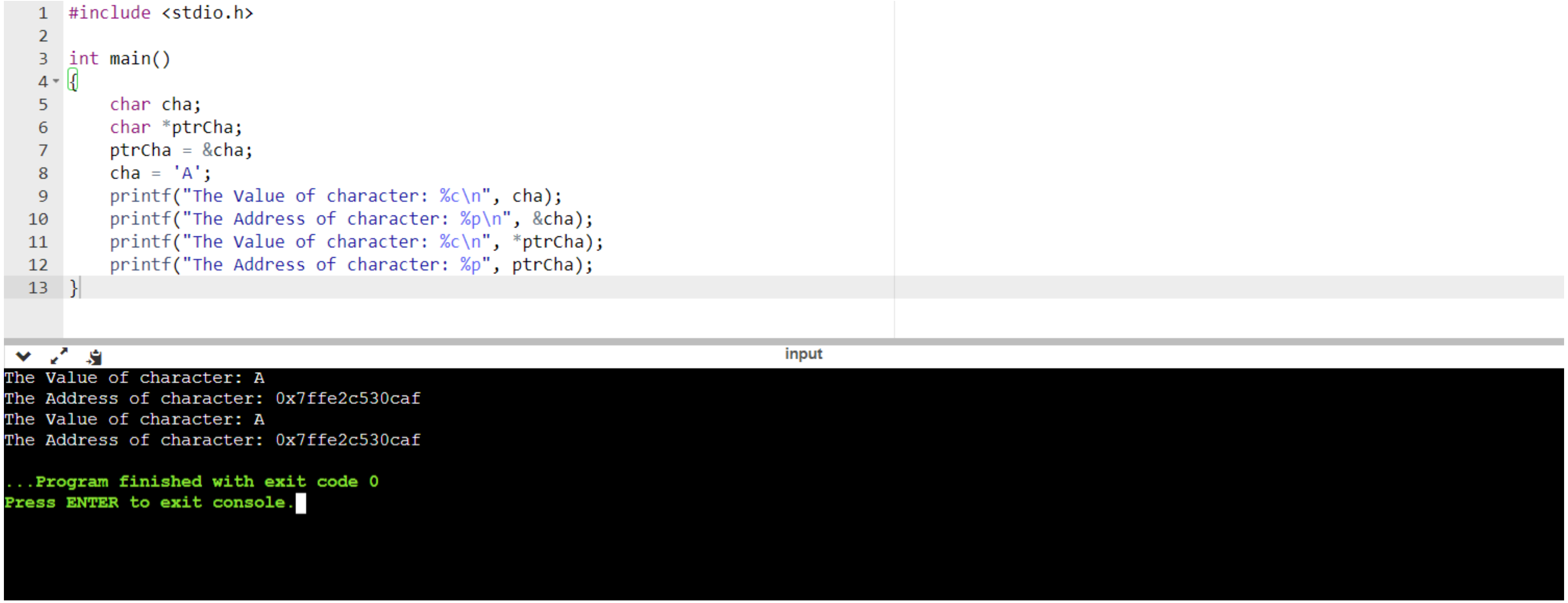
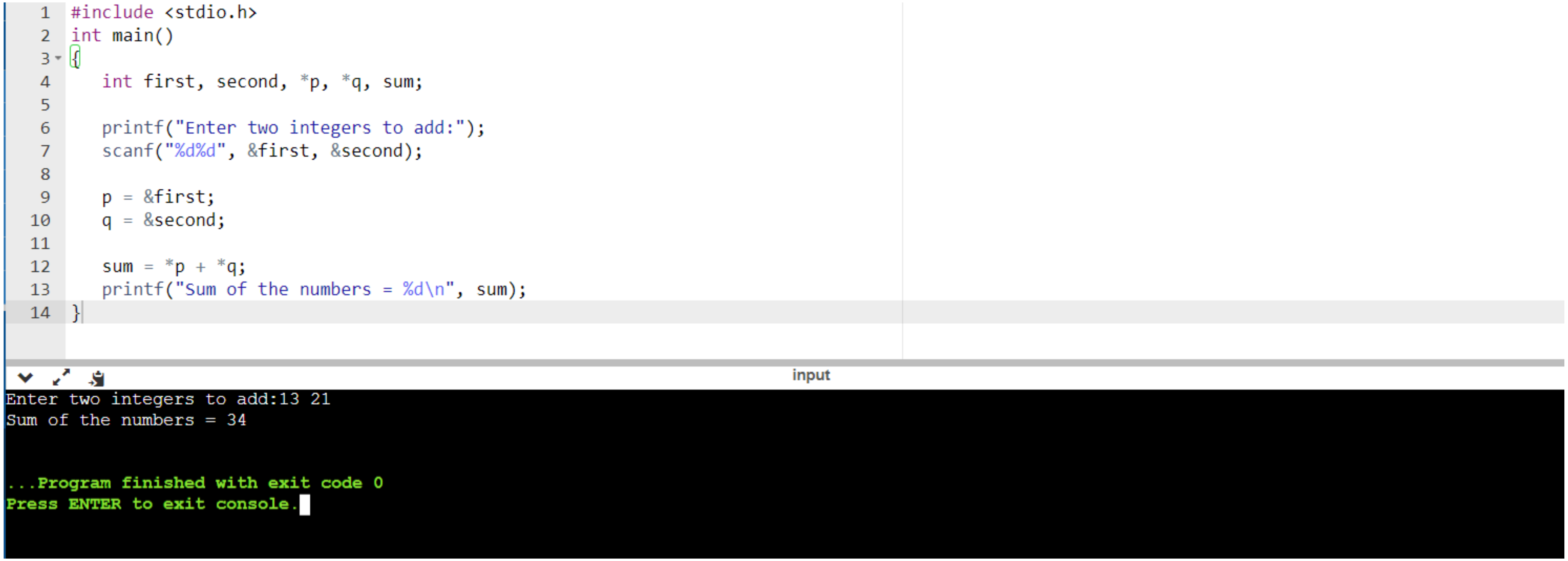
C program to create, initialize and use pointers.



```c
#include <stdio.h>

int main()
{
    char cha;
    char *ptrCha;
    ptrCha = &cha;
    cha = 'A';
    printf("The Value of character: %c\n", cha);
    printf("The Address of character: %p\n", &cha);
    printf("The Value of character: %c\n", *ptrCha);
    printf("The Address of character: %p", ptrCha);
}
```

```
The Value of character: A
The Address of character: 0x7ffe2c530caf
The Value of character: A
The Address of character: 0x7ffe2c530caf

...Program finished with exit code 0
Press ENTER to exit console.
```

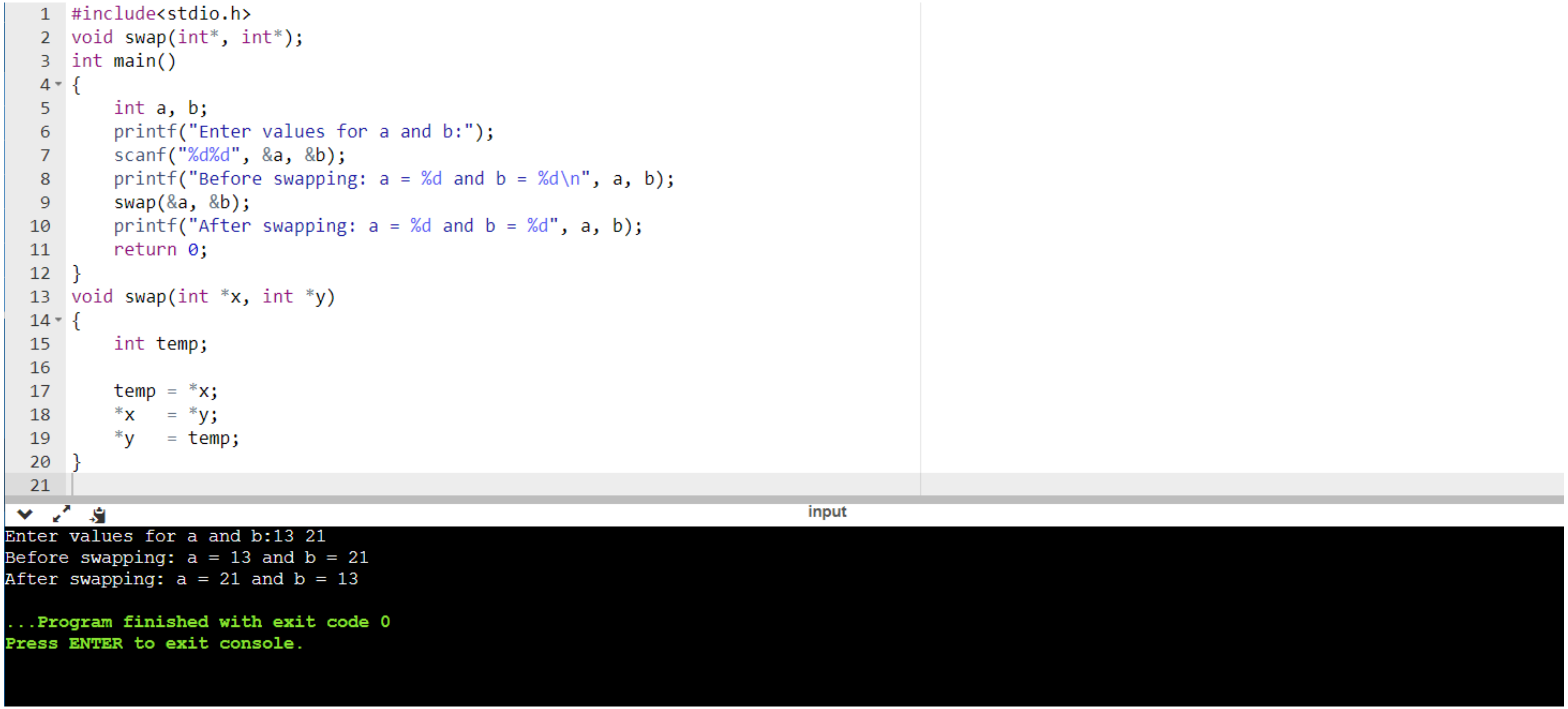
C program to add two numbers using pointers.



```c
#include <stdio.h>
int main()
{
    int first, second, *p, *q, sum;

    printf("Enter two integers to add:");
    scanf("%d%d", &first, &second);

    p = &first;
    q = &second;

    sum = *p + *q;
    printf("Sum of the numbers = %d\n", sum);
}
```

```
Enter two integers to add:13 21
Sum of the numbers = 34


...Program finished with exit code 0
Press ENTER to exit console.
```

