

My Project

Generated by Doxygen 1.10.0

1 Data Structure Index	1
1.1 Data Structures	1
2 File Index	3
2.1 File List	3
3 Data Structure Documentation	5
3.1 Matrix Struct Reference	5
3.1.1 Detailed Description	5
3.1.2 Field Documentation	5
3.1.2.1 colHead	5
3.1.2.2 rowHead	5
3.2 Node Struct Reference	6
3.2.1 Detailed Description	6
3.2.2 Field Documentation	6
3.2.2.1 nextCol	6
3.2.2.2 nextRow	6
3.2.2.3 value	6
4 File Documentation	7
4.1 Functions.c File Reference	7
4.1.1 Detailed Description	7
4.1.2 Macro Definition Documentation	8
4.1.2.1 INF	8
4.1.3 Function Documentation	8
4.1.3.1 changeConstant()	8
4.1.3.2 Function: changeConstant	8
4.1.3.3 createNode()	8
4.1.3.4 Function: createNode	8
4.1.3.5 displayMatrix()	8
4.1.3.6 Function: displayMatrix	8
4.1.3.7 displayTabularListing()	8
4.1.3.8 Function: displayTabularListing	9
4.1.3.9 insertColEnd()	9
4.1.3.10 Function: insertColEnd	9
4.1.3.11 insertNewColumn()	9
4.1.3.12 insertNewRow()	9
4.1.3.13 Function: insertNewRow	9
4.1.3.14 readMatrixFromFile()	9
4.1.3.15 Function: readMatrixFromFile	9
4.1.3.16 removeColumn()	10
4.1.3.17 Function: removeColumn	10
4.1.3.18 removeRow()	10

4.1.3.19 Function: removeRow	10
4.2 Header.h File Reference	10
4.2.1 Detailed Description	11
4.2.2 Macro Definition Documentation	11
4.2.2.1 MAX_SIZE	11
4.2.3 Typedef Documentation	11
4.2.3.1 Matrix	11
4.2.3.2 Node	11
4.2.4 Function Documentation	12
4.2.4.1 changeConstant()	12
4.2.4.2 Function: changeConstant	12
4.2.4.3 createNode()	12
4.2.4.4 Function: createNode	12
4.2.4.5 displayMatrix()	12
4.2.4.6 Function: displayMatrix	12
4.2.4.7 displayTabularListing()	12
4.2.4.8 Function: displayTabularListing	12
4.2.4.9 insertColEnd()	13
4.2.4.10 Function: insertColEnd	13
4.2.4.11 insertNewColumn()	13
4.2.4.12 insertNewRow()	13
4.2.4.13 Function: insertNewRow	13
4.2.4.14 readMatrixFromFile()	13
4.2.4.15 Function: readMatrixFromFile	13
4.2.4.16 removeColumn()	13
4.2.4.17 Function: removeColumn	14
4.2.4.18 removeRow()	14
4.2.4.19 Function: removeRow	14
4.3 Header.h	14
4.4 Main.c File Reference	14
4.4.1 Function Documentation	15
4.4.1.1 main()	15

Index	17
--------------	-----------

Chapter 1

Data Structure Index

1.1 Data Structures

Here are the data structures with brief descriptions:

Matrix	Struct que representa uma matriz	5
Node	Struct que representa um node de uma matriz	6

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

[Functions.c](#)

Este arquivo contém a implementação das funções do sistema. Essas funções abrangem a leitura de ficheiros, a criação de nodes, inserir e remover linhas e colunas na matriz, alteração de valores na matriz

7

[Header.h](#)

Este arquivo é uma livreria que contém as declarações das funções e structs utilizadas no sistema de forma a facilitar a reutilização do código

10

[Main.c](#)

.

14

Chapter 3

Data Structure Documentation

3.1 Matrix Struct Reference

Struct que representa uma matriz.

```
#include <Header.h>
```

Data Fields

- [Node](#) * `rowHead`
- [Node](#) * `colHead`

3.1.1 Detailed Description

Struct que representa uma matriz.

