

C program to create, initialize and use pointers.

```
1 #include <stdio.h>
2
3 int main()
4 {
5     char cha;
6     char *ptrCha;
7     ptrCha = &cha;
8     cha = 'A';
9     printf("The Value of character: %c\n", cha);
10    printf("The Address of character: %p\n", &cha);
11    printf("The Value of character: %c\n", *ptrCha);
12    printf("The Address of character: %p", ptrCha);
13 }
```

input

```
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.

```
1 #include <stdio.h>
2 int main()
3 {
4     int first, second, *p, *q, sum;
5
6     printf("Enter two integers to add:");
7     scanf("%d%d", &first, &second);
8
9     p = &first;
10    q = &second;
11
12    sum = *p + *q;
13    printf("Sum of the numbers = %d\n", sum);
14 }
```

input

```
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.

```
1  #include<stdio.h>
2  void swap(int*, int*);
3  int main()
4  {
5      int a, b;
6      printf("Enter values for a and b:");
7      scanf("%d%d", &a, &b);
8      printf("Before swapping: a = %d and b = %d\n", a, b);
9      swap(&a, &b);
10     printf("After swapping: a = %d and b = %d", a, b);
11     return 0;
12 }
13 void swap(int *x, int *y)
14 {
15     int temp;
16
17     temp = *x;
18     *x   = *y;
19     *y   = temp;
20 }
21
```

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.

```
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 int main()
4 {
5     int arr[MAX_SIZE];
6     int N, i;
7     int * ptr = arr;
8     printf("Enter size of array: ");
9     scanf("%d", &N);
10    printf("Enter elements in array:\n");
11    for (i = 0; i < N; i++)
12    {
13        scanf("%d", ptr);
14        ptr++;
15    }
16    ptr = arr;
17    printf("Array elements: ");
18    for (i = 0; i < N; i++)
19    {
20        printf("%d, ", *ptr);
21        ptr++;
22    }
23 }
```

input

```
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.

```
1 #include <stdio.h>
2 #define max_size 50
3 void display_arr(int *arr, int size)
4 {
5     int i;
6     for(i = 0; i <= size; i++)
7     {
8         printf("%d ", *(arr + i));
9     }
10 }
11 int main()
12 {
13     int arr1[max_size], arr2[max_size];
14     int size, i;
15     int *ptr1 = arr1;
16     int *ptr2 = arr2;
17     int *last_arr = arr1+9;
18     printf("Enter size of array: ");
19     scanf("%d", &size);
20     printf("Please input arr1 elements: ");
21     for(i = 0; i <= size; i++)
22     {
23         scanf("%d", ptr1 + i);
24     }
25     printf("\narr1 elements before coping: ");
26     display_arr(arr1, size);
27     printf("\narr2 elements before coping: ");
28     display_arr(arr2, size);
29     while(ptr1 <= last_arr)
30     {
31         *ptr2 = *ptr1;
32         ptr1++;
33         ptr2++;
34     }
35     printf("\narr1 elements after coping: ");
36     display_arr(arr1, size);
37     printf("\narr2 elements after coping: ");
38     display_arr(arr2, size);
39 }
```

```
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.

```
1  #include <stdio.h>
2  #define MAX_SIZE 100
3  void inputArray(int *arr, int size);
4  void printArray(int *arr, int size);
5  void swapArray(int *sourceArr, int *destArr, int size);
6  int main()
7  {
8      int sourceArr[MAX_SIZE];
9      int destArr[MAX_SIZE];
10     int size;
11     printf("Enter size of array: ");
12     scanf("%d", &size);
13     printf("Enter %d elements in source array: ", size);
14     inputArray(sourceArr, size);
15     printf("Enter %d elements in destination array: ", size);
16     inputArray(destArr, size);
17     printf("\n\nSource array before swapping: ");
18     printArray(sourceArr, size);
19     printf("\n\nDestination array before swapping: ");
20     printArray(destArr, size);
21     swapArray(sourceArr, destArr, size);
22     printf("\n\nSource array after swapping: ");
23     printArray(sourceArr, size);
24     printf("\n\nDestination array after swapping: ");
25     printArray(destArr, size);
26 }
27 void inputArray(int *arr, int size){
28     int *arrEnd = (arr + (size - 1));
29
30     while(arr <= arrEnd)
31         scanf("%d", arr++);
32 }
33 void printArray(int *arr, int size){
34     int *arrEnd = (arr + (size - 1));
35     while(arr <= arrEnd)
36         printf("%d, ", *(arr++));
37 }
38 void swapArray(int * sourceArr, int * destArr, int size){
39     int * sourceArrEnd = (sourceArr + (size - 1));
40
41     int * destArrEnd = (destArr + (size - 1));
42     while(sourceArr <= sourceArrEnd && destArr <= destArrEnd){
43         *sourceArr ^= *destArr;
44         *destArr ^= *sourceArr;
45         *sourceArr ^= *destArr;
46         sourceArr++;
47         destArr++;
48     }
```

