指標

講師 : 陳楷元

大綱

- 如果沒有指標
- 記憶體
- 指標的操作
- 指標與陣列

如果沒有指標

函式中改值

```
void swap(int a, int b)
{
    int temp=a;
    a=b;
    b=temp;
}
int main()
{
    int a=3,b=5;
    swap(a,b);
    printf("%d %d", a, b);
}
```

a=3　b=5

```
void swap(int *ptr1, int *ptr2)
{
    int temp=*ptr1;
    *ptr1=*ptr2;
    *ptr2=temp;
}
int main()
{
    int a=3,b=5;
    swap(&a, &b);
    printf("%d %d", a, b);
}
```

a=5　b=3

複製陣列

```c
#include <stdio.h>
int main()
{
    char a[6]="Hello";
    char b[6];
    b=a;
    return 0;
}
```

Error!

```c
#include <stdio.h>
int main()
{
    char a[6]="Hello";
    char *b=a;
    printf("b: %s", b);
    return 0;
}
```

b: Hello

資訊之芽
SPROUT

記憶體

參考自2021資芽指標ppt

資訊之芽
SPROUT

教室裡有10種人

- 1種是會二進位的人
- 另1種是不會二進位的人

解讀2進位與16進位

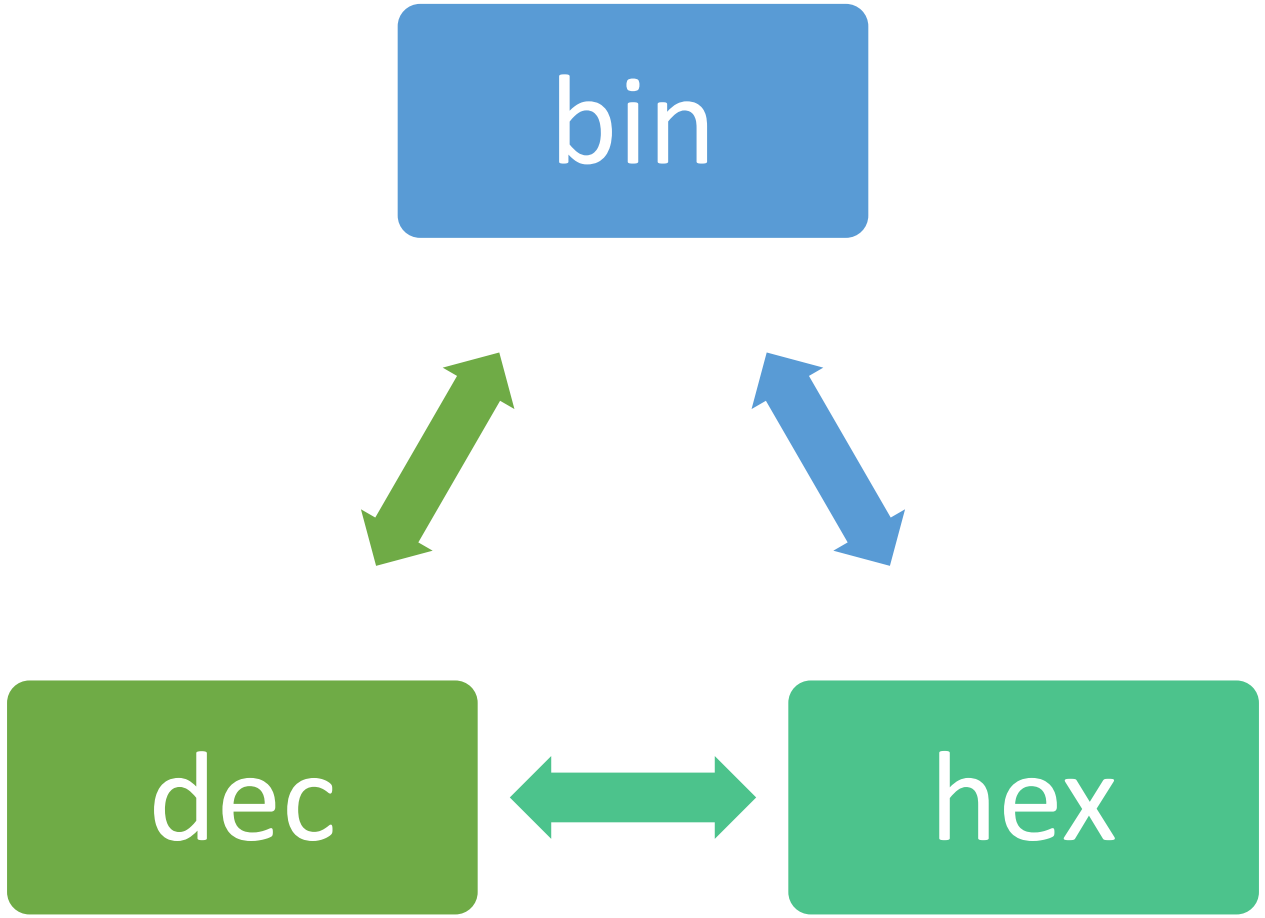
- Decimal (十進位) : $1234 = 1 * 10^3 + 2 * 10^2 + 3 * 10^1 + 4 * 10^0$
- Binary (二進位) : $1011_2 = 1 * 2^3 + 0 * 2^2 + 1 * 2^1 + 1 * 2^0$
- Hexadecimal (十六進位) :

$$0x19AF = 1 * 16^3 + 9 * 16^2 + 10 * 16^1 + 15 * 16^0$$

A=10	D=13
B=11	E=14
C=12	F=15

資訊之芽
SPROUT

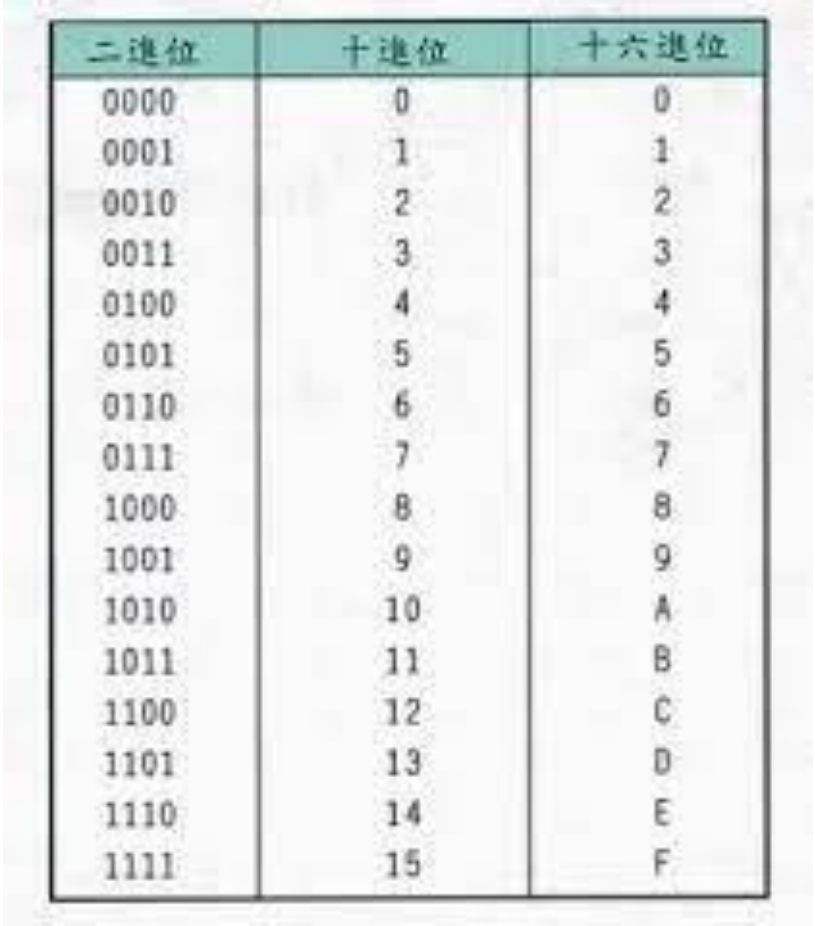
不同數字系統間的轉換



資訊之芽
SPROUT

不同數字系統間的轉換

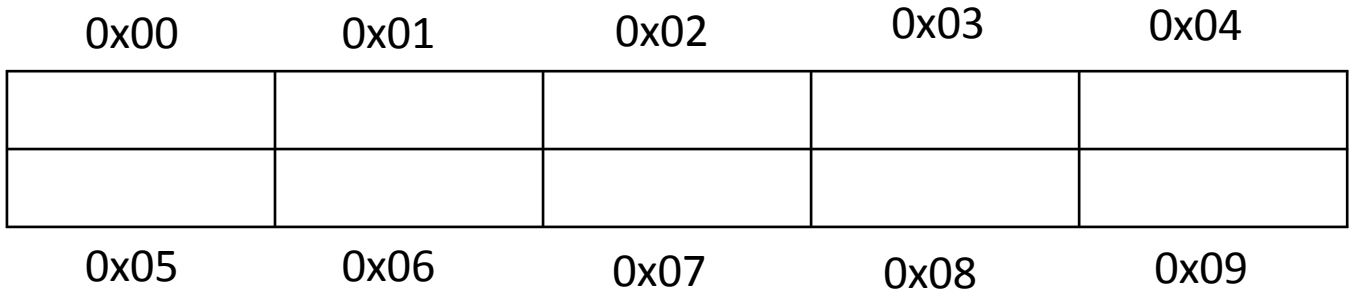
- 2轉10: 同解讀
- 10轉2: 用2的冪次湊
- 2轉16: 四個一組見對照表
- 16轉2: 見對照表
- 10轉16: 10轉2, 2再轉16
- 16轉10: 同解讀



二進位	十進位	十六進位
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

記憶體就像是陣列

陣列用index取值,
記憶體用16進位取值



資訊之芽
SPROUT

變數宣告與記憶體

int a=5;

Step 1. 電腦找到可以用的格子

Step 2. 標記變數名稱

Step 3. 將值存入

0xff a 5				

變數操作與記憶體

a++;

Step 1. 電腦利用變數名稱找到對應記憶體

Step 2. 根據地址對該變數進行操作

變數名稱給人方便,
但電腦是根據記憶體做事

0xff a 5+1				

指標的操作

指標是什麼？

- 指標是變數的一種
- 變數都有類別、名字、位址、值四大基本要素

指標變數

- 類別 : 指向變數的類別
- 名字
- 值 : 指向變數的位址
- 位址 : 無論指向何種變數, 皆占8bytes

不同變數值可能相同,
但位址必定不同

整理

- sizeof (int) = 4 bytes
- sizeof (float) = 8 bytes
- sizeof (double) = 8 bytes
- sizeof (char) = 1 byte
- sizeof(pointer) = 8 bytes

NULL vs nullptr

- C 只有 NULL，意思是空指標
- C ++ 的 NULL，意思是所有型別中的0
- C ++ 11 有nullptr, 意思是空指標

- 結論:當你意指空指標, 建議寫C用NULL, 寫C++用nullptr

宣告&初始化

* :(宣告時) 無意義, 表後面是指標
  (其他狀況) 取值

&: 取位址

宣告&初始化

```
int i;
i=5;
int *iptr;
*iptr=i;
```

```
int i;
i=5;
int *iptr;
iptr=&i;
```

```
int i=5;
int *iptr=&i;
```

Ex1

```c
int main()
{
    int i,k;
    int *iptr1, *iptr2;
    scanf("%d", &i);
    iptr1=&i;
    iptr2=iptr1;
    printf("&i=%p\n", &i);
    printf("iptr1=%p\n", iptr1);
    printf("&iptr1=%p\n", &iptr1);
    printf("iptr2=%p\n", iptr2);
    printf("&iptr2=%p\n", &iptr2);
    return 0;
}
```

```
5
&i=000000000062FE1C
iptr1=000000000062FE1C
&iptr1=000000000062FE10
iptr2=000000000062FE1C
&iptr2=000000000062FE08
```

	取值	的值	取位址
	*		&
i	無意義	5	a
iptr1	5	a	b
iptr2	5	a	c

資訊之芽
SPROUT

Ex2

```c
int main()
{
    int i,k;
    int *iptr1, **iptr2;
    scanf("%d", &i);
    iptr1=&i;
    iptr2=&iptr1;
    printf("&i=%p\n", &i);
    printf("iptr1=%p\n", iptr1);
    printf("&iptr1=%p\n", &iptr1);
    printf("iptr2=%p\n", iptr2);
    printf("&iptr2=%p\n", &iptr2);
    return 0;
}
```

```
5
&i=000000000062FE1C
iptr1=000000000062FE1C
&iptr1=000000000062FE10
iptr2=000000000062FE10
&iptr2=000000000062FE08
```

	取值	的值	取位址
	*		&
i	無意義	5	a
iptr1	5	a	b
iptr2	a	b	c

資訊之芽
SPROUT

思考1

```
int main()
{
    int a=3,b=5;
    int *ptr1=&a;
    int *ptr2=&b;
    int temp=*ptr1;
    *ptr1=*ptr2;
    *ptr2=temp;
    printf("%d %d\n", a, b);
    printf("%d %d\n", *ptr1, *ptr2);
}
```

a=5

b=3

*ptr1=5

*ptr2=3

思考2

```
int main()
{
    int a=3,b=5;
    int *ptr1=&a;
    int *ptr2=&b;
    int *temp=ptr1;
    ptr1=ptr2;
    ptr2=temp;
    printf("%d %d\n", a, b);
    printf("%d %d\n", *ptr1, *ptr2);
}
```

a=3

b=5

*ptr1=5

*ptr2=3

指標與陣列

指標陣列 vs 用指標存陣列

指標陣列

欲將v陣列元素歸零:

```
int v[3]={1,2,3};
int *p[3]={&v[0],&v[1],&v[2]};
for(int i=0;i<3;i++)
    *p[i]=0;
```

用指標存陣列

陣列名稱的值就是
陣列的起始位址,
即a=&(a[0])

用指標存陣列

```
int a[5]={1,2,3,4,5};
int *ptr;
ptr=&(a[0]);
```

```
int a[5]={1,2,3,4,5};
int *ptr;
*ptr=a[0];
```

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

```
int a[5]={1,2,3,4,5};
int *ptr=&(a[0]);
```

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

資訊之芽
SPROUT

用指標存陣列

欲將v陣列元素歸零：

```
int v[3]={1,2,3};
int *p=v;
for(int i=0;i<3;i++)
    p[i]=0;
```

指標存陣列，
指標可當成陣列用

用指標存陣列

欲將v陣列元素歸零:

```
int v[3]={1,2,3};
int *p=v;
for(int i=0;i<3;i++)
{
    *p=0;
    p++;
}
```

p++指的是讓p指向陣列中下一個元素，這裡是指加上sizeof(int)

指標當成陣列用，可以跑，更好用

課堂練習

https://neoj.sprout.tw/problem/8857/

找出給定陣列的最大值。但是你現在只有:
這個陣列有幾個元素
一個指向陣列頭的指標
一個指向陣列結尾的指標

資訊之芽
SPROUT