Esta struct representa uma matriz, que é composta por pointers que apontam para o início de uma linha e coluna na matriz.

3.1.2 Field Documentation

3.1.2.1 `colHead`

```
Node* colHead
```

3.1.2.2 `rowHead`

```
Node* rowHead
```

The documentation for this struct was generated from the following file:

- [Header.h](#)

3.2 Node Struct Reference

Struct que representa um node de uma matriz.

```
#include <Header.h>
```

Data Fields

- int [value](#)
- struct [Node](#) * [nextRow](#)
- struct [Node](#) * [nextCol](#)

3.2.1 Detailed Description

Struct que representa um node de uma matriz.

Esta struct representa um node de uma matriz, contendo um valor inteiro, bem como informações sobre a posição do node na matriz (linha e coluna). Além disso, possui ponteiros para o próximo node na mesma linha ([nextRow](#)) e na mesma coluna ([nextCol](#)).

3.2.2 Field Documentation

3.2.2.1 [nextCol](#)

```
struct Node* nextCol
```

3.2.2.2 [nextRow](#)

```
struct Node* nextRow
```

3.2.2.3 [value](#)

```
int value
```

The documentation for this struct was generated from the following file:

- [Header.h](#)

Chapter 4

File Documentation

4.1 Functions.c File Reference

Este arquivo contém a implementação das funções do sistema. Essas funções abrangem a leitura de ficheiros, a criação de nodes, inserir e remover linhas e colunas na matriz, alteração de valores na matriz.

```
#include <stdio.h>
#include <stdlib.h>
#include "Header.h"
#include <limits.h>
```

Macros

- #define `INF` INT_MAX

Functions

- `Matrix * readMatrixFromFile` (const char *filename)
- `Node * createNode` (int value)
- bool `displayMatrix` (Matrix *matrix)
- `Node * insertColEnd` (Node *head, int value)
- `Matrix * changeConstant` (Matrix *matrix, int rowIndex, int colIndex, int newValue)
- `Matrix * insertNewRow` (Matrix *matrix)
- `Matrix * insertNewColumn` (Matrix *matrix)
- `Matrix * removeRow` (Matrix *matrix, int rowIndex)
- `Matrix * removeColumn` (Matrix *matrix, int colIndex)
- bool `displayTabularListing` (Matrix *matrix)

4.1.1 Detailed Description

Este arquivo contém a implementação das funções do sistema. Essas funções abrangem a leitura de ficheiros, a criação de nodes, inserir e remover linhas e colunas na matriz, alteração de valores na matriz.

Author

Alexandre Vilaça 26590

Date

25.05.2024

4.1.2 Macro Definition Documentation

4.1.2.1 INF

```
#define INF INT_MAX
```

4.1.3 Function Documentation

4.1.3.1 changeConstant()

```
Matrix * changeConstant (
    Matrix * matrix,
    int rowIndex,
    int colIndex,
    int newValue )
```

4.1.3.2 Function: changeConstant

Esta função atualiza todos os valores na matriz com um novo valor. Percorre cada node e atualiza o seu valor para a nova constante.

4.1.3.3 createNode()

```
Node * createNode (
    int value )
```

4.1.3.4 Function: createNode

A função createNode cria um novo node para a matriz e aloca memória para o node, inicializa o seu valor com o input fornecido e define os pointers para a próxima linha e coluna com o valor NULL para defenir como fim da lista.

4.1.3.5 displayMatrix()

```
bool displayMatrix (
    Matrix * matrix )
```

4.1.3.6 Function: displayMatrix

A função displayMatrix mostra a matriz armazenada na memória, onde percorre cada linha e coluna da matriz a partir de pointers e imprime os seus valores.