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.

```
1 #include <stdio.h>
2 #define MAX_SIZE 100
3 void printArr(int *arr, int size);
4 int main()
5 {
6     int arr[MAX_SIZE];
7     int size;
8     int *left = arr;
9     int *right;
10    printf("Enter size of array: ");
11    scanf("%d", &size);
12    right = &arr[size - 1];
13    printf("Enter elements in array: ");
14    while(left <= right){
15        scanf("%d", left++);
16    }
17    printf("\nArray before reverse: ");
18    printArr(arr, size);
19    left = arr;
20    while(left < right) {
21        *left ^= *right;
22        *right ^= *left;
23        *left ^= *right;
24        left++;
25        right--;
26    }
27    printf("\nArray after reverse: ");
28    printArr(arr, size);
29 }
30 void printArr(int * arr, int size)
31 {
32     int * arrEnd = (arr + size - 1);
33     while(arr <= arrEnd)
34     {
35         printf("%d, ", *arr);
36         arr++;
37     }
38 }
```

```
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.

```
1  #include<stdio.h>
2  int i,l;
3  int search(int ,int *,int);
4  int main()
5  {
6      int n,m;
7      printf("Enter the size of array:");
8      scanf("%d",&n);
9      int a[n];
10     printf("Enter the elements:");
11     for(i=0;i<n;i++)
12     {
13         scanf("%d",&a[i]);
14     }
15     printf("enter the element to be searched:");
16     scanf("%d",&m);
17     search(n,a,m);
18 }
19 int search(int n,int *a,int m)
20 {
21     for(i=0;i<n;i++)
22     {
23         if(m==a[i])
24         {
25             l=1;
26             break;
27         }
28     }
29     if(l==1)
30     {
31         printf("%d is present in the array!",m);
32     } else
33     {
34         printf("%d is not present in the array!",m);
35     }
36 }
```

```
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.

```
1 #include <stdio.h>
2 #define ROWS 3
3 #define COLS 3
4 void inputMatrix(int matrix[][COLS], int rows, int cols);
5 void printMatrix(int matrix[][COLS], int rows, int cols);
6 int main()
7 {
8     int matrix[ROWS][COLS];
9     int i, j;
10    printf("Enter elements in %dx%d matrix:", ROWS, COLS);
11    inputMatrix(matrix, ROWS, COLS);
12    printf("Elements of %dx%d matrix:\n", ROWS, COLS);
13    printMatrix(matrix, ROWS, COLS);
14 }
15 void inputMatrix(int matrix[][COLS], int rows, int cols)
16 {
17     int i, j;
18     for(i = 0; i < rows; i++)
19     {
20         for(j = 0; j < cols; j++)
21         {
22             scanf("%d", (*(matrix + i) + j));
23         }
24     }
25 }
26 void printMatrix(int (*matrix)[COLS], int rows, int cols)
27 {
28     int i, j;
29     for (i = 0; i < rows; i++)
30     {
31         for (j = 0; j < cols; j++)
32         {
33             printf("%d ", (*(matrix + i) + j));
34         }
35         printf("\n");
36     }
37 }
```

```
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.