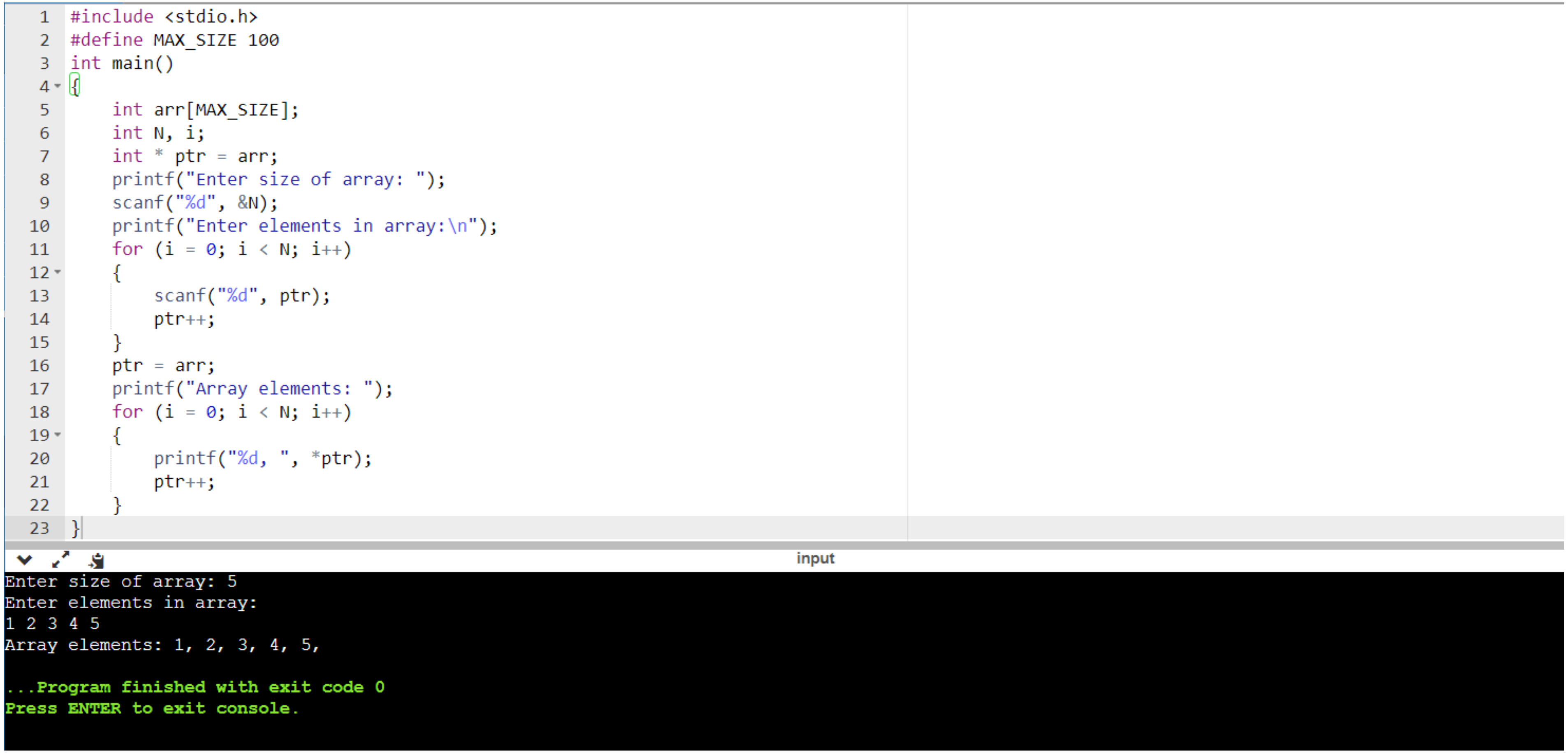
C program to swap two numbers using pointers.



```c
#include<stdio.h>
void swap(int*, int*);
int main()
{
    int a, b;
    printf("Enter values for a and b:");
    scanf("%d%d", &a, &b);
    printf("Before swapping: a = %d and b = %d\n", a, b);
    swap(&a, &b);
    printf("After swapping: a = %d and b = %d", a, b);
    return 0;
}
void swap(int *x, int *y)
{
    int temp;

    temp = *x;
    *x   = *y;
    *y   = temp;
}

```

```
Enter values for a and b:13 21
Before swapping: a = 13 and b = 21
After swapping: a = 21 and b = 13

...Program finished with exit code 0
Press ENTER to exit console.
```

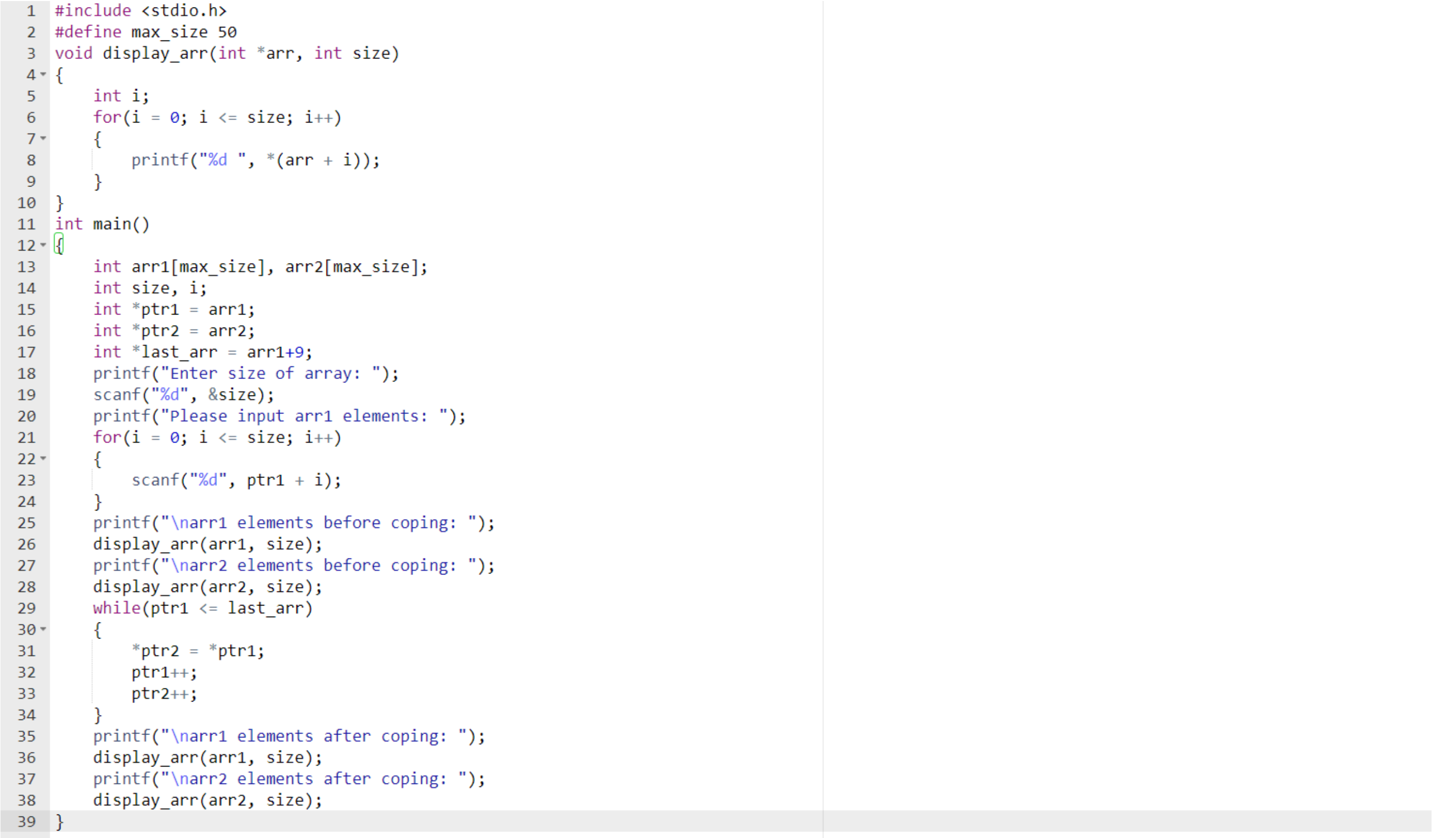
C program to input and print array elements using pointer.



