Part 1,

5주차 과제 실습(2차원 배열)

Project1 (전역 범위)

```c
#include <stdio.h>

int main() {

    int     i, iData[2][3] = { {0,1,2}, {3,4,5} };
    float   fData[2][3] = { {6,7,8}, {9,10,11} };
    char    cData[2][3] = { {'a','b','c'}, {'d','e','f'} };
    double  dData[2][3] = { {12,13,14}, {15,16,17} };

    int* iptr = &iData[0][0];
    for (i = 0; i < 6; i++) {
        printf("¥n address: %u  iData[%d]= %d", iptr, i, *iptr);
        iptr++;
    }
    float* fptr = &fData[0][0];
    for (i = 0; i < 6; i++) {
        printf("¥n address: %u  fData[%d]= %.3f", fptr, i, *fptr);
        fptr++;
    }
    char* cptr = &cData[0][0];
    for (i = 0; i < 6; i++) {
        printf("¥n address: %u  cData[%d]= %c", cptr, i, *cptr);
        cptr++;
    }
    double* dptr = &dData[0][0];
    for (i = 0; i <6; i++) {
        printf("¥n address: %u  dData[%d]= %.3f", dptr, i, *dptr);
        dptr++;
    }

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

```
address: 14350368  iData[0]= 0
address: 14350372  iData[1]= 1
address: 14350376  iData[2]= 2
address: 14350380  iData[3]= 3
address: 14350384  iData[4]= 4
address: 14350388  iData[5]= 5
address: 14350336  fData[0]= 6.000
address: 14350340  fData[1]= 7.000
address: 14350344  fData[2]= 8.000
address: 14350348  fData[3]= 9.000
address: 14350352  fData[4]= 10.000
address: 14350356  fData[5]= 11.000
address: 14350320  cData[0]= a
address: 14350321  cData[1]= b
address: 14350322  cData[2]= c
address: 14350323  cData[3]= d
address: 14350324  cData[4]= e
address: 14350325  cData[5]= f
address: 14350264  dData[0]= 12.000
address: 14350272  dData[1]= 13.000
address: 14350280  dData[2]= 14.000
address: 14350288  dData[3]= 15.000
address: 14350296  dData[4]= 16.000
address: 14350304  dData[5]= 17.000
E:₩School_Data₩1. 자료구조₩파제₩5주차₩Project1₩Debug₩Project1.ex
디버깅이 중지될 때 콘솔을 자동으로 닫으려면 [도구] -> [옵션] -> [
하도록 설정합니다.
이 창을 닫으려면 아무 키나 누르세요...
```

Part 1,

5주차 과제 실습(3차원 배열)

Project2 (전역 범위)

```c
#include <stdio.h>

int main() {

    int     i, iData[2][2][3] = { {{0,1,2}, {3,4,5} }, { {6,7,8}, {9,10,11}} };
    float     fData[2][2][3] = { {{12,13,14}, {15,16,17} }, { {18,19,20}, {21,22,23}} };
    char     cData[2][2][3] = { {{'a','b','c'}, {'d','e','f'} }, { {'g','h','i'}, {'j','k','l'}} };
    double    dData[2][2][3] = { {{24,25,26}, {27,28,29} }, { {30,31,32}, {33,34,35}} };

    int* iptr = &iData[0][0][0];
    for (i = 0; i < 12; i++) {
        printf("¥n address: %u  iData[%d]= %d", iptr, i, *iptr);
        iptr++;
    }
    float* fptr = &fData[0][0][0];
    for (i = 0; i < 12; i++) {
        printf("¥n address: %u  fData[%d]= %.2f", fptr, i, *fptr);
        fptr++;
    }
    char* cptr = &cData[0][0][0];
    for (i = 0; i < 12; i++) {
        printf("¥n address: %u  cData[%d]= %c", cptr, i, *cptr);
        cptr++;
    }
    double* dptr = &dData[0][0][0];
    for (i = 0; i < 12; i++) {
        printf("¥n address: %u  dData[%d]= %.2f", dptr, i, *dptr);
        dptr++;
    }

    return 0;
}
```

Microsoft Visual Studio 디버그 콘솔

```
address:  11533752   iData[0]= 0
address:  11533756   iData[1]= 1
address:  11533760   iData[2]= 2
address:  11533764   iData[3]= 3
address:  11533768   iData[4]= 4
address:  11533772   iData[5]= 5
address:  11533776   iData[6]= 6
address:  11533780   iData[7]= 7
address:  11533784   iData[8]= 8
address:  11533788   iData[9]= 9
address:  11533792   iData[10]= 10
address:  11533796   iData[11]= 11
address:  11533696   fData[0]= 12.00
address:  11533700   fData[1]= 13.00
address:  11533704   fData[2]= 14.00
address:  11533708   fData[3]= 15.00
address:  11533712   fData[4]= 16.00
address:  11533716   fData[5]= 17.00
address:  11533720   fData[6]= 18.00
address:  11533724   fData[7]= 19.00
address:  11533728   fData[8]= 20.00
address:  11533732   fData[9]= 21.00
address:  11533736   fData[10]= 22.00
address:  11533740   fData[11]= 23.00
address:  11533676   cData[0]= a
address:  11533677   cData[1]= b
address:  11533678   cData[2]= c
address:  11533679   cData[3]= d
address:  11533680   cData[4]= e
address:  11533681   cData[5]= f
address:  11533682   cData[6]= g
address:  11533683   cData[7]= h
address:  11533684   cData[8]= i
address:  11533685   cData[9]= j
address:  11533686   cData[10]= k
address:  11533687   cData[11]= l
address:  11533572   dData[0]= 24.00
address:  11533580   dData[1]= 25.00
address:  11533588   dData[2]= 26.00
address:  11533596   dData[3]= 27.00
address:  11533604   dData[4]= 28.00
address:  11533612   dData[5]= 29.00
address:  11533620   dData[6]= 30.00
address:  11533628   dData[7]= 31.00
address:  11533636   dData[8]= 32.00
address:  11533644   dData[9]= 33.00
address:  11533652   dData[10]= 34.00
address:  11533660   dData[11]= 35.00
```