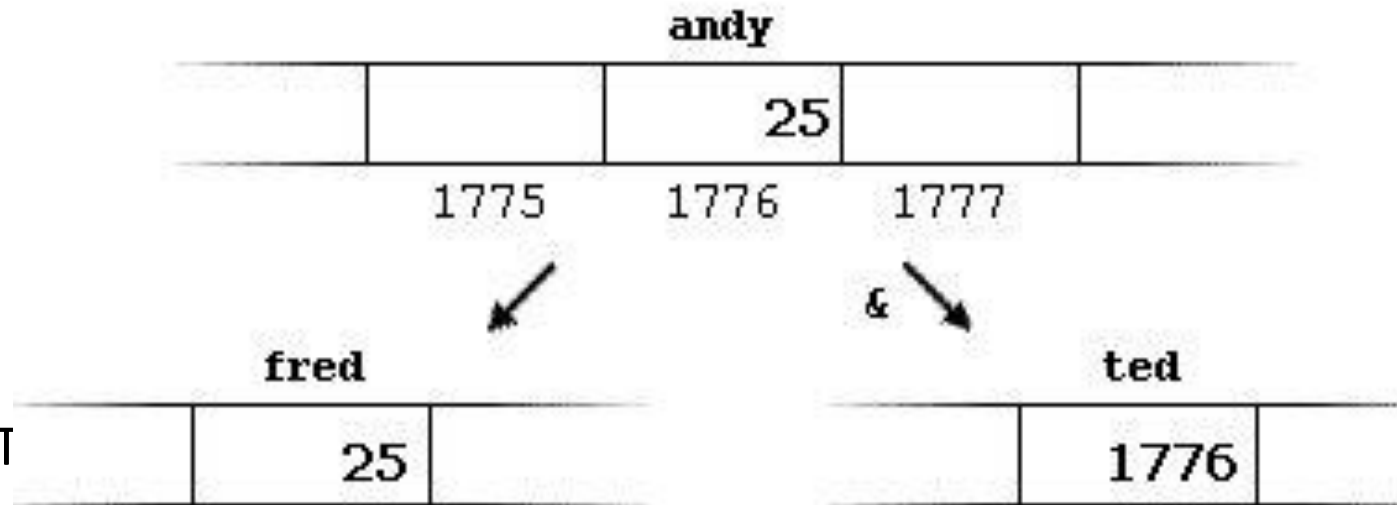


# POINTER (IN C/C++)

# What is a pointer?

Variable in a program is something with a name, the value of which can vary. The way the compiler and linker handles this is that it assigns a specific block of memory within the computer to hold the value of that variable.

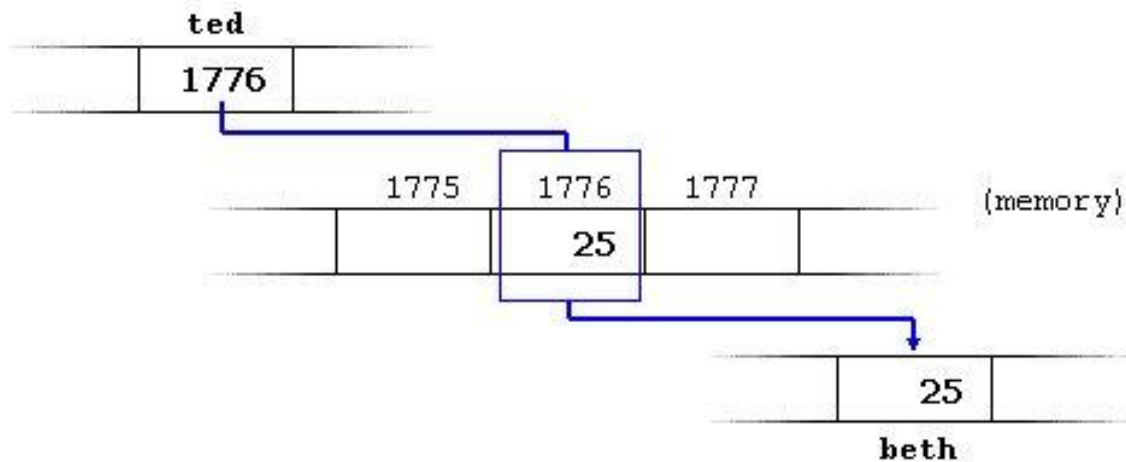


- T
- The right side is the address of that memory

# Dereferencing:

- `int bar = *foo_ptr;`
- `*foo_ptr = 42; // set foo to 42 which is also effect bar = 42`

```
beth = *ted;
```



contain in that is 25 which is what we need.

# Differences between & and \*

& is the reference operator and can be read as  
"address of"

\* is the dereference operator and can be read as  
"value pointed by"

# A variable referenced with & can be dereferenced with \*.

- `Andy = 25;`
- `Ted = &andy;`

All expressions below are true:

- `andy == 25 // true`
- `&andy == 1776 // true`
- `ted == 1776 // true`
- `*ted == 25 // true`

# How to declare pointer?

- Type + “\*” + name of variable.
- Example: `int * number;`
- `char * c;`
- 
- number or c is a variable is called a *pointer variable*



# How to use pointer?

- `int foo;`
- `int *foo_ptr = &foo;`
- *foo\_ptr* is declared as a pointer to int. We have initialized it to point to *foo*.
- *foo* occupies some memory. Its location in memory is called its address. `&foo` is the address of *foo*

