**Science College for Women/ Second Class/System programming/ Lec4**

**Lecture: Asraa A.H.**

## Pointer Arithmetic

Increment operator when used with a pointer variable returns next address pointed by the pointer. The next address returned is the sum of current pointed address and size of pointer data type. Or in simple terms, incrementing a pointer will cause the pointer to point to a memory location skipping **N bytes** from current pointed memory location. **Where N is size of pointer data type**.

Similarly, decrement operator returns the previous address pointed by the pointer. The returned address is the difference of current pointed address and size of pointer data type.

For example, consider the below statements::

**int** num = 5; // Suppose address of num = 0x1230

**int** \*ptr; // Pointer variable

ptr = &num; // ptr points to 0x1230 or ptr points to num

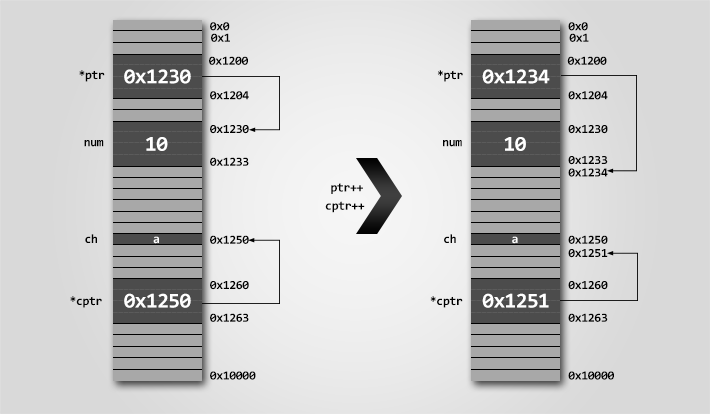
ptr++; // ptr now points to 0x1234, since integer size is 4 bytes

ptr--; // ptr now points to 0x1230

**Note:** Increment operation increments pointer address by the size of pointer data type.

If an integer pointer ptr pointing at 0x1230, after ptr++ it will point at 0x1234(assuming integer size is 4 bytes).

If a character pointer cptr pointing at 0x1250, after cptr++ it will point at 0x1251(since character occupies 1 byte).

Pointer increment decrement operation memory representation

**Pointer to Array**

We can also declare a pointer of any type  to point to the array .

int \*p;

p = arr;

// or

p = &arr[0]; //both the statements are equivalent.

we can use a pointer to point to an array and then used this pointer to access the array elements. Let's have an example::

#include <stdio.h>

void main()

{ int i;

int a[] = {1, 2, 3, 4, 5};

int \*p = a; // same as int\*p = &a[0]

printf(" addrass of array by pointer :: %u\n",p);

printf(" addrass of array by arry name ::%u\n",&a[0]);

for (i = 0; i < 5; i++)

{ printf(" the value %d ::", \*p);

printf("the addrass :: %u\n",p);

