

1. 2차원 포인터

2. 파일에 읽고 쓰기

1. 2차원 포인터

2. 파일에 읽고 쓰기

# 2차원 배열의 메모리 배치

## ▶ 2차원 배열

### ▶ array[3][2]

#### ▶ 4 x 4 배열을 그림으로 그린 예

[0][0][0][0]

[1][1][1][1]

[2][2][2][2]

[3][3][3][3]

#### ▶ 실제 메모리에서 배치

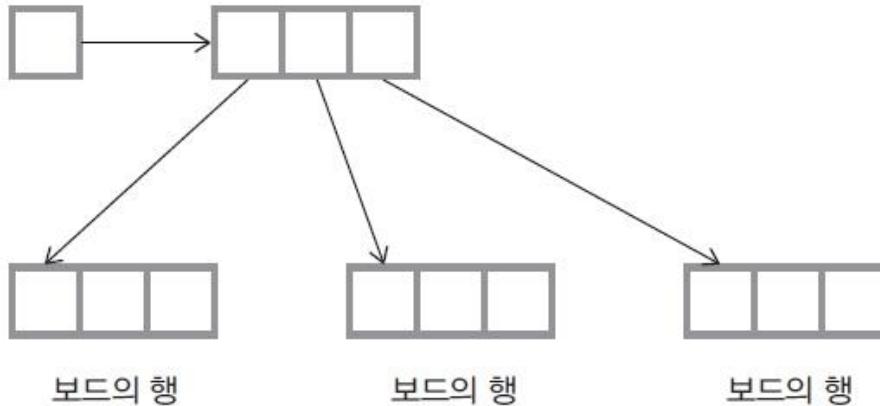
[0][0][0][0][1][1][1][1][2][2][2][2][3][3][3][3]

#### ▶ 포인터 연산으로 전환

▶  $*(array + 3 * \text{<배열의 너비>} + 2)$

# 포인터의 포인터를 사용한 2차원 배열

p\_p\_tictactoe    각 행을 가리키는 포인터



```
int **p_p_tictactoe;  
// 포인터 배열을 할당하고 있기 때문에 int*를 사용했다.  
p_p_tictactoe = new int*[ 3 ];  
  
// 각 포인터에는 정수 배열의 주소가 저장된다.  
for ( int i = 0; i < 3; i++ )  
{  
    p_p_tictactoe[ i ] = new int[ 3 ];  
}
```

# 포인터의 포인터를 사용한 2차원 배열

```
for ( int i = 0; i < 3; i++ )  
{  
    for ( int j = 0; j < 3; j++ )  
    {  
        p_p_tictactoe[ i ][ j ] = 0;  
    }  
}
```

2차원 배열 처럼 사용 가능

```
for ( int i = 0; i < 3; i++ )  
{  
    delete [] p_p_tictactoe[ i ];  
}
```

New와 반대 순서로 delete

```
delete [] p_p_tictactoe;
```

1. 2차원 포인터

2. 파일에 읽고 쓰기

- ▶ 파일을 읽고 쓰는 건 cout과 cin을 사용하는 것과 유사
- ▶ cin/cout : global variable. 따로 선언하지 않고 사용 가능
- ▶ 파일 입출력을 위한 데이터 타입: 우리가 선언해야 함
  - ▶ 파일 읽기 데이터 타입
  - ▶ **ifstream**: input file stream을 줄인 표현
  - ▶ **ofstream**: output file stream을 줄인 표현
    - ▶ 스트림(stream) – 한 묶음의 읽고 쓰는 데이터를 의미함

# 파일 읽기 – ifstream 타입

```
#include <fstream>
#include <iostream>

using namespace std;
int main ()
{
    ifstream file_reader( "myfile.txt" );
    if ( ! file_reader.is_open() )
    {
        cout << "Could not open file!" << '\n';
    }
    int number;
    // 여기서 정수를 제대로 읽어 들이는지 아닌지를 확인한다.
    if ( file_reader >> number )
    {
        cout << "The value is: " << number;
    }
}
```

현재 디렉토리에서 myfile.txt 찾는  
전체 경로를 지정해도 됨(c:\W\myfile.txt)

cin 을 쓰듯 사용



# 파일 읽기 – ifstream 타입

---

```
#include <fstream>
#include <iostream>
#include <vector>

using namespace std;

int main ()
{
    ifstream file_reader( "myfile.txt" );
    if ( ! file_reader.is_open() )
    {
        cout << "Could not open file!" << '\n';
    }
    while(1){
        int number;
        if ( ! ( file_reader >> number ) ) 파일의 끝(EOF)를 의미
        {
            break;
        }

        cout<<"The value is : " << number <<endl;
    }
}
```

# 파일 쓰기 – ofstream 타입

```
#include <fstream>
#include <iostream>
#include <cstdlib>

using namespace std;

int main ()
{
    ofstream file_writer( "highscores.txt" );
    if ( ! file_writer.is_open() )
    {
        cout << "Could not open file!" << '\n';
        return 0;
    }

    // 실제 점수는 아직 없으므로 일단 10에서 1까지 숫자를 출력한다.
    for ( int i = 0; i < 10; i++ )
    {
        file_writer << 10 - i << '\n';
    }
}
```

