***1.write a c-program to find the largest and smallest elements of an array using pointers***

***Code:***

#include <stdio.h>

int main() {

int arr[100], n;

int \*ptr;

int largest, smallest;

printf("Enter number of elements in the array: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

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

}

ptr = arr; // pointer points to the first element of the array

largest = \*ptr;

smallest = \*ptr;

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

ptr = arr + i; // move pointer to next element

if (\*ptr > largest)

largest = \*ptr;

if (\*ptr < smallest)

smallest = \*ptr;

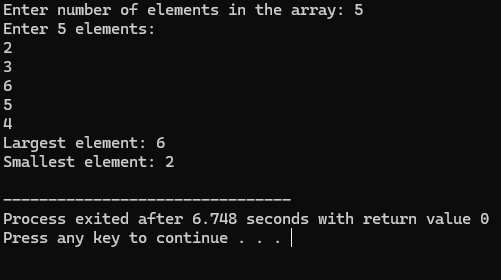
}

printf("Largest element: %d\n", largest);

printf("Smallest element: %d\n", smallest);

}

Output:



***2.write a c-program to find the sum of elements in an arrays using pointers***

Code:

#include <stdio.h>

int main() {

int arr[100], n, sum = 0;

int \*ptr;

printf("Enter number of elements in the array: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

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

}

ptr = arr;

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

sum += \*(ptr + i);

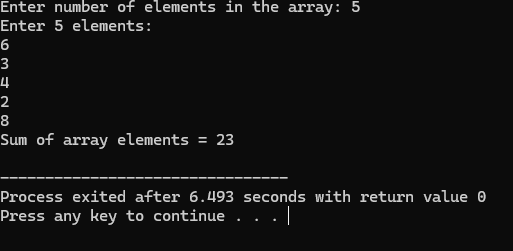
}

printf("Sum of array elements = %d\n", sum);

return 0;

}

Output:



***3.write a c program to search the target element in an array using poiners***

Code:

#include <stdio.h>

int main() {

int arr[100], n, target, found = 0;

int \*ptr;

printf("Enter number of elements in the array: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

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

}

printf("Enter the target element to search: ");

scanf("%d", &target);

ptr = arr;

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

if (\*(ptr + i) == target) {

printf("Target element %d found at position %d (index %d)\n", target, i + 1, i);

found = 1;

break;

}

}

if (!found) {

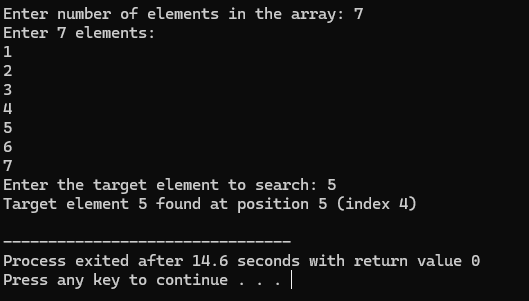
printf("Target element %d not found in the array.\n", target);

}

return 0;

}

Output:



***4.write a c-program to reverse of an array of the elements using pointers***

***Code:***

#include <stdio.h>

int main() {

int arr[100], n;

int \*start, \*end, temp;

printf("Enter number of elements in the array: ");

scanf("%d", &n);

printf("Enter %d elements:\n", n);

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

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

}

start = arr;

end = arr + n - 1;

while (start < end) {

temp = \*start;

\*start = \*end;

\*end = temp;

start++;

end--;

}

printf("Reversed array:\n");

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

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

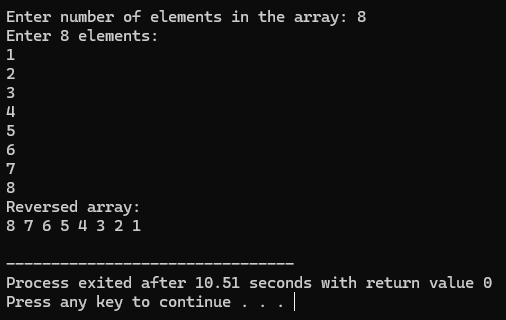
}

printf("\n");

return 0;

}

Output:



***5.write a c program to print transpose of a matrix using pointers***

***Code:***

#include <stdio.h>

int main() {

int matrix[10][10], transpose[10][10];

int rows, cols;

int \*ptrMatrix, \*ptrTranspose;

printf("Enter number of rows: ");

scanf("%d", &rows);

printf("Enter number of columns: ");

scanf("%d", &cols);

printf("Enter elements of the matrix (%d x %d):\n", rows, cols);

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

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

}

}

for (int i = 0; i < rows; i++) {

for (int j = 0; j < cols; j++) {

ptrMatrix = &matrix[i][j];

ptrTranspose = &transpose[j][i];

\*ptrTranspose = \*ptrMatrix;

}

}

printf("Transpose of the matrix (%d x %d):\n", cols, rows);

for (int i = 0; i < cols; i++) {

for (int j = 0; j < rows; j++) {

printf("%d ", transpose[i][j]);

}

printf("\n");

}

return 0;

}

Output:

