

105學年 資料結構TA課

指標(Pointer)與位址(Address)

Outline

- 位址?
- 指標?
- 指標的指標
- 指標使用
- 動態陣列的宣告

位址？

位址？

- 什麼是位址？
簡單的來說就是變數放在電腦記憶體中的哪個位置。
- 『&』運算子：
&是取址運算子(Address-of operator)可用來取出該變數的位址。
- DEMO：

```
int A = 20;  
cout << "A的地址:" << &A << endl;
```

A的地址:0x7fff5890ab88

位址

- 位址大小：
每個變數所需的位址大小皆不同。

int : 4位元

```
int A[5] = {1, 2, 3, 4, 5};  
for (int i = 0; i < 5; i++)  
    cout << "A[" << i << "]的Address:" << &A[i] << endl;
```

```
A[0]的 Address:0x7fff50581b80  
A[1]的 Address:0x7fff50581b84  
A[2]的 Address:0x7fff50581b88  
A[3]的 Address:0x7fff50581b8c  
A[4]的 Address:0x7fff50581b90
```

char : 1位元

```
char C[5] = {'1', '2', '3', '4', '5'};  
for (int i = 0; i < 5; i++)  
    cout << "C[" << i << "]的Address:" << (void *)&C[i] << endl;
```

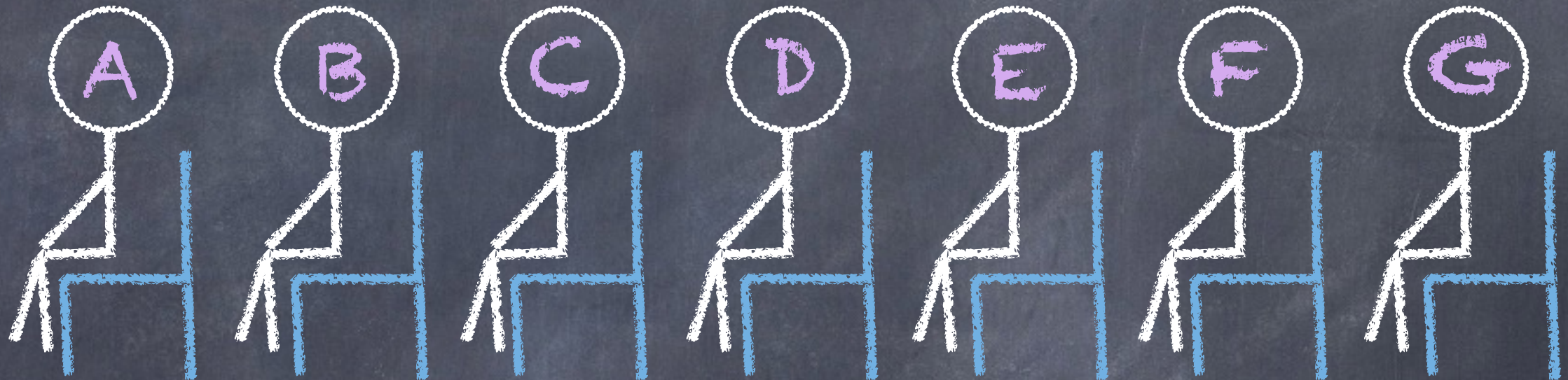
```
C[0]的 Address:0x7fff5ef3db87  
C[1]的 Address:0x7fff5ef3db88  
C[2]的 Address:0x7fff5ef3db89  
C[3]的 Address:0x7fff5ef3db8a  
C[4]的 Address:0x7fff5ef3db8b
```


你在說什麼？



位址

第一個位子
他的名字叫A



位子
名稱



0x01

0x02

0x03

0x04

0x05

0x06

0x07

- 坐在0x06的人叫什麼名字?
- G坐的位在?
- 坐在第四個位子的人叫什麼名字?

位址

```
char X[7];
```

```
X[0] = A; //0x01
```

```
X[1] = B; //0x02
```

```
X[2] = C; //0x03
```

```
X[3] = D; //0x04
```

```
X[4] = E; //0x05
```

```
X[5] = F; //0x06
```

```
X[6] = G; //0x07
```

坐在0x06的人叫什麼名字?

