf("The numbers arranged in descending order are given below \n");

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

{

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

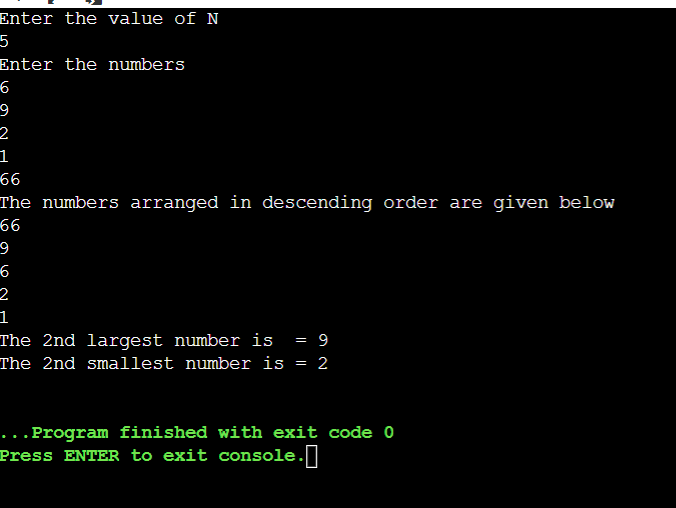
}

printf("The 2nd largest number is = %d\n", number[1]);

printf("The 2nd smallest number is = %d\n", number[n - 2]);

}

**Output**



1. **Write a C program for linear search and binary search.**

Code for linear search

#include <stdio.h>

int main()

{

int array[100], search, c, n;

printf("Enter number of elements in array\n");

scanf("%d", &n);

printf("Enter %d integer(s)\n", n);

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

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

printf("Enter a number to search\n");

scanf("%d", &search);

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

{

if (array[c] == search) /\* If required element is found \*/

{

printf("%d is present at location %d.\n", search, c+1);

break;

}

}

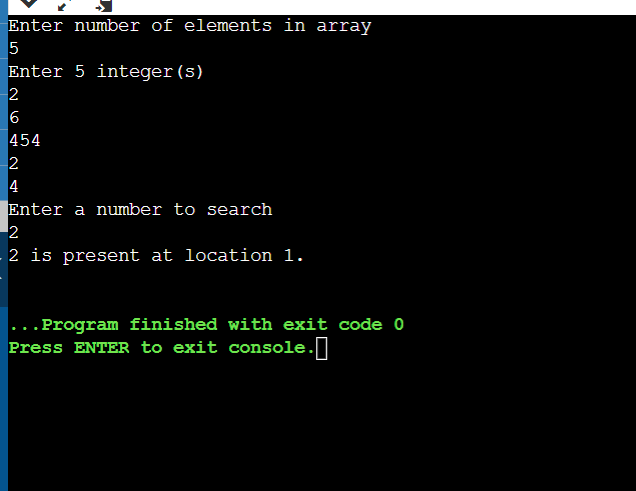
if (c == n)

printf("%d isn't present in the array.\n", search);

return 0;

}

**Output**



**Code for Binary Search**

#include <stdio.h>

int main()

{

int c, first, last, middle, n, search, array[100];

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

scanf("%d", &n);

printf("Enter %d integers\n", n);

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

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

printf("Enter value to find\n");

scanf("%d", &search);

first = 0;

last = n - 1;

middle = (first+last)/2;

while (first <= last) {

if (array[middle] < search)

first = middle + 1;

else if (array[middle] == search) {

printf("%d found at location %d.\n", search, middle+1);

break;

}

else

last = middle - 1;

middle = (first + last)/2;

}

if (first > last)

printf("Not found! %d isn't present in the list.\n", search);

return 0;

}

**Output**