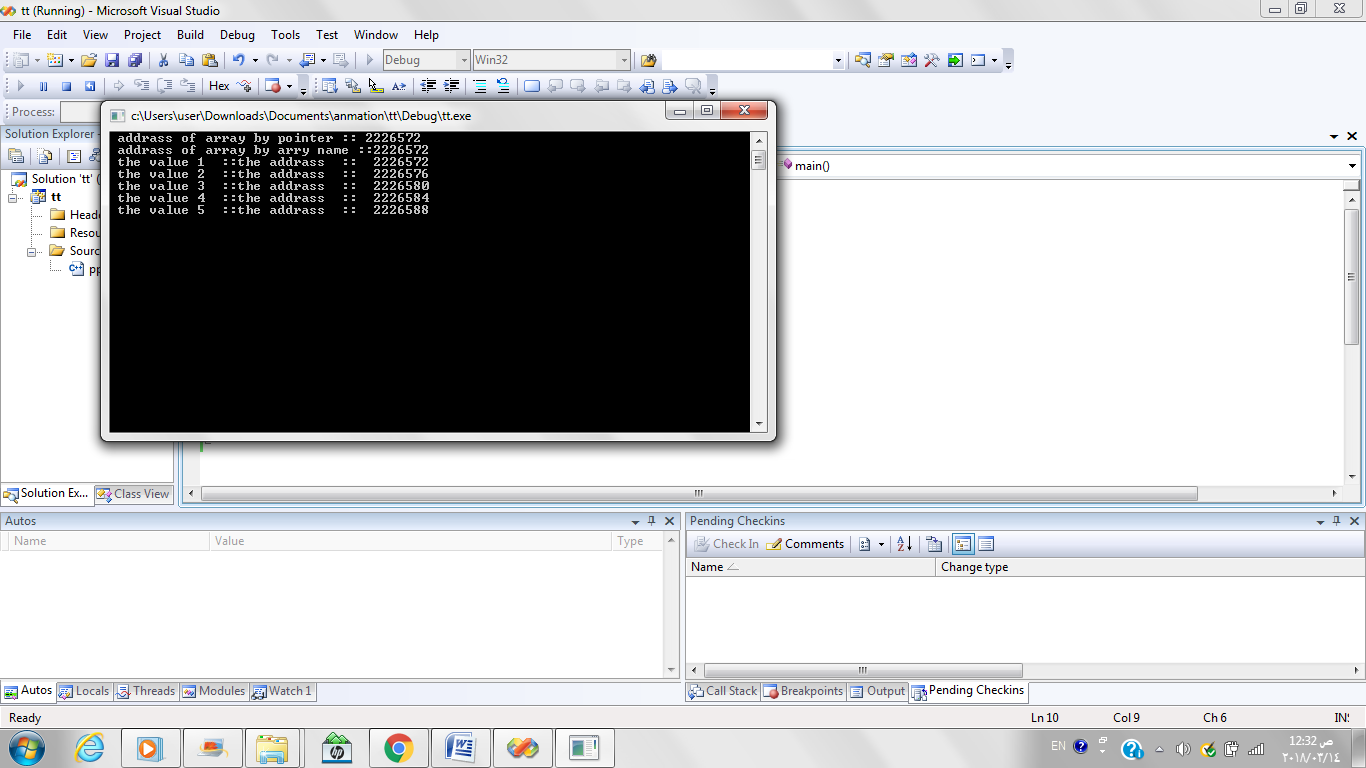
p++;

}

getch();

}

**Output**



In the above program, the pointer \*p will print all the values stored in the array one by one.

|  |
| --- |
| Q1// Output of following program?  #include <stdio.h>  void main()  { int v[] = {10, 100, 200};  int \*ptr;  int i;  ptr = v;  printf("Value of v[0] = %x\n", &v[0]);  for ( i = 0; i < 3; i++)  { printf("Value of \*ptr = %d\n", \*ptr);  printf("Value of ptr = %x\n\n", ptr);  ptr++;  }  getch();  } |

**Output**

Q2// Output of following program::

|  |
| --- |
| #include <stdio.h>  void main()  { int \*ptr;  int x;  ptr = &x;  \*ptr = 0;  printf(" x = %d\n", x);  printf(" \*ptr = %d\n", \*ptr);  \*ptr += 5;  printf(" x = %d\n", x);  printf(" \*ptr = %d\n", \*ptr);  \*ptr=\*ptr +1; // or (\*ptr)++;  printf(" x = %d\n", x);  printf(" \*ptr = %d\n", \*ptr);  getch();  } |

**Output**

Q3// Output of following program::

#include <stdio.h>

void main()

{ int arr[] = {1, 2, 3, 4, 5};

int \*p = arr;

p += 2;

printf("%d", \*p);

getch();

}

**Output**

Q4// Output of following program::

#include <stdio.h>

void main()

{

int var;

int \*ptr = &var;

\*ptr = 5;

printf("var=%d and \*ptr=%d",var,\*ptr);

getch();

}

**Output**