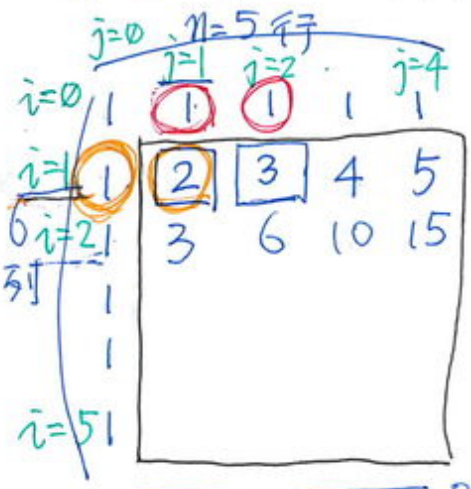
Dynamic Memory Allocation

m=6 n=5 Matrix: 6×5 (m×n)



$$f(i,j) = f(i-1,j) + f(i,j-1)$$

上方 左側

$$f(1,2) = f(0,2) + f(1,1) = 1 + 2 = 3$$

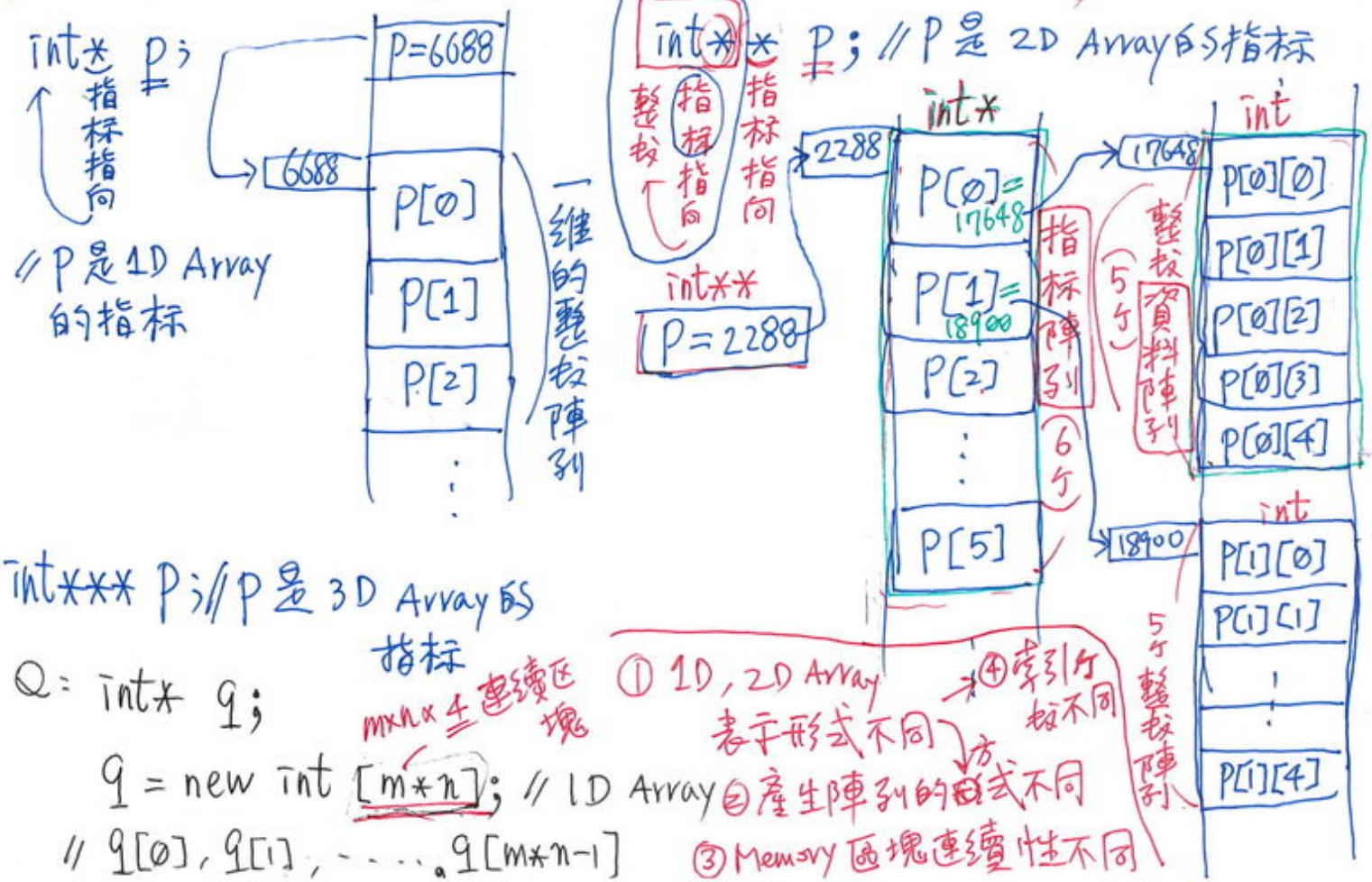
初始：第一列 & 第一行皆為1

---

* 2維陣列的表示形式
* 2維陣列如何產生 --- new/delete
* 2維陣列填入數值 --- 遞回公式
* 2維陣列記憶体配置出錯 ... Memory不足 ... 缺少足夠的連續區段
  ⇒ C++ Exception Handling 意外；例外 處理

2014-May-27
資工一甲

m×n ⇔ 6×5 (列 行)



```
int** p = NULL;
```

建立2D陣列

```
p = new int* [m];   //產生指標陣列
for (int i=0; i<m; i++) {
    p[i] = new int [n];   //產生資料陣列
}
```

刪除2D陣列

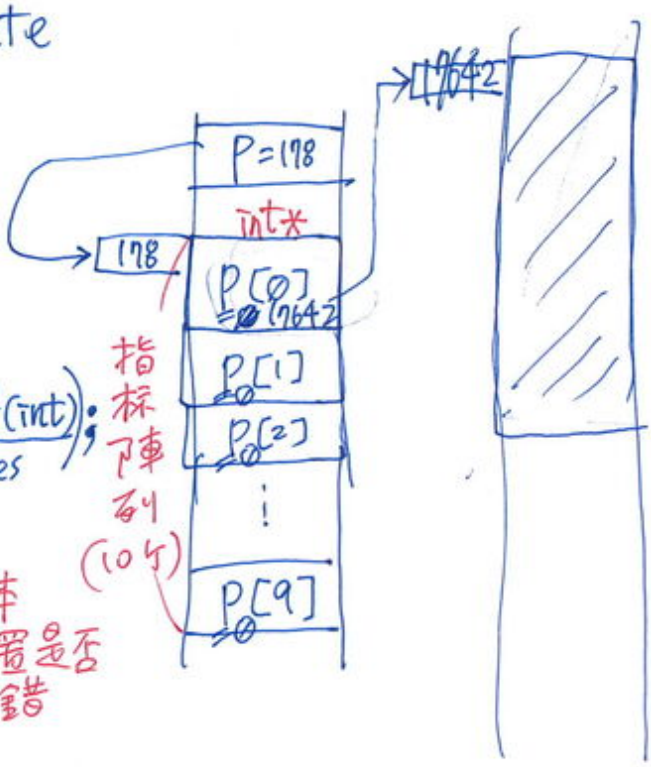
```
for (int i=0; i<m; i++) {
    delete [] p[i];   //刪除每個資料陣列
}
delete [] p;   //刪除指標陣列
p=NULL;
```

Exception Handling 意外處理

C: malloc / free, C++: new / delete

```
int* p[10] = {NULL};
```

int*：指標　p[10]：10個指標陣列




```
for (int i=0; i<10; i++) {
    P[i] = (int*) malloc (10240000 * sizeof(int));