## TEAM EX1 파일 복사하기

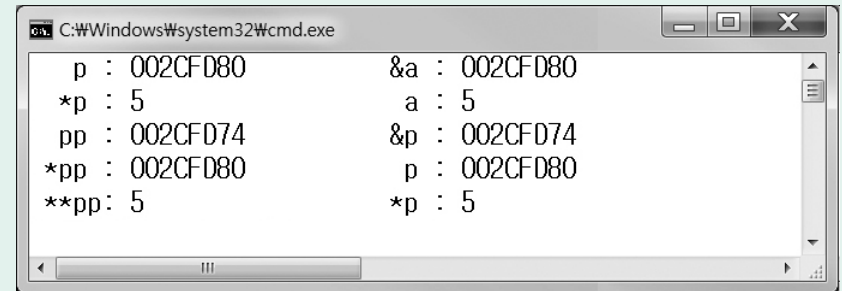
```
01 #include <iostream>
02 #include <fstream>
03 using namespace std;
04
05 int main(int argc, char *argv[])
06 {
07     ifstream fin;
08     ofstream fout;
09
10     char ch;
11
12     if(argc != 3){
13         cout<<"명령어 입력한 후 원본 파일명과 복사본 파일명을 입력하
14         세요"<<endl;
15     }
16
17     fin.open(argv[1]);
18     fout.open(argv[2]);
19
20     if(fin.fail() || fout.fail()){
21         cout<<"파일 오픈을 실패했습니다."<<endl;
22     }
23     return 0;
```

```
23 }
24
25 while(fin.get(ch))
26     fout.put(ch);
27
28 fin.close();
29 fout.close();
30 }
```

```
>> mycopy.exe file1 file2
```

## TEAM EX2 2차원 포인터 이해하기

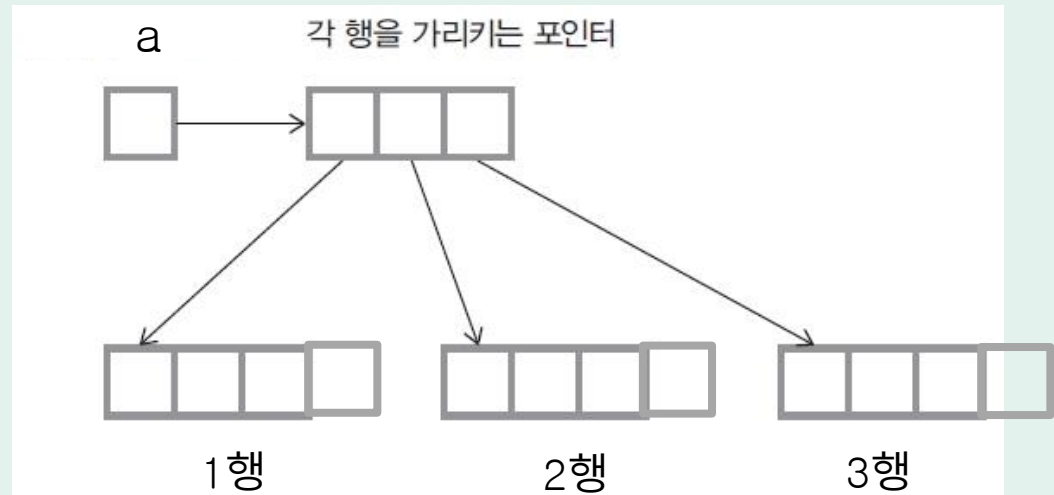
```
01 #include <iostream>
02 using namespace std;
03 void main()
04 {
05     int a=5;
06     int *p;
07     int **pp;
08
09     p=&a;
10     pp=&p;
11
12     cout<<" p : "<< p <<" \t &a : "<< &a << endl;
13     cout<<" *p : "<< *p <<" \t \t a : "<< a << endl;
14     cout<<" pp : "<< pp <<" \t &p : "<< &p << endl;
15     cout<<" *pp : "<< *pp <<" \t p : "<< p << endl;
16     cout<<" **pp: "<< **pp <<" \t \t *p : "<< *p << endl;
17 }
```



```
C:\Windows\system32\cmd.exe
p : 002CFD80      &a : 002CFD80
*p : 5           a : 5
pp : 002CFD74     &p : 002CFD74
*pp : 002CFD80    p : 002CFD80
**pp: 5          *p : 5
```

## TEAM EX3 배열의 원소를 포인터 연산자를 이용해서 출력하기

```
01 #include <iostream>
02 using namespace std;
03 #define ROW 3
04 #define COL 4
05 void main()
06 {
07     int a[ROW][COL] = { {90, 85, 95, 100},
08 {75, 95, 80, 90},
09 {90, 80, 70, 60}
10 };
11     int r, c;
12     for(r=0; r<ROW; r++){
13         for(c=0; c<COL; c++) {
14             cout<<"*(*(a+"<<r<<" "+"<<c<<"")):"<<"*(*(a+r)+c)<<" Wt";
15         }
16         cout<<"\n";
17     }
18 }
```



```
C:\Windows\system32\cmd.exe
*(*(a+0)+0):90      *(*(a+0)+1):85      *(*(a+0)+2):95      *(*(a+0)+3):100
*(*(a+1)+0):75      *(*(a+1)+1):95      *(*(a+1)+2):80      *(*(a+1)+3):90
*(*(a+2)+0):90      *(*(a+2)+1):80      *(*(a+2)+2):70      *(*(a+2)+3):60
```