

# myMatrix.h

```
#include "../include/myMatrix.h"
```

자신의 폴더에 위치에 있는 myMatrix.h라는 헤더파일 가져오는 방법

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

```
typedef struct {
    double** at; // 2d array, 더블 포인터
    int rows, cols; // dimension 정보 저장
}Matrix;
```

행렬을 표현할 구조체 선언해주기, 앞으로 이런 형식으로 행렬을 선언해 줄 것이다.

행렬을 가져올 때 기본적으로 지켜야 될 규칙들

```
#define ASGN      999      // enter your assignment number
#define EVAL      0       // [! DO NOT EDIT !!!]

#include "../include/myMatrix.h"
int main(int argc, char* argv[])
{
    /* [! DO NOT EDIT !!!] Resources file path setting for evaluation */
    std::string path = "C:/NP_data/Assignment" + std::to_string(ASGN) + "/";

    #if EVAL
        path += "eval/";
    #endif
    return 0;
}
```

## basic Matrix

- [createMat\(\)](#)
- [freeMat\(\)](#)
- [txt2Mat\(\)](#)
- [printMat\(\)](#)

### createMat()

Matrix를 새롭게 만들기 위해 사용한다.

행렬의 크기가 (\_rows x \_cols)인 행렬을 만든다.

```
Matrix createMat(int _rows, int _cols);
```

## Parameters

\_rows: 행렬의 전체 행의 숫자를 의미

\_cols: 행렬의 전체 열의 숫자를 의미

## Example code

```
#define ASGN      999      // enter your assignment number
#define EVAL      0        // [! DO NOT EDIT !!!]

#include "../include/myMatrix.h"
int main(int argc, char* argv[])
{
    /*    [! DO NOT EDIT !!!]    Resources file path setting for evaluation */
    std::string path = "C:/NP_data/Assignment" + std::to_string(ASGN) + "/";

    #if EVAL
        path += "eval/";
    #endif
    int rows = 2;
    int cols = 2;
    Matrix matA = createMat(rows, cols);
    printMat(matA, "matA== ");

    return 0;
}
```

## output

```
matA==
-6277438562204192487878988888393020692503707483087375482269988814848.000000
-6277438562204192487878988888393020692503707483087375482269988814848.000000
-6277438562204192487878988888393020692503707483087375482269988814848.000000
-6277438562204192487878988888393020692503707483087375482269988814848.000000
```

값을 초기화 해주지 않아 이상한 값이 출력이 나오지만 2\*2행렬임을 볼 수 있다.

## Warning

- \_rows, \_cols는 int의 형식이어야 한다

## Error Handling

- \_rows, \_cols 둘 중에 하나라도 0보다 작으면 오류가 발생한다.

## freeMat()

만든 Matrix를 동적할당을 해제하기 위해 사용이 된다.

각 열들의 동적할당을 해주고 행들을 동적할당 해준다.

```
void freeMat(Matrix _A);
```

## Parameters

\_A: 동적할당을 해제할 행렬

## Example code

```
#define ASGN      999      // enter your assignment number
#define EVAL      0        // [! DO NOT EDIT !!!]

#include "../include/myMatrix.h"
int main(int argc, char* argv[])
{
    /* [! DO NOT EDIT !!!] Resources file path setting for evaluation */
    std::string path = "C:/NP_data/Assignment" + std::to_string(ASGN) + "/";

    #if EVAL
        path += "eval/";
    #endif
    int rows = 2;
    int cols = 2;
    Matrix matA = createMat(rows, cols);
    freeMat(matA);

    return 0;
}
```

출력이 되지는 않지만 해당 행렬의 동적할당을 해제해준다.

## Warning

- 행렬을 입력해주어야 한다.
- 동적할당을 해제하지 않으면 메모리를 계속 사용을 하게 된다.

## txt2Mat()

메모장에 적은 행렬을 불러온다.

행은 tab으로 열은 enter로 구분을 실시한다.

행렬의 저장위치는 "C:/NP\_data/Assignment"이다.

```
Matrix txt2Mat(std::string _filePath, std::string _fileName);
```

## Parameters

std::string \_filePath: 파일의 현재 위치를 의미한다

std::string \_fileName: 파일의 제목을 의미한다

## Example code

```
#define ASGN      999      // enter your assignment number
#define EVAL      0        // [! DO NOT EDIT !!!]
```

```
#include "../include/myMatrix.h"
int main(int argc, char* argv[])
{
    /*  [! DO NOT EDIT !!!]  Resources file path setting for evaluation */
    std::string path = "C:/NP_data/Assignment" + std::to_string(ASGN) + "/";

    #if EVAL
        path += "eval/";
    #endif
    Matrix matA = txt2Mat(path, "prob1_matA");
    printMat(matA, "matA ");

    return 0;
}
```

## output

```
matA
      4.000000      -2.000000      -3.000000      6.000000
     -6.000000      7.000000      6.500000     -6.000000
      1.000000      7.500000      6.250000      5.500000
     -12.000000     22.000000     15.500000     -1.000000
```

## Warning

- path를 마음대로 바꿔서는 안 된다.

## Error Handling

- 파일이 열리지 않으면 오류가 발생한다.

# printMat()

행렬을 출력하기 위해 사용을 한다. 출력 시 소수점 6자리까지 출력을 실시한다.

```
void    printMat(Matrix _A, const char* _name);;
```

## Parameters

\_A: 출력을 할 행렬

\_name: 출력을 할 문자열을 의미 혹은 출력을 할 제목을 의미

## Example code

```
#define ASGN      999      // enter your assignment number
#define EVAL      0      // [! DO NOT EDIT !!!]

#include "../include/myMatrix.h"
int main(int argc, char* argv[])
{
    /*  [! DO NOT EDIT !!!]  Resources file path setting for evaluation */
    std::string path = "C:/NP_data/Assignment" + std::to_string(ASGN) + "/";
```

```
#if EVAL
    path += "eval/";
#endif
    Matrix matA = txt2Mat(path, "prob1_matA");
    printMat(matA, "matA ");

    return 0;
}
```

## output

matA				
	4.000000	-2.000000	-3.000000	6.000000
	-6.000000	7.000000	6.500000	-6.000000
	1.000000	7.500000	6.250000	5.500000
	-12.000000	22.000000	15.500000	-1.000000

## Warning

- \_A는 행렬이어야 한다.
- \_name은 문자열이어야 한다.