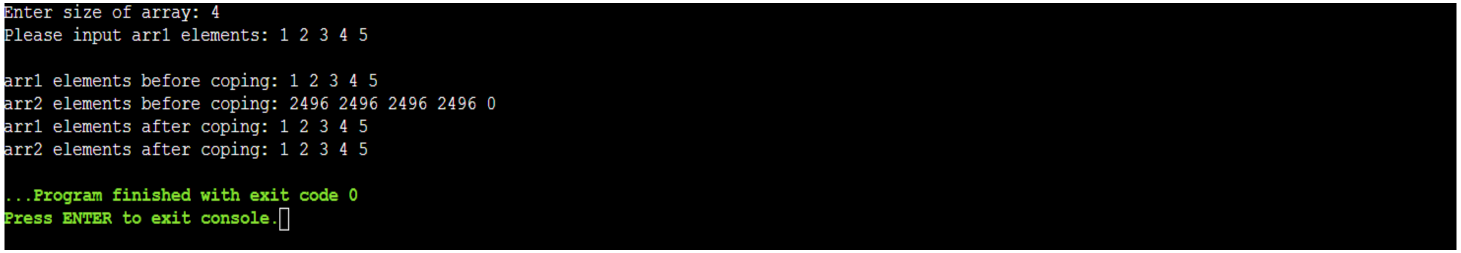
```c
#include <stdio.h>
#define MAX_SIZE 100
int main()
{
    int arr[MAX_SIZE];
    int N, i;
    int * ptr = arr;
    printf("Enter size of array: ");
    scanf("%d", &N);
    printf("Enter elements in array:\n");
    for (i = 0; i < N; i++)
    {
        scanf("%d", ptr);
        ptr++;
    }
    ptr = arr;
    printf("Array elements: ");
    for (i = 0; i < N; i++)
    {
        printf("%d, ", *ptr);
        ptr++;
    }
}
```

```
Enter size of array: 5
Enter elements in array:
1 2 3 4 5
Array elements: 1, 2, 3, 4, 5,

...Program finished with exit code 0
Press ENTER to exit console.
```

C program to copy one array to another using pointers.

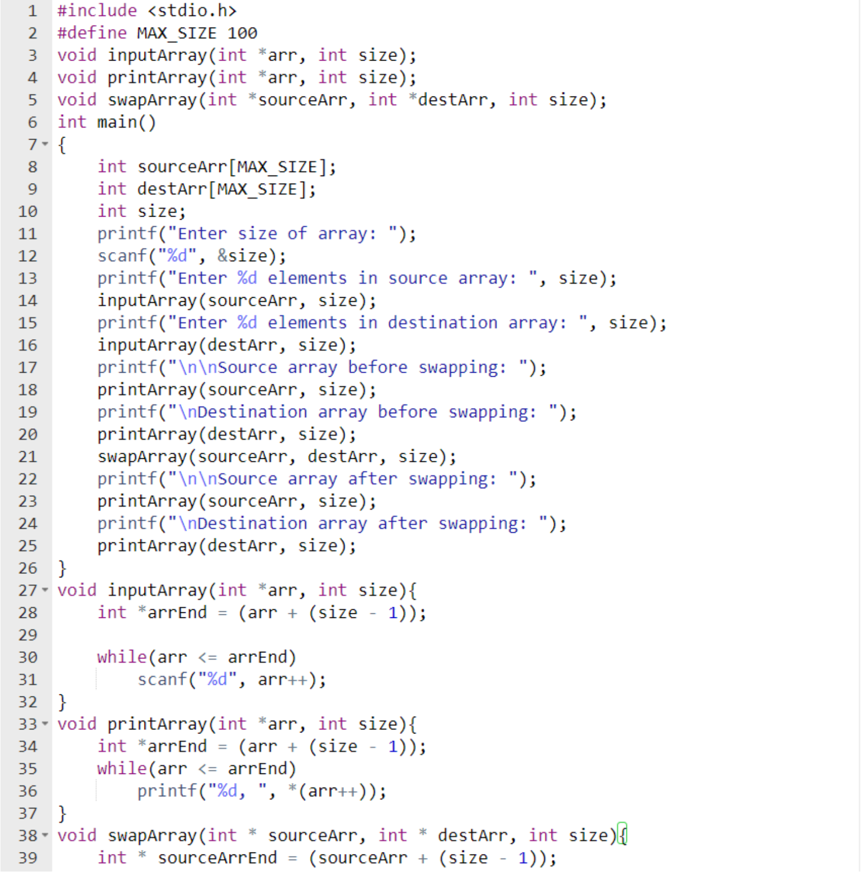


```c
#include <stdio.h>
#define max_size 50
void display_arr(int *arr, int size)
{
    int i;
    for(i = 0; i <= size; i++)
    {
        printf("%d ", *(arr + i));
    }
}
int main()
{
    int arr1[max_size], arr2[max_size];
    int size, i;
    int *ptr1 = arr1;
    int *ptr2 = arr2;
    int *last_arr = arr1+9;
    printf("Enter size of array: ");
    scanf("%d", &size);
    printf("Please input arr1 elements: ");
    for(i = 0; i <= size; i++)
    {
        scanf("%d", ptr1 + i);
    }
    printf("\narr1 elements before coping: ");
    display_arr(arr1, size);
    printf("\narr2 elements before coping: ");
    display_arr(arr2, size);
    while(ptr1 <= last_arr)
    {
        *ptr2 = *ptr1;
        ptr1++;
        ptr2++;
    }
    printf("\narr1 elements after coping: ");
    display_arr(arr1, size);
    printf("\narr2 elements after coping: ");
    display_arr(arr2, size);
}
```

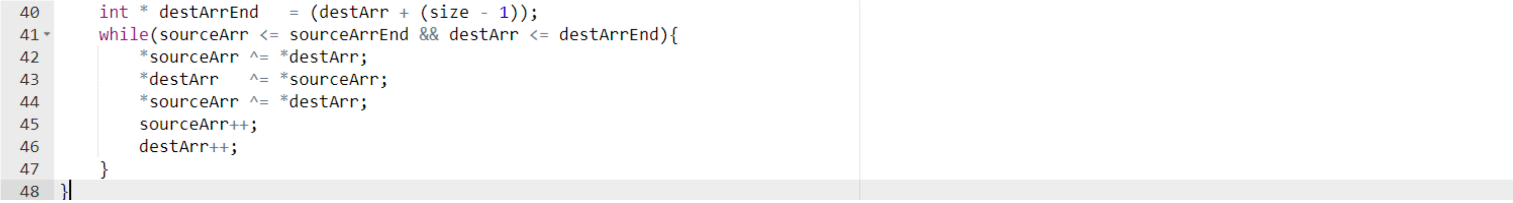


```
Enter size of array: 4
Please input arr1 elements: 1 2 3 4 5

arr1 elements before coping: 1 2 3 4 5
arr2 elements before coping: 2496 2496 2496 2496 0
arr1 elements after coping: 1 2 3 4 5
arr2 elements after coping: 1 2 3 4 5

...Program finished with exit code 0
Press ENTER to exit console.
```

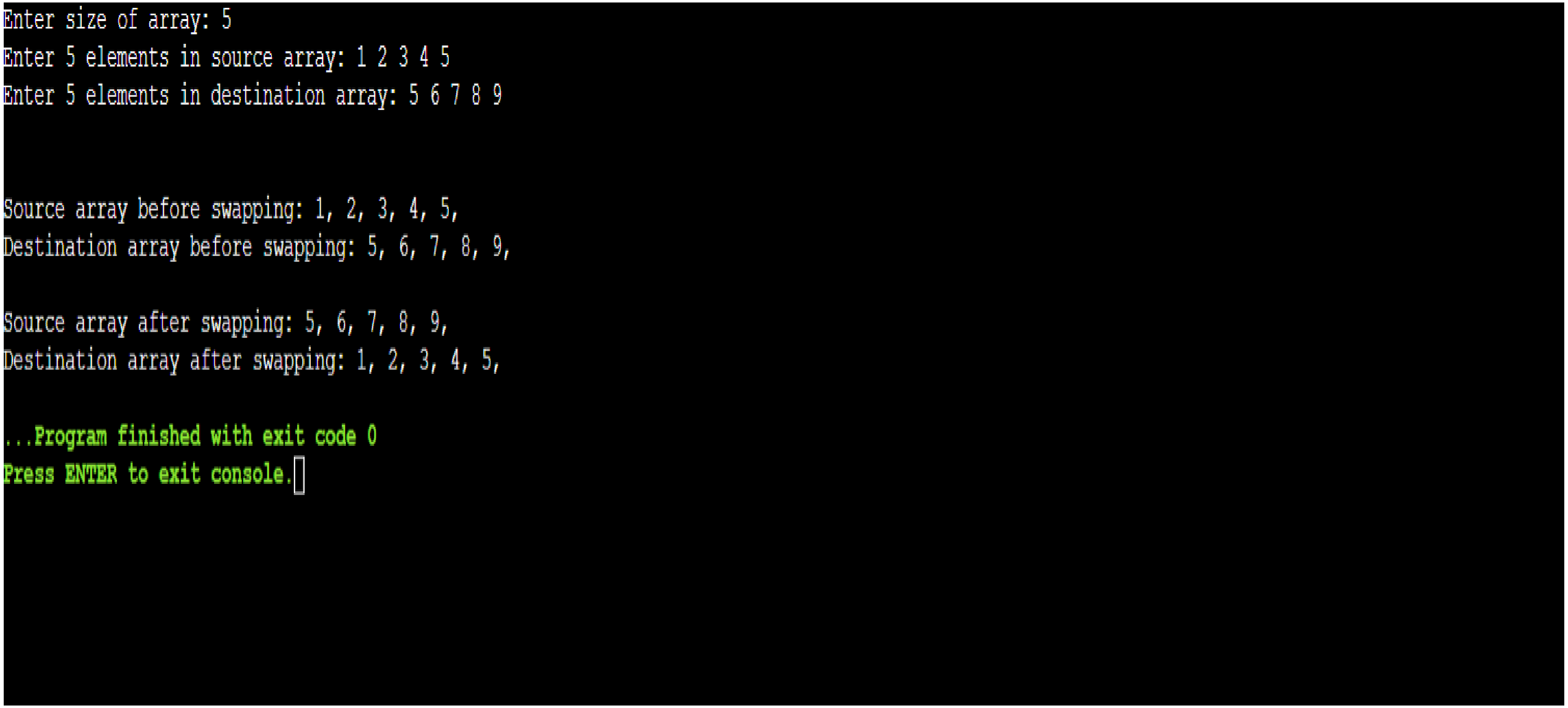
C program to swap two arrays using pointers.



```c
#include <stdio.h>
#define MAX_SIZE 100
void inputArray(int *arr, int size);
void printArray(int *arr, int size);
void swapArray(int *sourceArr, int *destArr, int size);
int main()
{
    int sourceArr[MAX_SIZE];
    int destArr[MAX_SIZE];
    int size;
    printf("Enter size of array: ");
    scanf("%d", &size);
    printf("Enter %d elements in source array: ", size);
    inputArray(sourceArr, size);
    printf("Enter %d elements in destination array: ", size);
    inputArray(destArr, size);
    printf("\n\nSource array before swapping: ");
    printArray(sourceArr, size);
    printf("\nDestination array before swapping: ");
    printArray(destArr, size);
    swapArray(sourceArr, destArr, size);
    printf("\n\nSource array after swapping: ");
    printArray(sourceArr, size);
    printf("\nDestination array after swapping: ");
    printArray(destArr, size);
}
void inputArray(int *arr, int size){
    int *arrEnd = (arr + (size - 1));

    while(arr <= arrEnd)
        scanf("%d", arr++);
}
void printArray(int *arr, int size){
    int *arrEnd = (arr + (size - 1));
    while(arr <= arrEnd)
        printf("%d, ", *(arr++));
}
void swapArray(int * sourceArr, int * destArr, int size){
    int * sourceArrEnd = (sourceArr + (size - 1));
```

