
TASK 2 - Understanding Design Patterns in C

---

matrix.h

```
#ifndef MAT
#define MAT

#define SUCCESS 0
#define DELETED 1
#define MATRIX_NOT_FOUND 2
#define INVALID_DATA 3

typedef struct _matrix *matrix_t;

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

#endif
```

matrix.c

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

typdef struct _matrix{
	int i,j; //rows and columns
	float *value; // data
};


matrix_t *matrix_create(int n_i, int n_j);
{
	if(n_i > 0 && n_j > 0)
	{
		struct matrix_t *matrix = malloc(sizeof(*matrix));
		matrix->i=n_i;
		matrix->j=n_j;
		matrix->value = malloc( i * j * sizeof(float*));
		return SUCCESS;
	}
	else
	{
		return INVALID_DATA;
	}
}
```

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