G坐的位子?

坐在第四個位子的人叫什麼名字?

翻↓譯

在位址0x06的值為多少?

G的位址為?

在陣列中第四個的值為多少?

指標？

指標?

- 什麼是指標？
用來表示或儲存一個記憶體位址，這個位址的值直接指向存在該位址的物件的值。
- 『*』運算子：
*是取值運算子(dereference operator)可用來取出該指標指向位址的值。
- 定義一個指標：

```
int *a;    // 定義整數指標  
char *b;   // 定義字元指標  
double *c; // 定義浮點數指標
```

```
int A = 10;  
int *ptr = &A;
```

變數
值
位址

A
10
0x84

ptr
0x84
0x80

*ptr
10
0x84

指標的指標

指標的指標

- 當指標指向另一個指標.....

```
int a = 10;
int *ptr_to_int = &a;
int **ptr_to_ptr = &ptr_to_int; ←
cout << "a" << endl;
cout << a << " " << &a << endl;
cout << "ptr_to_int" << endl;
cout << ptr_to_int << " " << &ptr_to_int << endl;
cout << *ptr_to_int << " " << &*ptr_to_int << endl;
cout << "ptr_to_ptr" << endl;
cout << ptr_to_ptr << " " << &ptr_to_ptr << endl;
cout << *ptr_to_ptr << " " << &*ptr_to_ptr << endl;
cout << **ptr_to_ptr << " " << &**ptr_to_ptr << endl;
```

分別印出**值**和**位址**

↓ 執行結果 ↓

```
a
10 0x7fff59527b08
ptr_to_int
0x7fff59527b08 0x7fff59527b00
10 0x7fff59527b08
ptr_to_ptr
0x7fff59527b00 0x7fff59527af8
0x7fff59527b08 0x7fff59527b00
10 0x7fff59527b08
```


指標地獄

$\text{ptr_to_A} = ?$

$*\text{ptr_to_A}$



$*\text{ptr_to_A} = ?$



A

$\&\text{ptr_to_A} = ?$

$\&\text{ptr_to_ptr} = ?$

$**\text{ptr_to_ptr}$

$\&* \text{ptr_to_ptr} = ?$



指標的使用

指標的使用

● 指標操作：

1. 將指標指向一個變數

```
int a = 10;  
int *ptr_to_int = &a;
```

```
int a = 20;  
int *ptr_to_int;  
ptr_to_int = &a;
```

2. 對指標指向的值進行更改

```
int a = 10;  
int *ptr_to_int = &a;  
*ptr_to_int += 1;  
cout << a << endl;
```

← 對指標指向的值+1

指標的使用

● 指標操作：

3. 將指標指向一個陣列

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

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

4. 用指標印出陣列所有的值（方法不只一種）

```
int a[5] = {1, 2, 3, 4, 5};  
  
for (int *ptr = a; ptr < a+5; ptr++)  
{  
    cout << *ptr << endl;  
}
```

1
2
3
4
5

動態陣列的宣告

動態陣列的宣告

- 一維陣列動態宣告：

```
int s = 5;  
int *ptr = new int[s];
```

- 二維陣列動態宣告：

```
int m = 3, n = 2;  
int **ptr = new int*[m];  
for (int i = 0; i < m; ++i)  
    ptr[i] = new int[n];
```


動態陣列的使用

• DEMO :

```
int s;  
cin >> s; // 設定陣列大小  
int *ptr = new int[s];  
for(int i = 0; i < s; i++)  
    cin >> ptr[i];  
// 用ptr[i]的方式印出  
for(int i = 0; i < s; i++)  
    cout << ptr[i] << " ";  
cout << endl;  
// 用*ptr的方式印出  
for(int i = 0; i < s; i++)  
    cout << *(ptr+i) << " ";  
cout << endl;
```

```
6  
1 2 9 10 20 77  
1 2 9 10 20 77  
1 2 9 10 20 77
```


「動手自己玩玩看，玩過比較有感覺。」



TenTail