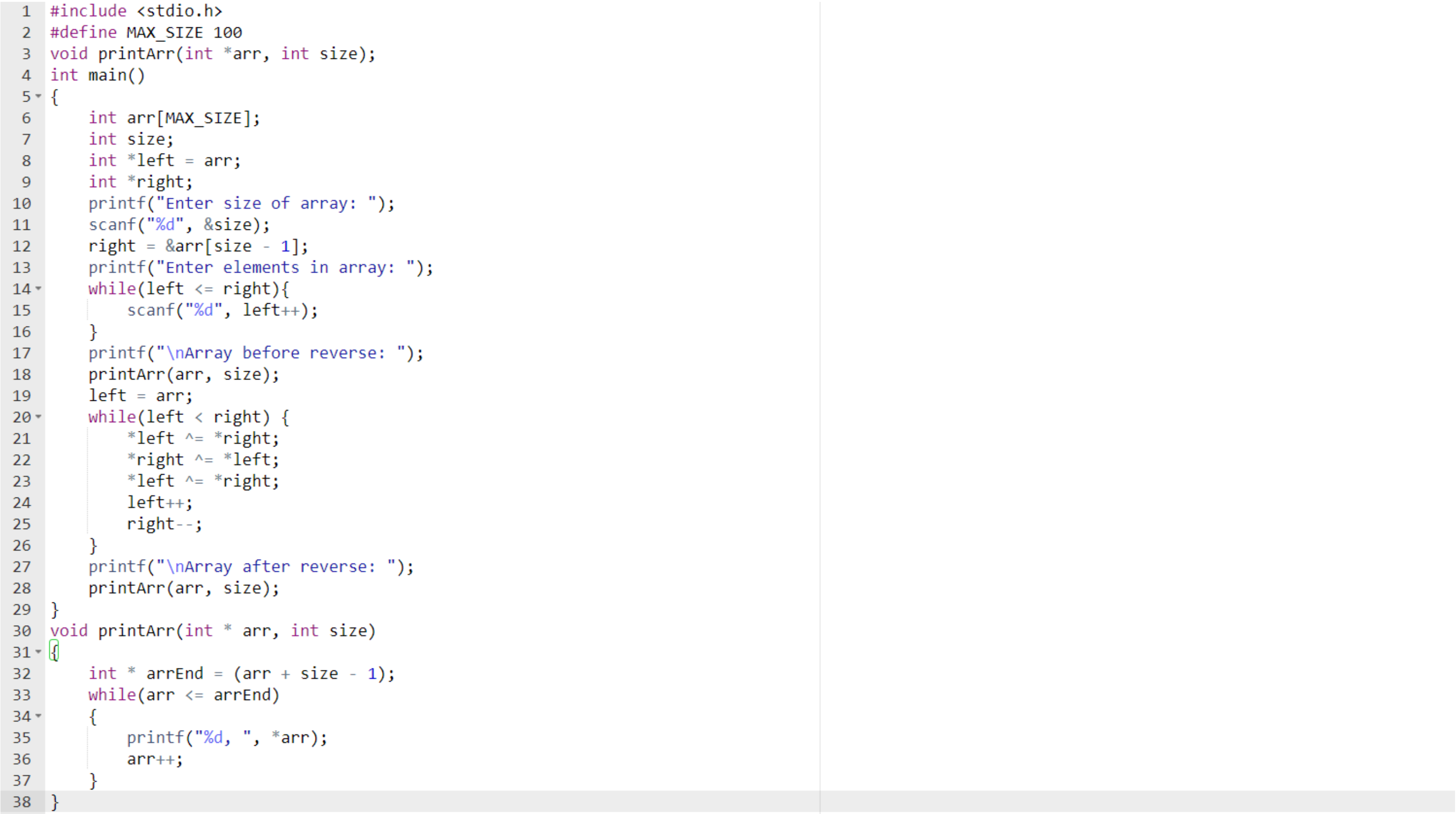


```c
    int * destArrEnd   = (destArr + (size - 1));
    while(sourceArr <= sourceArrEnd && destArr <= destArrEnd){
        *sourceArr ^= *destArr;
        *destArr   ^= *sourceArr;
        *sourceArr ^= *destArr;
        sourceArr++;
        destArr++;
    }
}
```

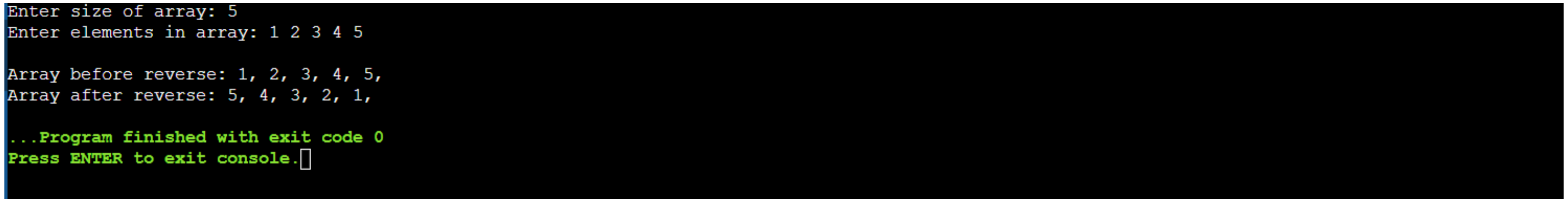


```
Enter size of array: 5
Enter 5 elements in source array: 1 2 3 4 5
Enter 5 elements in destination array: 5 6 7 8 9


Source array before swapping: 1, 2, 3, 4, 5,
Destination array before swapping: 5, 6, 7, 8, 9,

Source array after swapping: 5, 6, 7, 8, 9,
Destination array after swapping: 1, 2, 3, 4, 5,

...Program finished with exit code 0
Press ENTER to exit console.
```

C program to reverse an array using pointers.

