
TASK 2 - Understanding Design Patterns in C

matrix.h

```
#ifndef MAT
#define MAT

#define DELETED 2
#define MATRIX_NOT_FOUND 3

typedef struct matrix matrix_t;

matrix_t* matrix_create(int n_i, int n_j, int* status); // for creating a new matrix
int matrix_del(matrix_t* matrix); // deleting the matrix

#endif
```

matrix.c

```
#include<stdio.h>
#include "matrix.h"
#include<errno.h>

extern int errno;

typedef struct matrix{
    int i,j; //rows and columns
    float* value; // data
}matrix_t;

matrix_t* matrix_create(int n_i, int n_j, int* status);
{
    if(n_i <= 0 || n_j <= 0)
    {
        *status = errno;
        printf("Invalid Parameters Provided, Exiting...")
        exit(1)
    }
    matrix_t* matrix = malloc(sizeof(matrix_t));
    matrix->i=n_i;
    matrix->j=n_j;
    matrix->value = malloc( matrix->i * matrix->j * sizeof(float));
    return matrix;
}
```

```
int matrix_del(matrix_t* matrix);
{
    if (matrix == NULL)
    {
        return MATRIX_NOT_FOUND;
    }
    else{
        free(matrix->value);
        free(matrix);
        return DELETE;
    }
}
```