# Assignment and pointer:

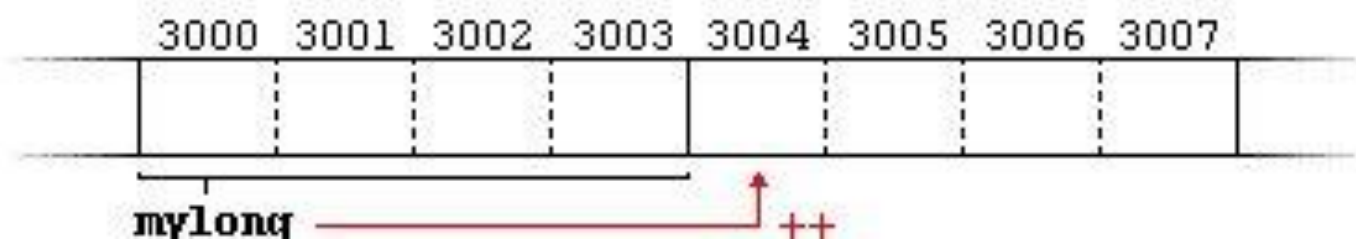
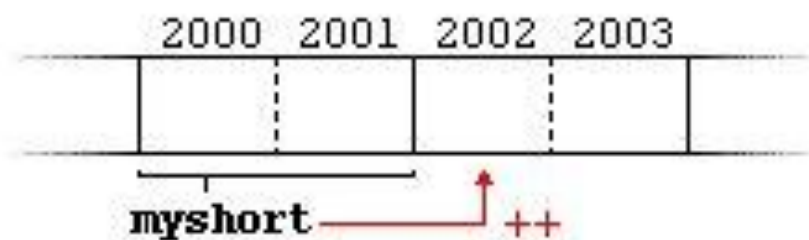
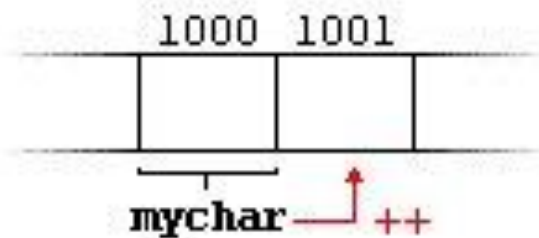
- `int *foo_pr = 5; // wrong`
- `int foo = 5;`
- `int *foo_pr = &foo; // correct way`

# Change the pointer to the next memory block:

- `int foo = 5;`
- `int *foo_pr = &foo;`
- `foo_pr ++;`

# Pointer arithmetics

- `char *mychar; // sizeof 1 byte`
- `short *myshort; // sizeof 2 bytes`
- `long *mylong; // sizeof 4 byts`
  
- `mychar++; // increase by 1 byte`
- `myshort++; // increase by 2 bytes`
- `mylong++; // increase by 4 bytes`



# Increase pointer is different from increase the dereference

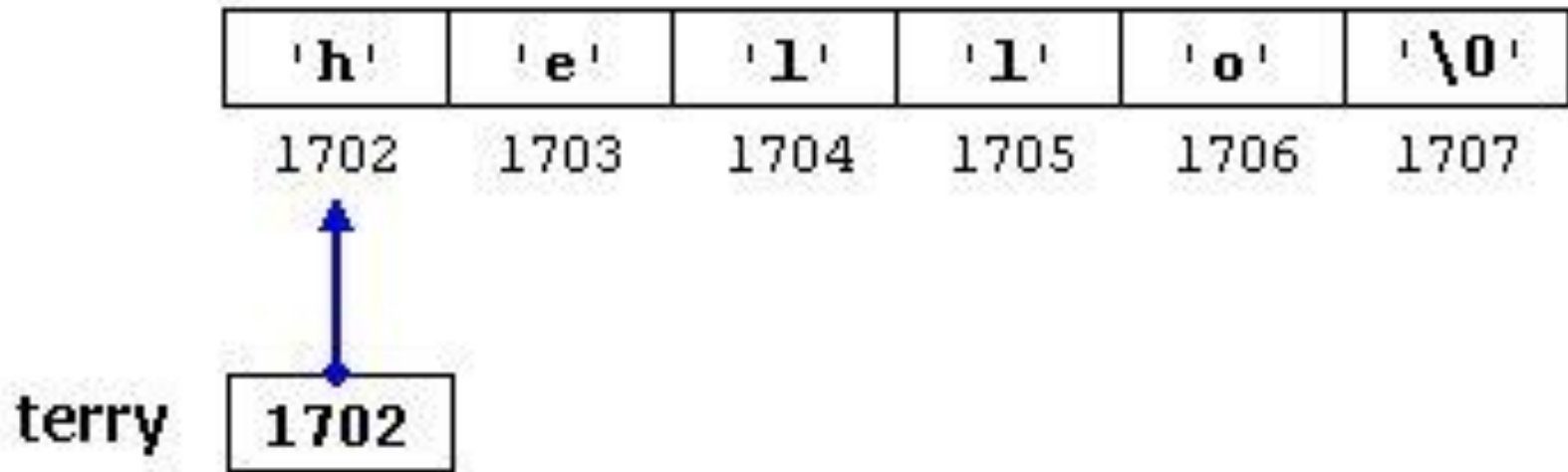
- `*P++;` // unary operation: go to the address of the pointer then increase its address and return a value
- `(*P)++;` // get the value from the address of p then increase the value by 1

# Arrays:

- `int array[] = {45,46,47};`
- we can call the first element in the array by saying: `*array` or `array[0]`.
- Also the second element would be call: `*(array +1)` or `array[1]`

# Array of character pointer

```
char * terry = "hello";
```





# Example:

- `char *p1 = &str1[0], *p2 = &str2[0];`
- `while(1) {`
  - `if(*p1 != *p2)`
    - `return *p1 - *p2;`
  - `if(*p1 == '\0' || *p2 == '\0')`
  - `return 0; p1++; p2++;`
  - `}`