4.1.3.7 displayTabularListing()

```
bool displayTabularListing (
    Matrix * matrix )
```

4.1.3.8 Function: displayTabularListing

Esta função mostra a matriz num formato tabular. É parecida com a [displayMatrix\(\)](#), percorre cada linha e coluna da matriz e imprime os valores só que no formato tabular.

4.1.3.9 insertColEnd()

```
Node * insertColEnd (
    Node * head,
    int value )
```

4.1.3.10 Function: insertColEnd

A seguinte função insere um novo node no final de uma coluna na matriz. Se a coluna estiver vazia, define o início com o novo node, caso contrário, percorre a coluna até encontrar o último node e insere-o lá.

4.1.3.11 insertNewColumn()

```
Matrix * insertNewColumn (
    Matrix * matrix )
```

A função insertNewColumn insere uma nova coluna no final da matriz, cria uma nova coluna com valores inseridos pelo utilizador e coloca-os no final da matriz.

4.1.3.12 insertNewRow()

```
Matrix * insertNewRow (
    Matrix * matrix )
```

4.1.3.13 Function: insertNewRow

A função insertNewRow insere uma nova linha no final da matriz, cria uma nova linha com valores inseridos pelo utilizador e coloca-os no final da matriz.

4.1.3.14 readMatrixFromFile()

```
Matrix * readMatrixFromFile (
    const char * filename )
```

4.1.3.15 Function: readMatrixFromFile

Esta função recebe um nome de ficheiro como entrada, abre o ficheiro em modo de leitura e lê o conteúdo do ficheiro e cria nodes conforme os valores do ficheiro. A matriz é representada como uma lista ligada de nós, onde cada nó contém um valor e ponteiros para o próximo node na linha e coluna.

4.1.3.16 removeColumn()

```
Matrix * removeColumn (
    Matrix * matrix,
    int colIndex )
```

4.1.3.17 Function: removeColumn

A seguinte função remove uma coluna específica, referida pelo utilizador, de uma matriz.

4.1.3.18 removeRow()

```
Matrix * removeRow (
    Matrix * matrix,
    int rowIndex )
```

4.1.3.19 Function: removeRow

A seguinte função remove uma linha específica, referida pelo utilizador, de uma matriz.

4.2 Header.h File Reference

Este arquivo é uma livreria que contém as declarações das funções e structs utilizadas no sistema de forma a facilitar a reutilização do código.

```
#include <stdbool.h>
```

Data Structures

- struct [Node](#)
Struct que representa um node de uma matriz.
- struct [Matrix](#)
Struct que representa uma matriz.

Macros

- #define [MAX_SIZE](#) 100

Typedefs

- typedef struct Node [Node](#)
Struct que representa um node de uma matriz.
- typedef struct Matrix [Matrix](#)
Struct que representa uma matriz.

Functions

- [Matrix * readMatrixFromFile](#) (const char *filename)
- [Node * createNode](#) (int value)
- bool [displayMatrix](#) (Matrix *matrix)
- [Node * insertColEnd](#) (Node *head, int value)
- [Matrix * changeConstant](#) (Matrix *matrix, int rowIndex, int colIndex, int newValue)
- [Matrix * insertNewRow](#) (Matrix *matrix)
- [Matrix * insertNewColumn](#) (Matrix *matrix)
- [Matrix * removeRow](#) (Matrix *matrix, int rowIndex)
- [Matrix * removeColumn](#) (Matrix *matrix, int colIndex)
- bool [displayTabularListing](#) (Matrix *matrix)

4.2.1 Detailed Description

Este arquivo é uma livreria que contém as declarações das funções e structs utilizadas no sistema de forma a facilitar a reutilização do código.

Author

Alexandre Vilaça 26590

Date

25.05.2024

4.2.2 Macro Definition Documentation

4.2.2.1 MAX_SIZE

```
#define MAX_SIZE 100
```

4.2.3 Typedef Documentation

4.2.3.1 Matrix

```
typedef struct Matrix Matrix
```

Struct que representa uma matriz.