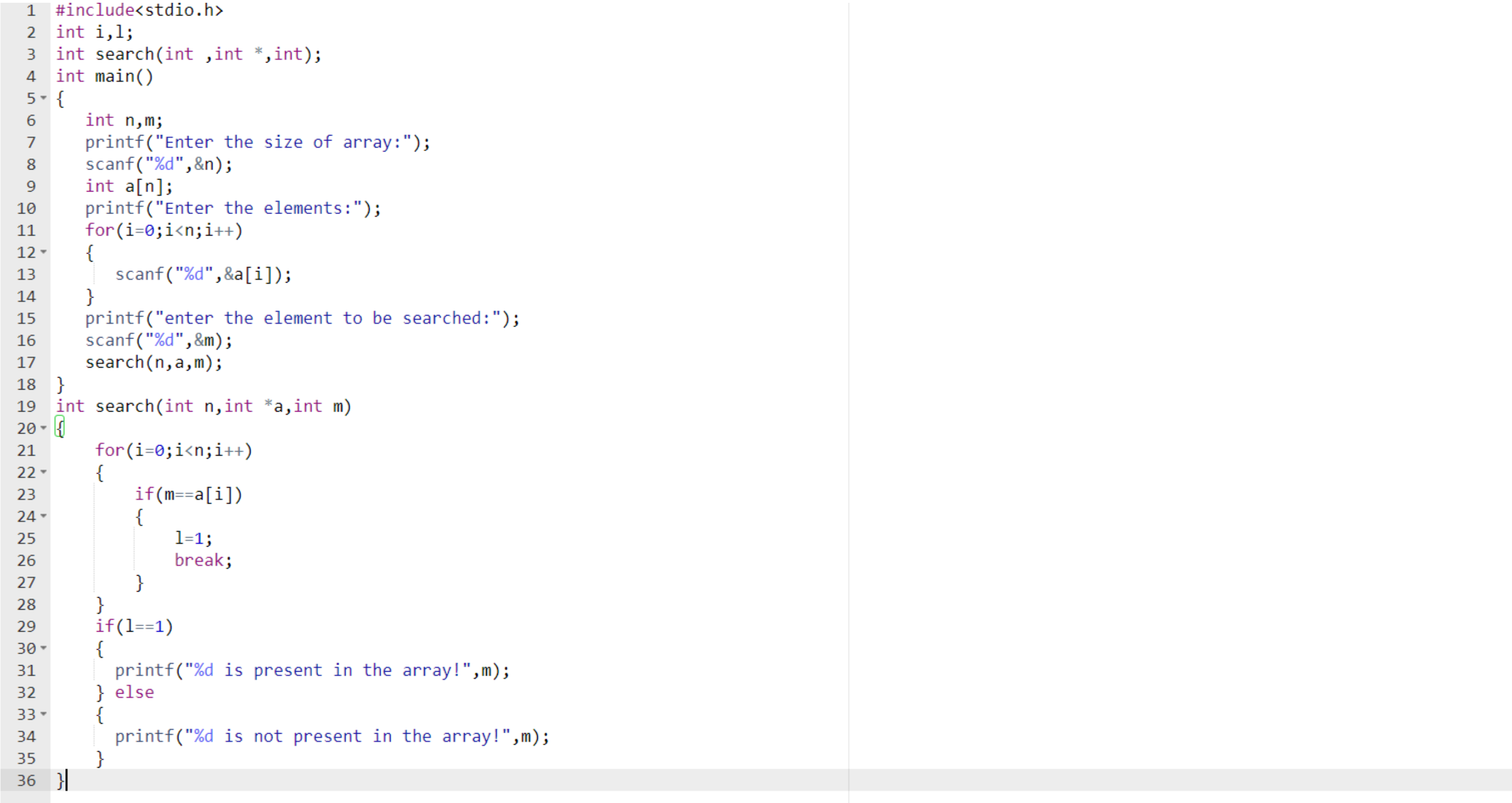


```
#include <stdio.h>
#define MAX_SIZE 100
void printArr(int *arr, int size);
int main()
{
    int arr[MAX_SIZE];
    int size;
    int *left = arr;
    int *right;
    printf("Enter size of array: ");
    scanf("%d", &size);
    right = &arr[size - 1];
    printf("Enter elements in array: ");
    while(left <= right){
        scanf("%d", left++);
    }
    printf("\nArray before reverse: ");
    printArr(arr, size);
    left = arr;
    while(left < right) {
        *left ^= *right;
        *right ^= *left;
        *left ^= *right;
        left++;
        right--;
    }
    printf("\nArray after reverse: ");
    printArr(arr, size);
}
void printArr(int * arr, int size)
{
    int * arrEnd = (arr + size - 1);
    while(arr <= arrEnd)
    {
        printf("%d, ", *arr);
        arr++;
    }
}
```

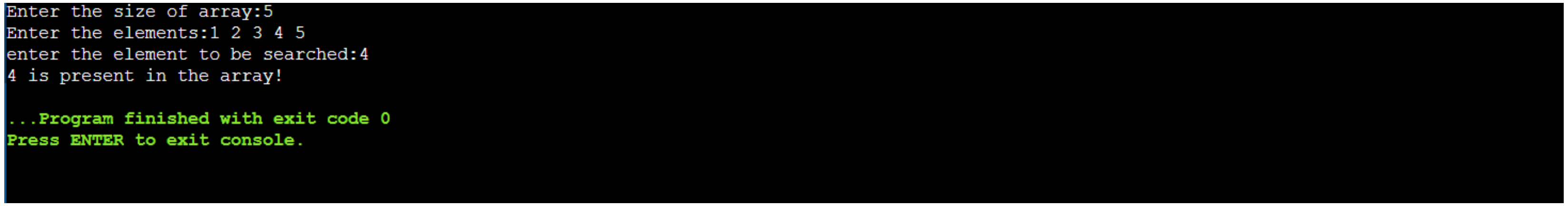


```
Enter size of array: 5
Enter elements in array: 1 2 3 4 5

Array before reverse: 1, 2, 3, 4, 5,
Array after reverse: 5, 4, 3, 2, 1,

...Program finished with exit code 0
Press ENTER to exit console.
```

C program to search an element in array using pointers.

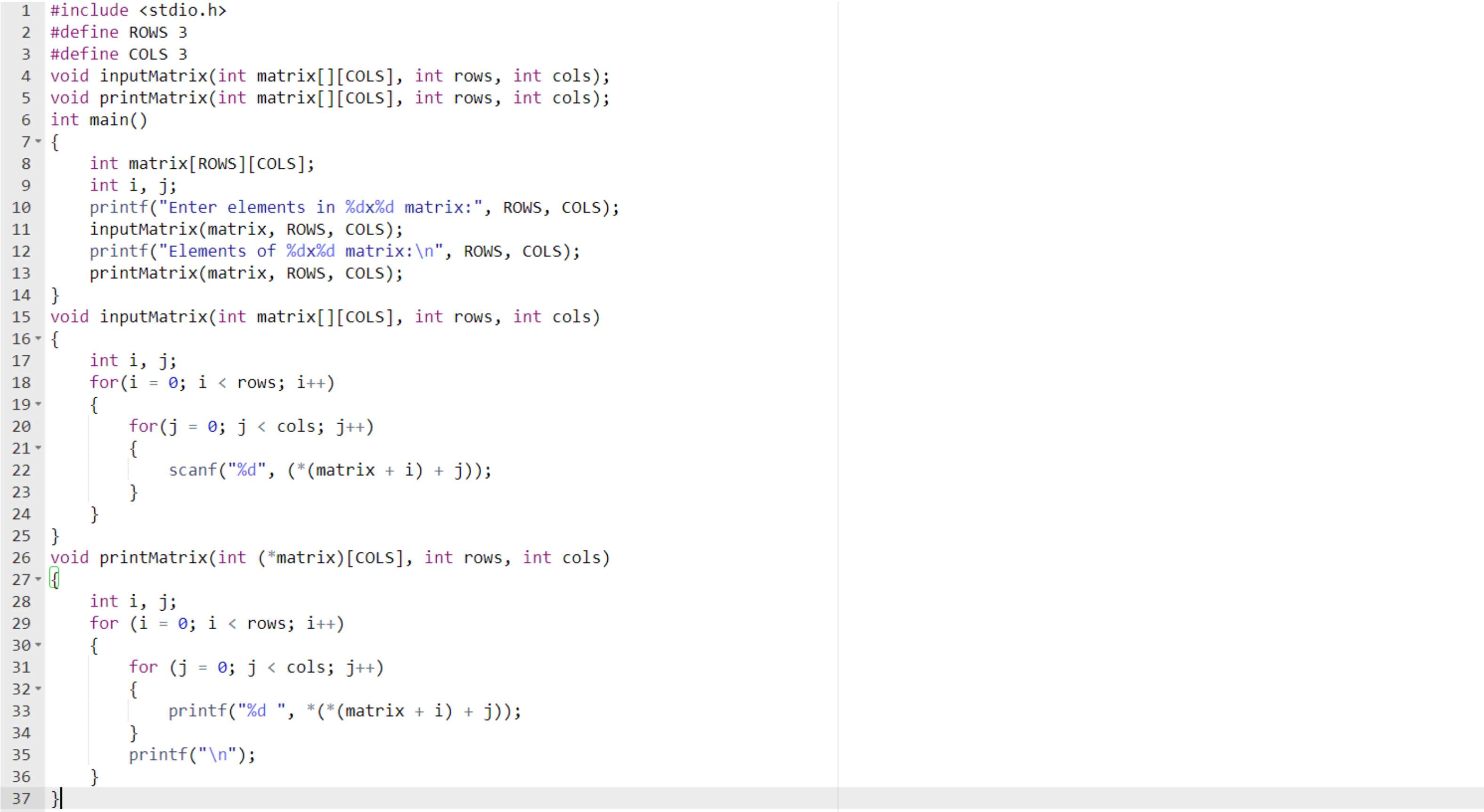


```c
#include<stdio.h>
int i,l;
int search(int ,int *,int);
int main()
{
   int n,m;
   printf("Enter the size of array:");
   scanf("%d",&n);
   int a[n];
   printf("Enter the elements:");
   for(i=0;i<n;i++)
   {
      scanf("%d",&a[i]);
   }
   printf("enter the element to be searched:");
   scanf("%d",&m);
   search(n,a,m);
}
int search(int n,int *a,int m)
{
    for(i=0;i<n;i++)
    {
        if(m==a[i])
        {
            l=1;
            break;
        }
    }
    if(l==1)
    {
      printf("%d is present in the array!",m);
    } else
    {
      printf("%d is not present in the array!",m);
    }
}
```

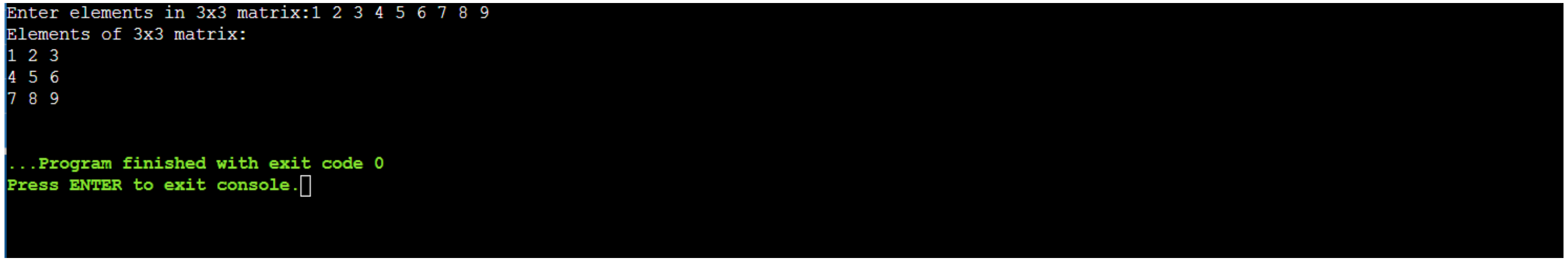


```
Enter the size of array:5
Enter the elements:1 2 3 4 5
enter the element to be searched:4
4 is present in the array!

...Program finished with exit code 0
Press ENTER to exit console.
```

C program to access two-dimensional array using pointers.

