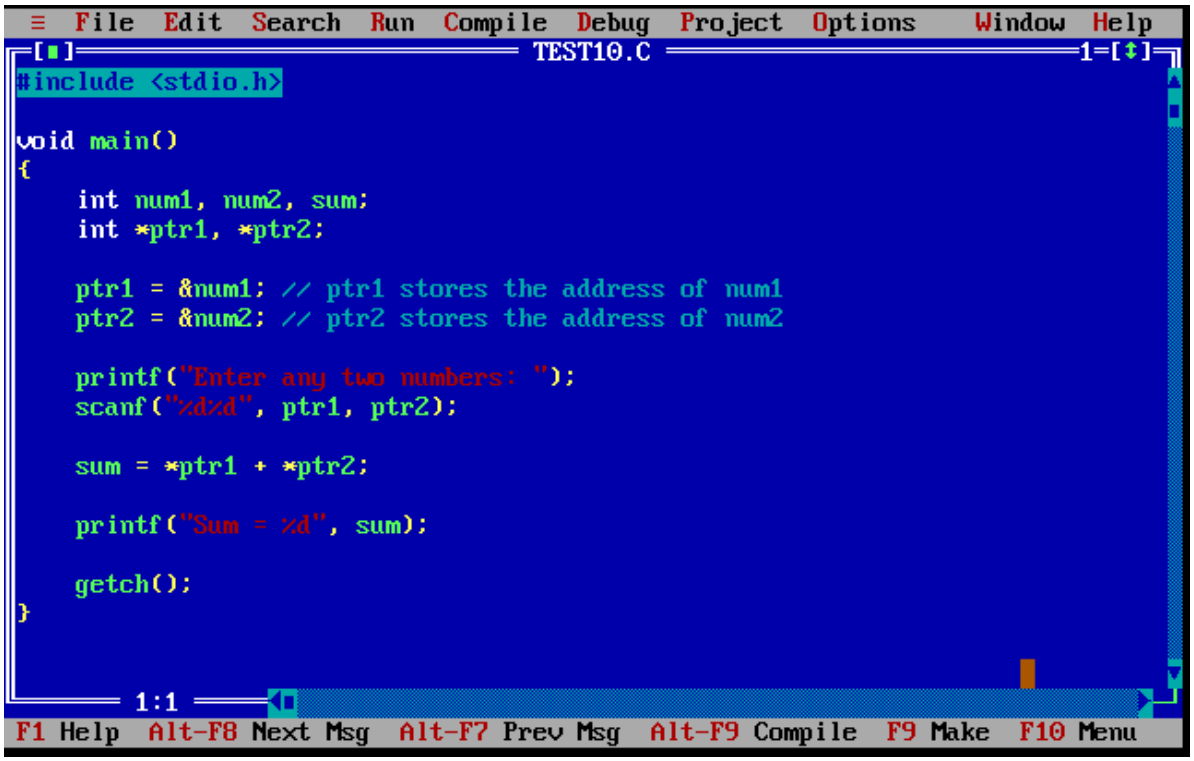
Pointer Basics Revision

Q. Write a program to add two numbers by using Pointers.

Ans.



You have noticed that, I haven't used any & (address of) operator in the scanf() function. scanf() takes the actual memory address where the user input is to be stored.

The pointer variable ptr1 and ptr2 contains the actual memory addresses of num1 and num2. Therefore we need not to prefix the & operator in scanf() function in case of pointers.

Now using this, it should be an easy workaround to compute all arithmetic operations using pointers. Let us write another program that performs all arithmetic operations using pointers.

Program to perform all arithmetic operations using pointers

```
#include <stdio.h>

void main()

{

   float num1, num2;    // Normal variables

   float *ptr1, *ptr2;  // Pointer variables

   float sum, diff, mult, div;

   ptr1 = &num1; // ptr1 stores the address of num1

   ptr2 = &num2; // ptr2 stores the address of num2

   /* User input through pointer */

   printf("Enter any two  numbers: ");
```

```
    scanf("%f %f", ptr1, ptr2);
    /* Perform arithmetic operation */
    sum  = (*ptr1) + (*ptr2);
    diff = (*ptr1) - (*ptr2);
    mult = (*ptr1) * (*ptr2);
    div  = (*ptr1) / (*ptr2);
    /* Print the results */
    printf("Sum = %.2f\n", sum);
    printf("Difference = %.2f\n", diff);
    printf("Product = %.2f\n", mult);
    printf("Quotient = %.2f\n", div);
    getch();
}
```

Write a C program to swap two numbers using pointers and functions. How to swap two numbers using call by reference method. Logic to swap two number using pointers in C program.

