TASK 2 - Understanding Design Patterns in C

---

matrix.h

```
#ifndef MAT
#define MAT

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

matrix *matrix_create(int n_i, int n_j); // for creating a new matrix
void matrix_print( const struct matrix *m); //for printing the matrix
void matrix_del(struct matrix *m); // deleting the matrix

#endif
```

matrix.c

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

matrix *matrix_create(int n_i, int n_j);
{
        struct matrix *matrix = malloc(sizeof(matrix))
        matrix->i=n_i;
        matrix->j=n_j;
        matrix->value = malloc( i * j * sizeof(float*)):
        return matrix
}

void matrix_print(const struct matrix* matrix) {
   for(int a = 0; a < matrix->i; a++) {
      printf("%f", "\n");
      for(int b = 0; b < matrix->j; b++) {
          printf("%f\t", matrix->value[a][b]);
      }
   }
}

void matrix_del(struct matrix* matrix)
{
        free(matrix->value)
        free(matrix)
}
```