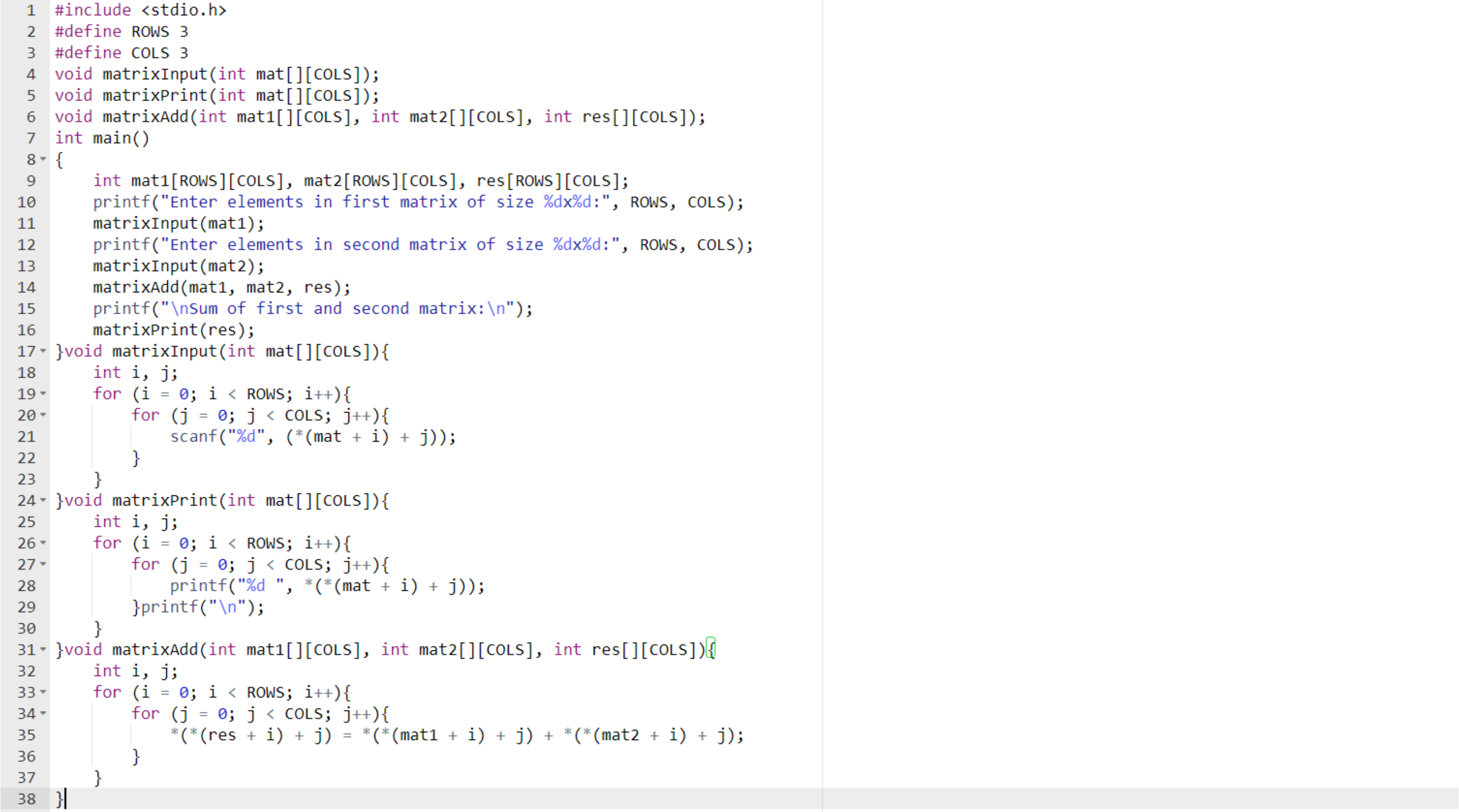


```c
#include <stdio.h>
#define ROWS 3
#define COLS 3
void inputMatrix(int matrix[][COLS], int rows, int cols);
void printMatrix(int matrix[][COLS], int rows, int cols);
int main()
{
    int matrix[ROWS][COLS];
    int i, j;
    printf("Enter elements in %dx%d matrix:", ROWS, COLS);
    inputMatrix(matrix, ROWS, COLS);
    printf("Elements of %dx%d matrix:\n", ROWS, COLS);
    printMatrix(matrix, ROWS, COLS);
}
void inputMatrix(int matrix[][COLS], int rows, int cols)
{
    int i, j;
    for(i = 0; i < rows; i++)
    {
        for(j = 0; j < cols; j++)
        {
            scanf("%d", (*(matrix + i) + j));
        }
    }
}
void printMatrix(int (*matrix)[COLS], int rows, int cols)
{
    int i, j;
    for (i = 0; i < rows; i++)
    {
        for (j = 0; j < cols; j++)
        {
            printf("%d ", *(*(matrix + i) + j));
        }
        printf("\n");
    }
}
```

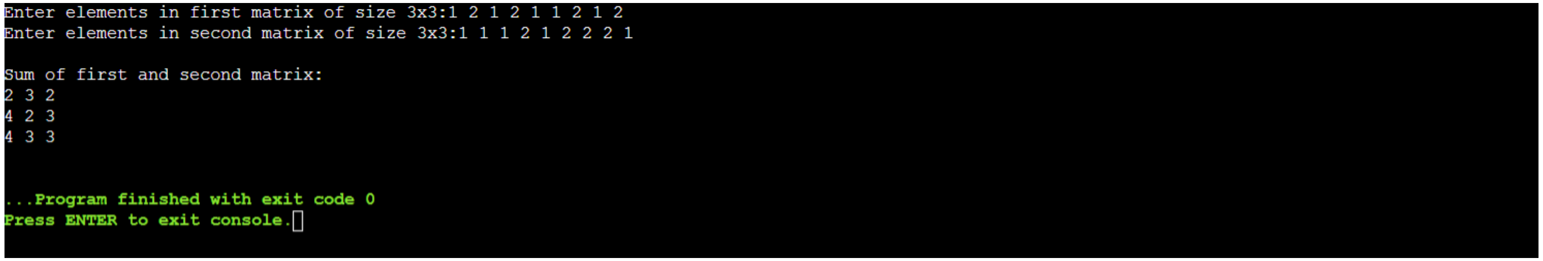


```
Enter elements in 3x3 matrix:1 2 3 4 5 6 7 8 9
Elements of 3x3 matrix:
1 2 3
4 5 6
7 8 9


...Program finished with exit code 0
Press ENTER to exit console.
```

C program to add two matrix using pointers.

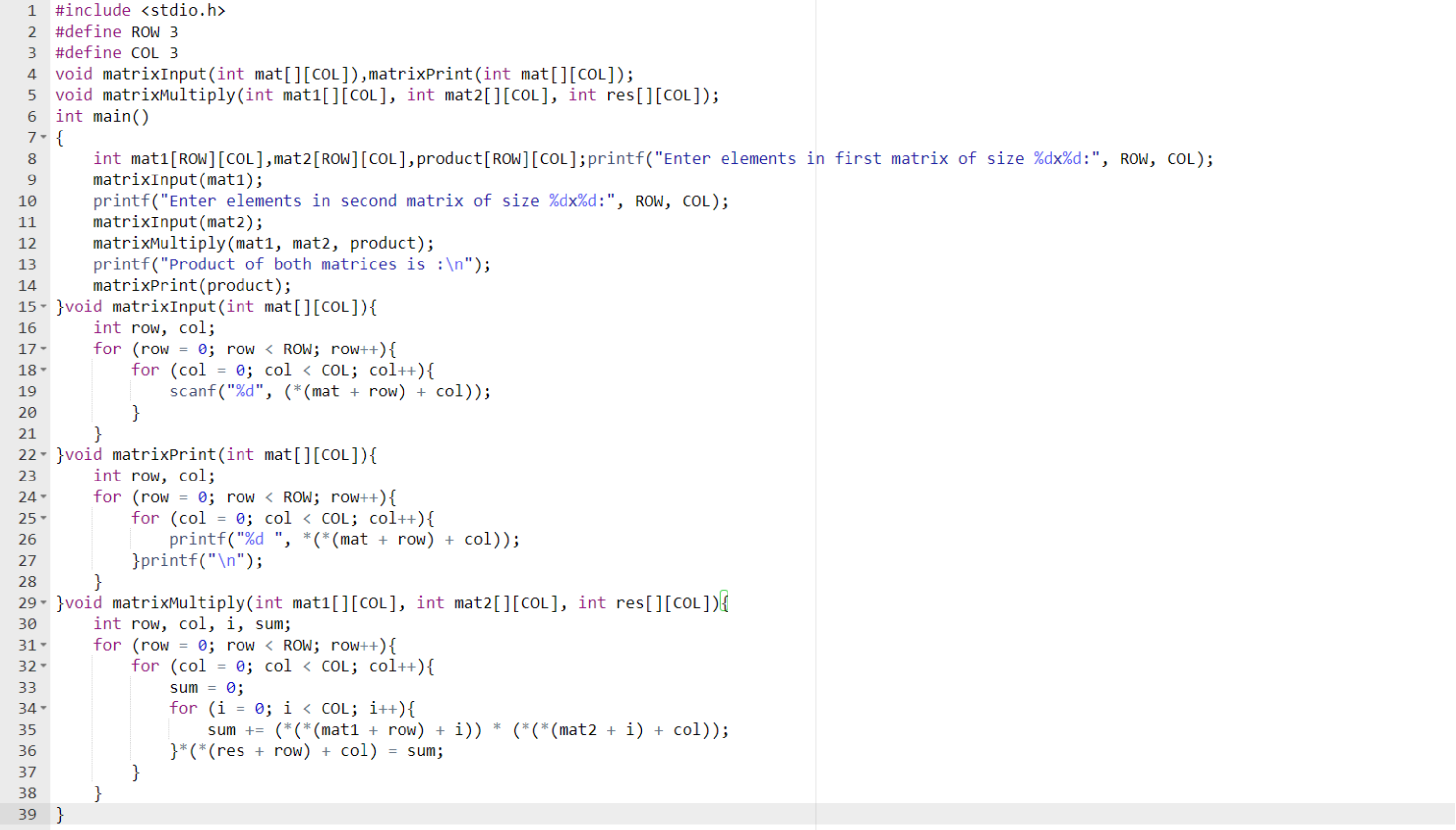


```c
#include <stdio.h>
#define ROWS 3
#define COLS 3
void matrixInput(int mat[][COLS]);
void matrixPrint(int mat[][COLS]);
void matrixAdd(int mat1[][COLS], int mat2[][COLS], int res[][COLS]);
int main()
{
    int mat1[ROWS][COLS], mat2[ROWS][COLS], res[ROWS][COLS];
    printf("Enter elements in first matrix of size %dx%d:", ROWS, COLS);
    matrixInput(mat1);
    printf("Enter elements in second matrix of size %dx%d:", ROWS, COLS);
    matrixInput(mat2);
    matrixAdd(mat1, mat2, res);
    printf("\nSum of first and second matrix:\n");
    matrixPrint(res);
}void matrixInput(int mat[][COLS]){
    int i, j;
    for (i = 0; i < ROWS; i++){
        for (j = 0; j < COLS; j++){
            scanf("%d", (*(mat + i) + j));
        }
    }
}void matrixPrint(int mat[][COLS]){
    int i, j;
    for (i = 0; i < ROWS; i++){
        for (j = 0; j < COLS; j++){
            printf("%d ", *(*(mat + i) + j));
        }printf("\n");
    }
}void matrixAdd(int mat1[][COLS], int mat2[][COLS], int res[][COLS]){
    int i, j;
    for (i = 0; i < ROWS; i++){
        for (j = 0; j < COLS; j++){
            *(*(res + i) + j) = *(*(mat1 + i) + j) + *(*(mat2 + i) + j);
        }
    }
}
```

