

# Introduction to C Programming

Section 9

# Recall Variables

```
int x1=1;  
int x2=7;
```

| <i>name</i> | <i>address</i> | <i>Memory - content</i> |
|-------------|----------------|-------------------------|
|             | 0              |                         |
|             | 1              |                         |
| x1          | 2              | 1 = 000000001           |
|             | 3              |                         |
|             | 4              |                         |
| x2          | 5              | 7 = 00000111            |
|             | 6              |                         |
|             | ...            |                         |

# Pointers

- When the value of a variable is used, the contents in the memory are used
  - `y=x;` will read the contents in the 4 bytes of memory, and then assign it to variable `y`
- `&x` can get the address of `x` (referencing operator `&`)
- The address can be passed to a function:
  - `scanf("%d", &x);`
- The address can also be stored in a variable .....

# Pointer: Reference to Memory

- Pointer is a variable that
- Contains the address of another variable
- Pointer refers to an address
- Examples

```
int i;  
int *pi;  
i = 20;  
pi = &i;
```

# Pointers

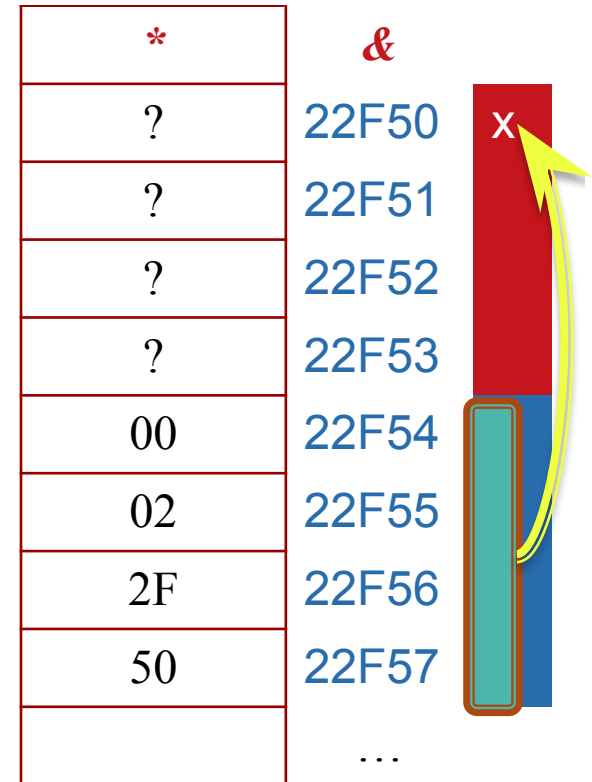
- To declare a pointer variable

**type** \* **PointerName;**



- For example:

```
int    x;  
int    * p;    //p is a int pointer  
// char *p2;  
p = &x;    /* Initializing p */
```



# Pointer: Declaration and Initialization

```
int i, *pi;
```

```
pi = &i;
```

```
float f;
```

```
float *pf = &f;
```

```
char c, *pc = &c;
```

# Addresses and Pointers

```
int a, b;  
int *c, *d;  
a = 5;  
c = &a;  
d = &b;  
*d = 9;  
printf(...,c, *c,&c)  
printf(...,a, b)
```

*name*    *address*    *memory*

|   |   |   |
|---|---|---|
|   | 0 |   |
| a | 1 | 5 |
| b | 2 | 9 |
| c | 3 | 1 |
| d | 4 | 2 |

c=1    \*c=5    &c=3

a=5   b=9

# Addresses and Pointers

- A pointer variable is a variable!
  - It is stored in memory somewhere and has an address.
  - It is a string of bits (just like any other variable).
  - Pointers are 32 bits long on most systems.



# Using Pointers

- You can use pointers to access the values of other variables, *i.e.* the contents of the memory for other variables
- To do this, use the \* operator (**dereferencing operator**)
  - Depending on different context, \* has different meanings