Logic to swap two numbers using call by reference

Swapping two numbers is simple and a fundamental thing. You need not to know any rocket science for swapping two numbers. Simple swapping can be achieved in three steps -

- Copy the value of first number say num1 to some temporary variable say temp.
- Copy the value of second number say num2 to the first number. Which is num1 = num2.
- Copy back the value of first number stored in temp to second number. Which is num2 = temp.

Let us implement this logic using call by reference concept in functions.

```
#include <stdio.h>

/* Swap function declaration */
void swap(int * num1, int * num2);

void main()
{
    int num1, num2;

    /* Input numbers */
    printf("Enter two numbers: ");
    scanf("%d%d", &num1, &num2);

    /* Print original values of num1 and num2 */
    printf("Before swapping in main n");
    printf("Value of num1 = %d \n", num1);
```

```
    printf("Value of num2 = %d \n\n", num2);

    /* Pass the addresses of num1 and num2 */
    swap(&num1, &num2);

    /* Print the swapped values of num1 and num2 */
    printf("After swapping in main n");
    printf("Value of num1 = %d \n", num1);
    printf("Value of num2 = %d \n\n", num2);

    getch();
}


/**
 * Function to swap two numbers
 */
void swap(int * num1, int * num2)
{
    int temp;

    // Copy the value of num1 to some temp variable
    temp = *num1;

    // Copy the value of num2 to num1
    *num1= *num2;

    // Copy the value of num1 stored in temp to num2
    *num2= temp;
}
```

Write a C program to input elements in an array and print array using pointers. How to input and display array elements using pointer in C programming.

Example

Input

Input array size: 10

Input elements: 1

2

3

4

5

6

7

8

9

10

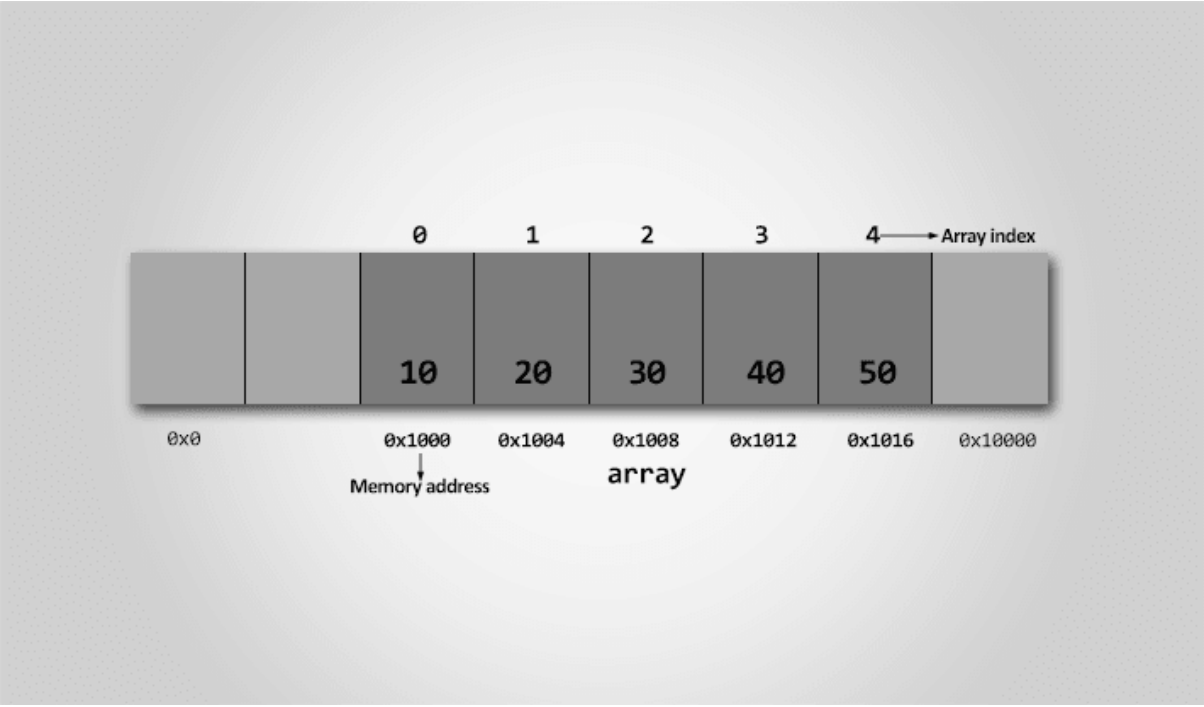
Output

Array elements: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

How to access array using pointer

Array elements in memory are stored sequentially. For example, consider the given array and its memory representation

int arr[] = {10, 20, 30, 40, 50};



If you have a pointer say ptr pointing at arr[0]. Then you can easily apply pointer arithmetic to get reference of next array element. You can either use (ptr + 1) or ptr++ to point to arr[1].