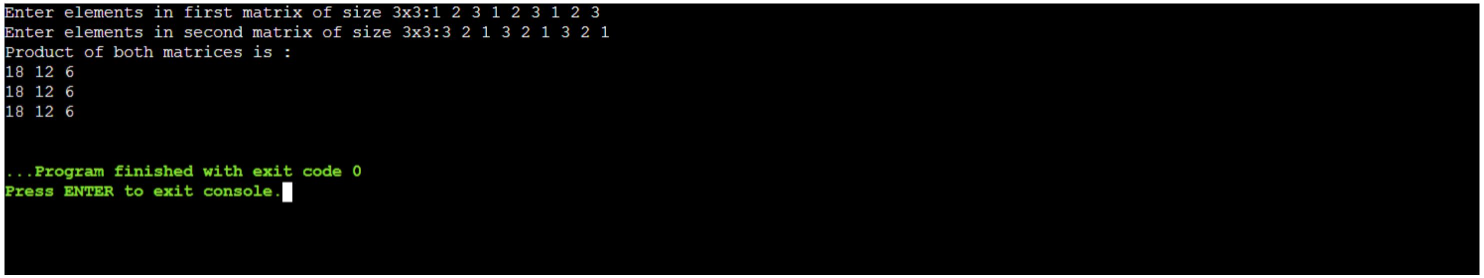


```
Enter elements in first matrix of size 3x3:1 2 1 2 1 1 2 1 2
Enter elements in second matrix of size 3x3:1 1 1 2 1 2 2 2 1

Sum of first and second matrix:
2 3 2
4 2 3
4 3 3


...Program finished with exit code 0
Press ENTER to exit console.
```

C program to multiply two matrix using pointers.

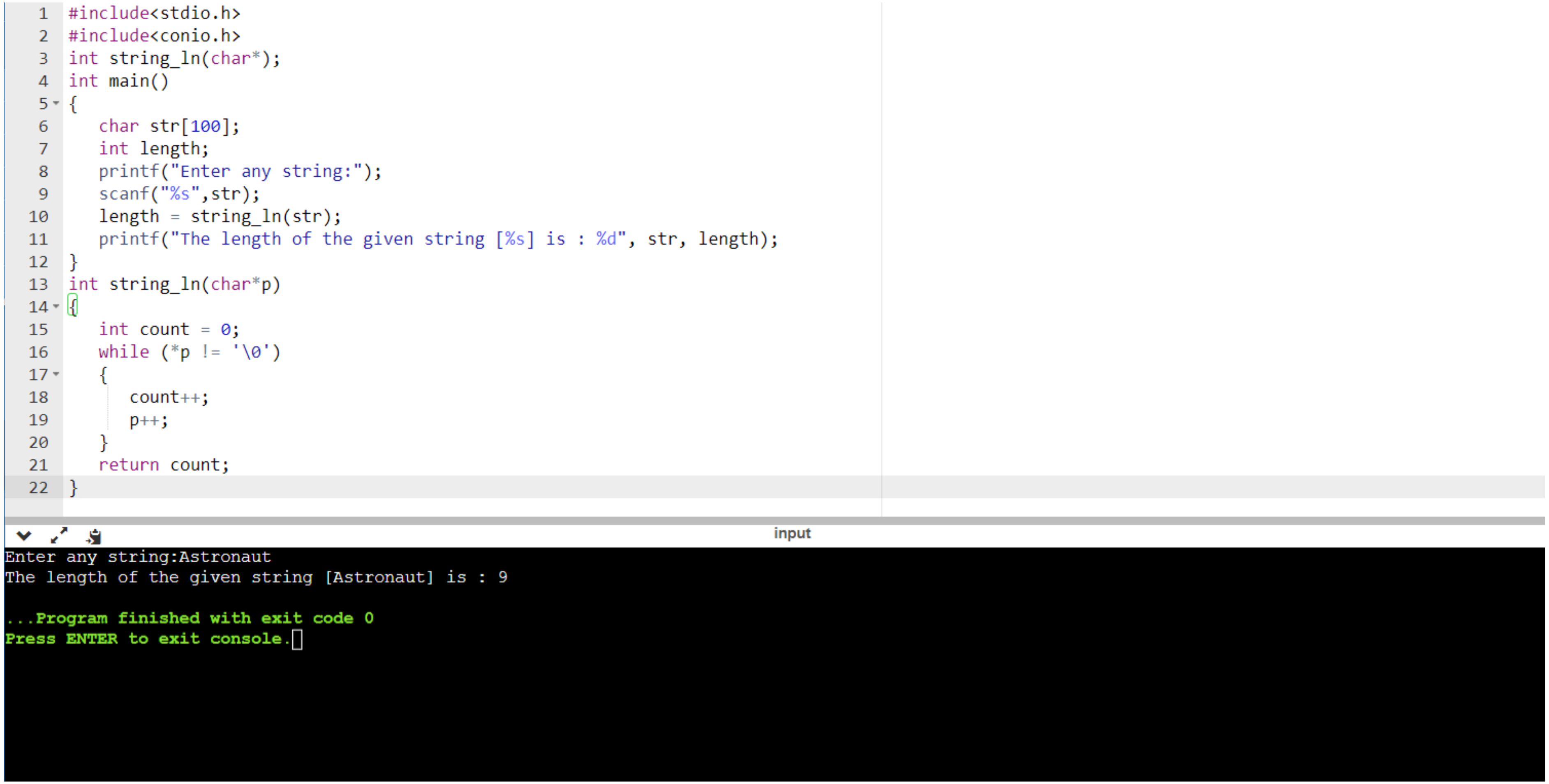


```c
#include <stdio.h>
#define ROW 3
#define COL 3
void matrixInput(int mat[][COL]),matrixPrint(int mat[][COL]);
void matrixMultiply(int mat1[][COL], int mat2[][COL], int res[][COL]);
int main()
{
    int mat1[ROW][COL],mat2[ROW][COL],product[ROW][COL];printf("Enter elements in first matrix of size %dx%d:", ROW, COL);
    matrixInput(mat1);
    printf("Enter elements in second matrix of size %dx%d:", ROW, COL);
    matrixInput(mat2);
    matrixMultiply(mat1, mat2, product);
    printf("Product of both matrices is :\n");
    matrixPrint(product);
}void matrixInput(int mat[][COL]){
    int row, col;
    for (row = 0; row < ROW; row++){
        for (col = 0; col < COL; col++){
            scanf("%d", (*(mat + row) + col));
        }
    }
}void matrixPrint(int mat[][COL]){
    int row, col;
    for (row = 0; row < ROW; row++){
        for (col = 0; col < COL; col++){
            printf("%d ", *(*(mat + row) + col));
        }printf("\n");
    }
}void matrixMultiply(int mat1[][COL], int mat2[][COL], int res[][COL]){
    int row, col, i, sum;
    for (row = 0; row < ROW; row++){
        for (col = 0; col < COL; col++){
            sum = 0;
            for (i = 0; i < COL; i++){
                sum += (*(*(mat1 + row) + i)) * (*(*(mat2 + i) + col));
            }*(*(res + row) + col) = sum;
        }
    }
}
```



```
Enter elements in first matrix of size 3x3:1 2 3 1 2 3 1 2 3
Enter elements in second matrix of size 3x3:3 2 1 3 2 1 3 2 1
Product of both matrices is :
18 12 6
18 12 6
18 12 6


...Program finished with exit code 0
Press ENTER to exit console.
```