Esta struct representa uma matriz, que é composta por pointers que apontam para o início de uma linha e coluna na matriz.

4.2.3.2 Node

```
typedef struct Node Node
```

Struct que representa um node de uma matriz.

Esta struct representa um node de uma matriz, contendo um valor inteiro, bem como informações sobre a posição do node na matriz (linha e coluna). Além disso, possui ponteiros para o próximo node na mesma linha (nextRow) e na mesma coluna (nextCol).

4.2.4 Function Documentation

4.2.4.1 changeConstant()

```
Matrix * changeConstant (
    Matrix * matrix,
    int rowIndex,
    int colIndex,
    int newValue )
```

4.2.4.2 Function: changeConstant

Esta função atualiza todos os valores na matriz com um novo valor. Percorre cada node e atualiza o seu valor para a nova constante.

4.2.4.3 createNode()

```
Node * createNode (
    int value )
```

4.2.4.4 Function: createNode

A função createNode cria um novo node para a matriz e aloca memória para o node, inicializa o seu valor com o input fornecido e define os pointers para a próxima linha e coluna com o valor NULL para definir como fim da lista.

4.2.4.5 displayMatrix()

```
bool displayMatrix (
    Matrix * matrix )
```

4.2.4.6 Function: displayMatrix

A função displayMatrix mostra a matriz armazenada na memória, onde percorre cada linha e coluna da matriz a partir de pointers e imprime os seus valores.

4.2.4.7 displayTabularListing()

```
bool displayTabularListing (
    Matrix * matrix )
```

4.2.4.8 Function: displayTabularListing

Esta função mostra a matriz num formato tabular. É parecida com a [displayMatrix\(\)](#), percorre cada linha e coluna da matriz e imprime os valores só que no formato tabular.

4.2.4.9 insertColEnd()

```
Node * insertColEnd (
    Node * head,
    int value )
```

4.2.4.10 Function: insertColEnd

A seguinte função insere um novo node no final de uma coluna na matriz. Se a coluna estiver vazia, define o início com o novo node, caso contrário, percorre a coluna até encontrar o último node e insere-o lá.

4.2.4.11 insertNewColumn()

```
Matrix * insertNewColumn (
    Matrix * matrix )
```

A função insertNewColumn insere uma nova coluna no final da matriz, cria uma nova coluna com valores inseridos pelo utilizador e coloca-os no final da matriz.

4.2.4.12 insertNewRow()

```
Matrix * insertNewRow (
    Matrix * matrix )
```

4.2.4.13 Function: insertNewRow

A função insertNewRow insere uma nova linha no final da matriz, cria uma nova linha com valores inseridos pelo utilizador e coloca-os no final da matriz.

4.2.4.14 readMatrixFromFile()

```
Matrix * readMatrixFromFile (
    const char * filename )
```

4.2.4.15 Function: readMatrixFromFile

Esta função recebe um nome de ficheiro como entrada, abre o ficheiro em modo de leitura e lê o conteúdo do ficheiro e cria nodes conforme os valores do ficheiro. A matriz é representada como uma lista ligada de nós, onde cada nó contém um valor e ponteiros para o próximo node na linha e coluna.

4.2.4.16 removeColumn()

```
Matrix * removeColumn (
    Matrix * matrix,
    int colIndex )
```

4.2.4.17 Function: removeColumn

A seguinte função remove uma coluna específica, referida pelo utilizador, de uma matriz.

4.2.4.18 removeRow()

```
Matrix * removeRow (
    Matrix * matrix,
    int rowIndex )
```

4.2.4.19 Function: removeRow

A seguinte função remove uma linha específica, referida pelo utilizador, de uma matriz.