```
1 #include <stdio.h>
2 #define ROWS 3
3 #define COLS 3
4 void matrixInput(int mat[][COLS]);
5 void matrixPrint(int mat[][COLS]);
6 void matrixAdd(int mat1[][COLS], int mat2[][COLS], int res[][COLS]);
7 int main()
8 {
9     int mat1[ROWS][COLS], mat2[ROWS][COLS], res[ROWS][COLS];
10    printf("Enter elements in first matrix of size %dx%d:", ROWS, COLS);
11    matrixInput(mat1);
12    printf("Enter elements in second matrix of size %dx%d:", ROWS, COLS);
13    matrixInput(mat2);
14    matrixAdd(mat1, mat2, res);
15    printf("\nSum of first and second matrix:\n");
16    matrixPrint(res);
17 } void matrixInput(int mat[][COLS]){
18     int i, j;
19     for (i = 0; i < ROWS; i++){
20         for (j = 0; j < COLS; j++){
21             scanf("%d", (*(mat + i) + j));
22         }
23     }
24 } void matrixPrint(int mat[][COLS]){
25     int i, j;
26     for (i = 0; i < ROWS; i++){
27         for (j = 0; j < COLS; j++){
28             printf("%d ", (*(mat + i) + j));
29         } printf("\n");
30     }
31 } void matrixAdd(int mat1[][COLS], int mat2[][COLS], int res[][COLS]){
32     int i, j;
33     for (i = 0; i < ROWS; i++){
34         for (j = 0; j < COLS; j++){
35             (*(res + i) + j) = (*(mat1 + i) + j) + (*(mat2 + i) + j);
36         }
37     }
38 }
```

```
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 2 1 2 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.

```
1 #include <stdio.h>
2 #define ROW 3
3 #define COL 3
4 void matrixInput(int mat[][COL]),matrixPrint(int mat[][COL]);
5 void matrixMultiply(int mat1[][COL], int mat2[][COL], int res[][COL]);
6 int main()
7 {
8     int mat1[ROW][COL],mat2[ROW][COL],product[ROW][COL];printf("Enter elements in first matrix of size %dx%d:", ROW, COL);
9     matrixInput(mat1);
10    printf("Enter elements in second matrix of size %dx%d:", ROW, COL);
11    matrixInput(mat2);
12    matrixMultiply(mat1, mat2, product);
13    printf("Product of both matrices is :\n");
14    matrixPrint(product);
15 }void matrixInput(int mat[][COL]){
16     int row, col;
17     for (row = 0; row < ROW; row++){
18         for (col = 0; col < COL; col++){
19             scanf("%d", (*(mat + row) + col));
20         }
21     }
22 }void matrixPrint(int mat[][COL]){
23     int row, col;
24     for (row = 0; row < ROW; row++){
25         for (col = 0; col < COL; col++){
26             printf("%d ", (*(mat + row) + col));
27         }printf("\n");
28     }
29 }void matrixMultiply(int mat1[][COL], int mat2[][COL], int res[][COL]){
30     int row, col, i, sum;
31     for (row = 0; row < ROW; row++){
32         for (col = 0; col < COL; col++){
33             sum = 0;
34             for (i = 0; i < COL; i++){
35                 sum += (*(mat1 + row) + i) * (*(mat2 + i) + col);
36             }*(res + row) + col = sum;
37         }
38     }
39 }
```

```
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.

```
1 #include<stdio.h>
2 #include<conio.h>
3 int string_ln(char*);
4 int main()
5 {
6     char str[100];
7     int length;
8     printf("Enter any string:");
9     scanf("%s",str);
10    length = string_ln(str);
11    printf("The length of the given string [%s] is : %d", str, length);
12 }
13 int string_ln(char*p)
14 {
15     int count = 0;
16     while (*p != '\0')
17     {
18         count++;
19         p++;
20     }
21     return count;
22 }
```

input

```
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.

```
1  #include<stdio.h>
2
3  int main() {
4
5      char strA[100] = "\n  Hey Houston this is ISS commander Pritha Singh speaking!" ;
6      char strB[100];
7      char *ptrA;
8      char *ptrB;
9      ptrA = strA;
10     ptrB = strB;
11     puts(ptrA);
12     while(*ptrA != '\0')
13     {
14         *ptrB++ = *ptrA++;
15     }
16     *ptrB = '\0';
17     puts(ptrB);
18 }
```

input

```
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.

```
1 #include <stdio.h>
2 #include <string.h>
3
4 void main()
5 {
6     char str1[256], str2[256], *p, *q;
7     printf("Enter the first string:");
8     gets(str1);
9     printf("Enter the second string:");
10    gets(str2);
11    p = str1;
12    q = str2;
13    while(*p!='\0')
14        p++;
15    while(*q!='\0')
16    {
17        *p=*q;
18        q++;
19        p++;
20    }
21    *p='\0';
22    printf("After the concatenation: %s ",str1);
23 }
```

```
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.
```