    if (P[i] == NULL)   //檢查記憶體配置是否出錯
        cout << "Memory Error\n";
}                       //錯誤處理
```

int*：轉型　malloc：傳回17642 addr.　(void*)　sizeof(int)：4 bytes

```
for (int i=0; i<10; i++) {
    if (P[i] != NULL) {   // P[i]有成功配置 memory
        free (P[i]);
        P[i]=NULL;
    }
}
```

C++: new / delete

Unhandle Exception
未處理 意外
丟出 訊息

```
int* p[10]={NULL};
for (int i=0; i<10; i++) {
    p[i] = new int [1024000000];
    if (p[i]==NULL)
        cout << i << ": mem error\n";
}
```

改寫 → `p[i] = new(nothrow) int [1024000000]`

nothrow
不丟意外訊息

```
for (int i=0; i<10; i++) {
    delete [] p[i];
    p[i] = NULL;
}
```

刪除 陣列 依p[i]的Addr.

改寫：攔截意外訊息

```
for (int i=0; i<10; i++) {
    p[i] = new int [1024000000];
    try { // 試
        p[i]=new int [1024000000];
    } catch (...) {
        // 意外錯誤處理
        cout << i << ": Memory Error\n";
    }
}
```

catch 攔截 (...) 意外訊息 → 各式各樣的意外