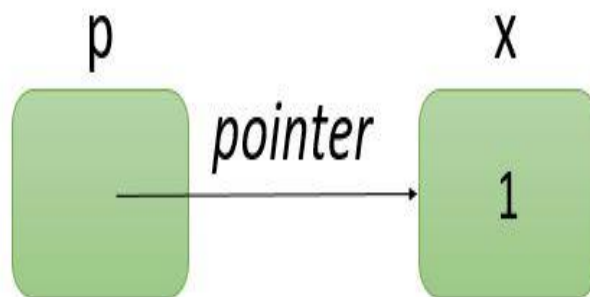


## Pointers



- **Pointers** are memory locations used to save values in C.
- Pointers are used to store information about the location (address) of any other data.

### Pointers Declaration

- To declare an integer pointer, we use the following syntax:

**Int \*P;**

- This declaration sets up a variable named p. It isn't an int - **the asterisk (\*) means that p is a pointer** - and will be used to store information about the location of the data of type int.
- Now we make the P pointer points to the var variable: **ptr= &var;**
- To get the value pointed to by the pointer we have to use the operator (an asterisk: **"\*"**)
- To print the value pointed to by the pointer: **printf("value of var = %d", \*P);**
- To set the value pointed to by the pointer: **\*p=4;**

This statement **will not change the pointer value**; it will change the value pointed to by the pointer instead.

```
#include <stdio.h>

int main() {

// declare variable

int var = 7;

// declare pointer

int *p = &var;

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

*p = 15;

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

return 0;

}
```

### **EXERCISE1:**

The C program below defines a pointer variable and prints out its value.

```
#include <stdio.h>
int main()
{
    int *p, q;
    q = 100;
    p = &q;
    printf("q = %d and p = %d \n", q, *p);
    return 0;
}
```

1. Compile and run the program in your machine, and complete the table below:

q	p
-200	
0	
null	

- Given the value 100 for \*p and the value of q is unknown. Modify the above program to display the same results.

### **EXERCISE2:**

The main reasons for using pointers to arrays are ones of notational convenience and of program efficiency. Pointers to arrays generally result in code that uses less memory and executes faster.

The C program below defines a pointer to an array and prints out the first 3 values.

```
#include <stdio.h>
int main()
{
    int *x;
    int y[ ] = {10,20,30,40,50,60,70,80,90,100};
    x = y;
    printf("%d %d %d \n", *x, *(x+1), *(x+2));
    return 0;
}
```

- Compile and run the program in your machine, and complete the table below:

pointer	value
*x	
*(x*3)	
*(x/2)	
*(x%3)	
*x+1	
*(x+1)	
*x+2	
*x-2	
*(x-2)	
*(x+10)	

- Modify your code to compute sum of the array elements.

2. Modify your code by adding pointer to array in a separate function to compute and return the average of the array elements. The function takes as input the array of 10 integer numbers and returns their average.

**EXERCISE 3:**

1. Write a C program to calculate and display the sum of two numbers x and y. The result is stored in z. Your program should prompt the user to enter the two numbers and then call a function named summation to add the two numbers. The addresses of x, y, and z must be passed as parameters to the function summation.
2. Modify your program to include a function named avg which calculates and returns the average of the two numbers. The addresses of x, y, and z must be passed as parameters to the function avg.
3. Modify your program to include a function named maxm which finds and returns the max of the two numbers. The addresses of x, y, and z must be passed as parameters to the function maxm.

**EXERCISE 4:**

Write a program that reads an array of integers of size n and sort the elements using pointers and array. The program should contain a function with a pointer parameter then sort the array.