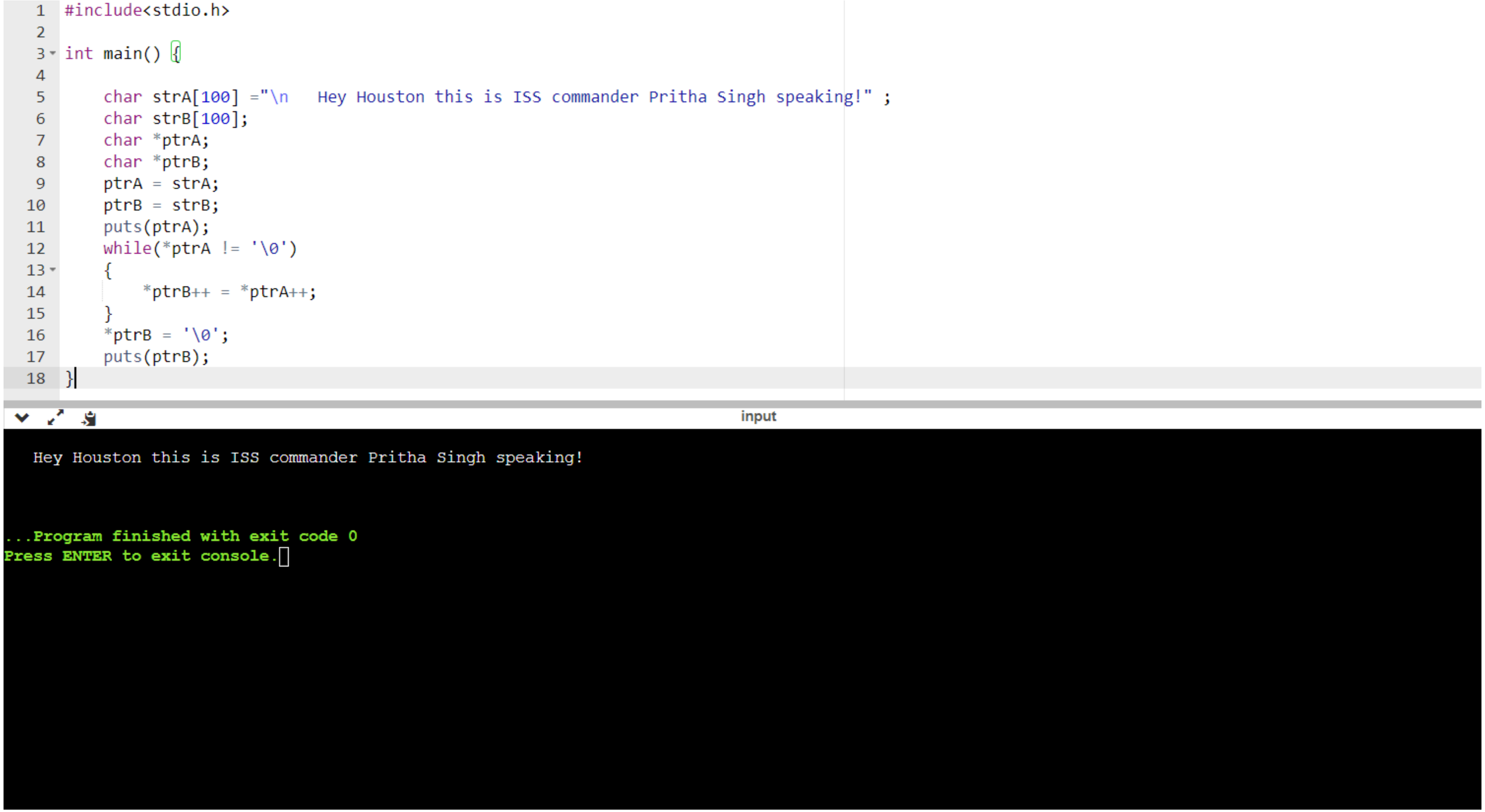
C program to find length of string using pointers.



```c
#include<stdio.h>
#include<conio.h>
int string_ln(char*);
int main()
{
    char str[100];
    int length;
    printf("Enter any string:");
    scanf("%s",str);
    length = string_ln(str);
    printf("The length of the given string [%s] is : %d", str, length);
}
int string_ln(char*p)
{
    int count = 0;
    while (*p != '\0')
    {
        count++;
        p++;
    }
    return count;
}
```

```
Enter any string:Astronaut
The length of the given string [Astronaut] is : 9

...Program finished with exit code 0
Press ENTER to exit console.
```

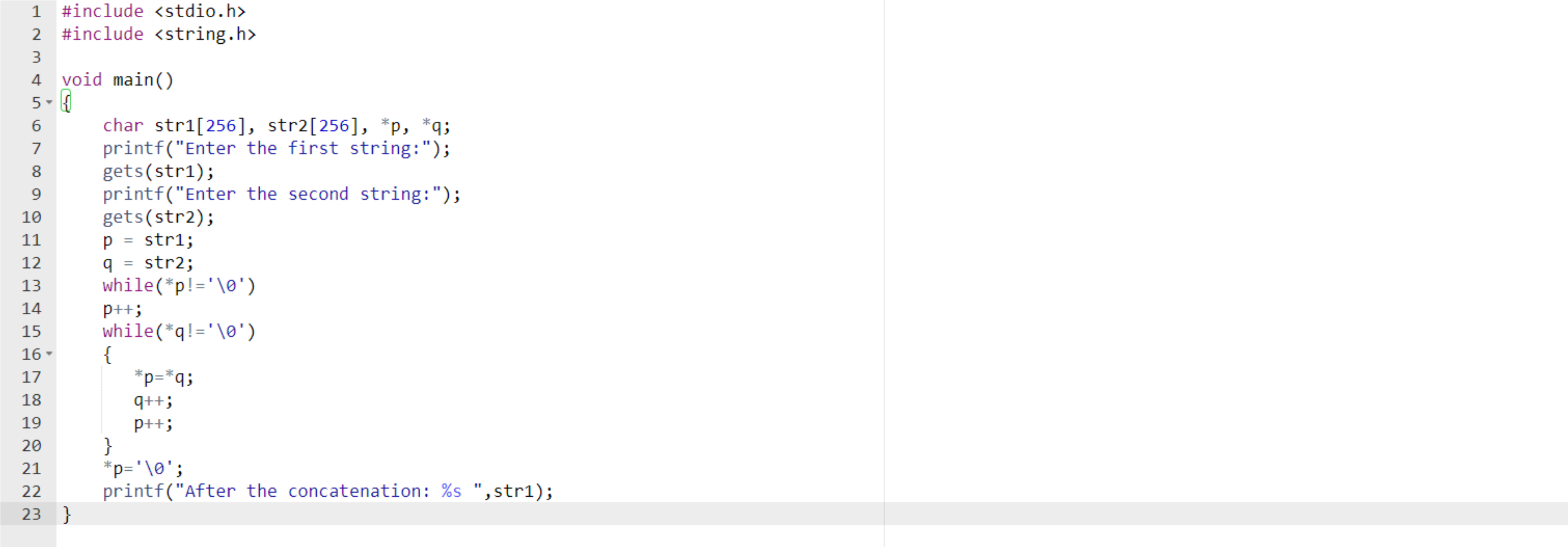
C program to copy one string to another using pointers.



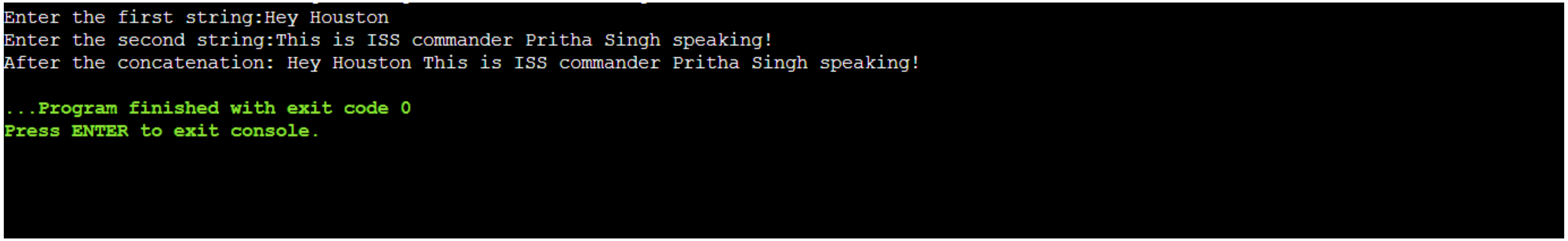
```c
#include<stdio.h>

int main() {

    char strA[100] ="\n   Hey Houston this is ISS commander Pritha Singh speaking!" ;
    char strB[100];
    char *ptrA;
    char *ptrB;
    ptrA = strA;
    ptrB = strB;
    puts(ptrA);
    while(*ptrA != '\0')
    {
        *ptrB++ = *ptrA++;
    }
    *ptrB = '\0';
    puts(ptrB);
}
```

```
   Hey Houston this is ISS commander Pritha Singh speaking!


...Program finished with exit code 0
Press ENTER to exit console.
```

C program to concatenate two strings using pointers.



```c
#include <stdio.h>
#include <string.h>

void main()
{
    char str1[256], str2[256], *p, *q;
    printf("Enter the first string:");
    gets(str1);
    printf("Enter the second string:");
    gets(str2);
    p = str1;
    q = str2;
    while(*p!='\0')
    p++;
    while(*q!='\0')
    {
       *p=*q;
       q++;
       p++;
    }
    *p='\0';
    printf("After the concatenation: %s ",str1);
}
```



```
Enter the first string:Hey Houston
Enter the second string:This is ISS commander Pritha Singh speaking!
After the concatenation: Hey Houston This is ISS commander Pritha Singh speaking!

...Program finished with exit code 0
Press ENTER to exit console.
```