4.3 Header.h

[Go to the documentation of this file.](#)

```
00001 #include <stdbool.h>
00002 #define MAX_SIZE 100
00018 typedef struct Node
00019 {
00020     int value;
00021     struct Node *nextRow;
00022     struct Node *nextCol;
00023 } Node;
00024
00030 typedef struct Matrix
00031 {
00032     Node *rowHead;
00033     Node *colHead;
00034 } Matrix;
00035
00042 Matrix *readMatrixFromFile(const char *filename);
00043
00050 Node *createNode(int value);
00051
00057 bool displayMatrix(Matrix *matrix);
00058
00065 Node *insertColEnd(Node *head, int value);
00066
00073 Matrix *changeConstant(Matrix *matrix, int rowIndex, int colIndex, int newValue);
00079 Matrix *insertNewRow(Matrix *matrix);
00080
00084 Matrix *insertNewColumn(Matrix *matrix);
00085
00091 Matrix *removeRow(Matrix *matrix, int rowIndex);
00092
00098 Matrix *removeColumn(Matrix *matrix, int colIndex);
00099
00106 bool displayTabularListing(Matrix *matrix);
00107
```

4.4 Main.c File Reference

```
#include "Header.h"
#include <stdio.h>
#include <stdbool.h>
```

Functions

- int `main` ()

Função principal que gerencia a execução do programa.

4.4.1 Function Documentation

4.4.1.1 `main()`

```
int main ( )
```

Função principal que gerencia a execução do programa.

Author

Alexandre Vilaça 26590

Date

25.05.2024

Returns

Retorna 0 se a execução foi concluída com sucesso.

Index

- changeConstant
 - Functions.c, [8](#)
 - Header.h, [12](#)
- colHead
 - Matrix, [5](#)
- createNode
 - Functions.c, [8](#)
 - Header.h, [12](#)
- displayMatrix
 - Functions.c, [8](#)
 - Header.h, [12](#)
- displayTabularListing
 - Functions.c, [8](#)
 - Header.h, [12](#)
- Functions.c, [7](#)
 - changeConstant, [8](#)
 - createNode, [8](#)
 - displayMatrix, [8](#)
 - displayTabularListing, [8](#)
 - INF, [8](#)
 - insertColEnd, [9](#)
 - insertNewColumn, [9](#)
 - insertNewRow, [9](#)
 - readMatrixFromFile, [9](#)
 - removeColumn, [9](#)
 - removeRow, [10](#)
- Header.h, [10](#), [14](#)
 - changeConstant, [12](#)
 - createNode, [12](#)
 - displayMatrix, [12](#)
 - displayTabularListing, [12](#)
 - insertColEnd, [12](#)
 - insertNewColumn, [13](#)
 - insertNewRow, [13](#)
 - Matrix, [11](#)
 - MAX_SIZE, [11](#)
 - Node, [11](#)
 - readMatrixFromFile, [13](#)
 - removeColumn, [13](#)
 - removeRow, [14](#)
- INF
 - Functions.c, [8](#)
- insertColEnd
 - Functions.c, [9](#)
 - Header.h, [12](#)
- insertNewColumn
 - Functions.c, [9](#)
 - Header.h, [13](#)
- insertNewRow
 - Functions.c, [9](#)
 - Header.h, [13](#)
- main
 - Main.c, [15](#)
- Main.c, [14](#)
 - main, [15](#)
- Matrix, [5](#)
 - colHead, [5](#)
 - Header.h, [11](#)
 - rowHead, [5](#)
- MAX_SIZE
 - Header.h, [11](#)
- nextCol
 - Node, [6](#)
- nextRow
 - Node, [6](#)
- Node, [6](#)
 - Header.h, [11](#)
 - nextCol, [6](#)
 - nextRow, [6](#)
 - value, [6](#)
- readMatrixFromFile
 - Functions.c, [9](#)
 - Header.h, [13](#)
- removeColumn
 - Functions.c, [9](#)
 - Header.h, [13](#)
- removeRow
 - Functions.c, [10](#)
 - Header.h, [14](#)
- rowHead
 - Matrix, [5](#)
- value
 - Node, [6](#)