```c
#include<stdlib.h>
#include<stdio.h>

/* Definição da estrutura de dados */

struct matriz {
  int lin;
  int col;
  float* v;
};

/* Defini9ção do tipo Matriz */
typedef struct matriz Matriz;

/* Cabeçalhos das funções */

Matriz* cria (int m, int n);
void libera (Matriz* mat);
float acessa (Matriz* mat, int i, int j);
void atribui (Matriz* mat, int i, int j, float v);
int linhas (Matriz* mat);
int colunas (Matriz* mat);


// Exemplo simples de programa que faz uso do TAD
int main(int argc, char *argv[])
{
  float a,b,c,d;
  Matriz *M;

  // criação de uma matriz
  M = cria(5,5);

  // inserção de valores na matriz
  atribui(M,1,2,40);
  atribui(M,2,3,3);
  atribui(M,3,0,15);
  atribui(M,4,1,21);

  // verificando se a inserção foi feita corretamente
  a = acessa(M,1,2);
  b = acessa(M,2,3);
  c = acessa(M,3,0);
  d = acessa(M,4,1);

  printf ("M[1][2]: %4.2f \n", a);
  printf ("M[2][3]: %4.2f \n", b);
  printf ("M[3][0]: %4.2f \n", c);
  printf ("M[4][1]: %4.2f \n", d);

  return 0;
}

/* Implementação das funções
 * do TAD Matriz */


/* Função cria
 * Aloca e retorna uma matriz de dimensão m por n */
Matriz* cria (int m, int n) {
  Matriz* mat = (Matriz*) malloc(sizeof(Matriz));
  if (mat == NULL) {
    printf("Memória insuficiente!\n");
    exit(1);
  }
  mat->lin = m;
  mat->col = n;
  mat->v = (float*) malloc(m*n*sizeof(float));
```

```
  return mat;
}

/* Função libera
 * Libera a memória de uma matriz previamente criada. */
void libera (Matriz* mat){
  free(mat->v);
  free(mat);
}


/* Função acessa
 * Retorna o valor do elemento da linha i e coluna j da matriz */
float acessa (Matriz* mat, int i, int j) {
  int k; /* índice do elemento no vetor */

  if (i<0 || i>=mat->lin || j<0 || j>=mat->col) {
        printf("Acesso inválido!\n");
        exit(1);
  }
  k = i*mat->col + j;
  return mat->v[k];
}

/* Função atribui
 * Atribui o valor dado ao elemento da linha i e coluna j da matriz */
void atribui (Matriz* mat, int i, int j, float v) {
  int k; /* índice do elemento no vetor */

  if (i<0 || i>=mat->lin || j<0 || j>=mat->col) {
    printf("Atribuição inválida!\n");
    exit(1);
  }
  k = i*mat->col + j;
  mat->v[k] = v;
}

/* Função linhas
 * Retorna o número de linhas da matriz */
int linhas (Matriz* mat) {
  return mat->lin;
}


/* Função colunas
 * Retorna o número de colunas da matriz */
int colunas (Matriz* mat) {
  return mat->col;
}
```