

Documentation: Méthode de Gauss

Generated by Doxygen 1.9.1

1 File Index	1
1.1 File List	1
2 File Documentation	3
2.1 gauss.c File Reference	3
2.1.1 Macro Definition Documentation	4
2.1.1.1 TAILLE_MATRICE	4
2.1.1.2 VECTEUR_COLONNE	4
2.1.2 Function Documentation	4
2.1.2.1 methodeDeGauss()	4
2.1.2.2 permutationLigne()	5
2.1.2.3 valeurAbsolue()	6
2.2 gauss.h File Reference	7
2.2.1 Macro Definition Documentation	7
2.2.1.1 TAILLE_MATRICE	7
2.2.1.2 VECTEUR_COLONNE	7
2.2.2 Function Documentation	8
2.2.2.1 methodeDeGauss()	8
2.2.2.2 permutationLigne()	9
2.2.2.3 valeurAbsolue()	10
2.3 main.c File Reference	10
2.3.1 Detailed Description	11
2.3.2 Function Documentation	11
2.3.2.1 main()	11
2.4 matriceTest.c File Reference	12
2.4.1 Detailed Description	14
2.4.2 Macro Definition Documentation	14
2.4.2.1 TAILLE_MATRICE	15
2.4.3 Function Documentation	15
2.4.3.1 afficheMatrice()	15
2.4.3.2 creationMatriceDynamique()	16
2.4.3.3 liberationMemoireMatrice()	18
2.4.3.4 matriceA1Dynamique()	19
2.4.3.5 matriceA2Dynamique()	20
2.4.3.6 matriceA3Dynamique()	21
2.4.3.7 matriceA4Dynamique()	21
2.4.3.8 matriceA5Dynamique()	22
2.4.3.9 matriceA6Dynamique()	23
2.4.3.10 matriceB1Dynamique()	24
2.4.3.11 matriceB2Dynamique()	25
2.4.3.12 matriceB3Dynamique()	25
2.4.3.13 matriceB4Dynamique()	26

2.4.3.14 matriceB5_10Dynamique()	27
2.4.3.15 matriceB5_3Dynamique()	28
2.4.3.16 matriceB5_6Dynamique()	28
2.4.3.17 matriceB5_8Dynamique()	29
2.4.3.18 matriceB6_10Dynamique()	30
2.4.3.19 matriceB6_3Dynamique()	31
2.4.3.20 matriceB6_6Dynamique()	31
2.4.3.21 matriceB6_8Dynamique()	32
2.5 matriceTest.h File Reference	33
2.5.1 Function Documentation	34
2.5.1.1 afficheMatrice()	34
2.5.1.2 creationMatriceDynamique()	35
2.5.1.3 liberationMemoireMatrice()	36
2.5.1.4 matriceA1Dynamique()	37
2.5.1.5 matriceA2Dynamique()	38
2.5.1.6 matriceA3Dynamique()	39
2.5.1.7 matriceA4Dynamique()	39
2.5.1.8 matriceA5Dynamique()	40
2.5.1.9 matriceA6Dynamique()	41
2.5.1.10 matriceB1Dynamique()	42
2.5.1.11 matriceB2Dynamique()	43
2.5.1.12 matriceB3Dynamique()	43
2.5.1.13 matriceB4Dynamique()	44
2.5.1.14 matriceB5_10Dynamique()	45
2.5.1.15 matriceB5_3Dynamique()	46
2.5.1.16 matriceB5_6Dynamique()	46
2.5.1.17 matriceB5_8Dynamique()	47
2.5.1.18 matriceB6_10Dynamique()	48
2.5.1.19 matriceB6_3Dynamique()	49
2.5.1.20 matriceB6_6Dynamique()	49
2.5.1.21 matriceB6_8Dynamique()	50
Index	53

Chapter 1

File Index

1.1 File List

Here is a list of all files with brief descriptions:

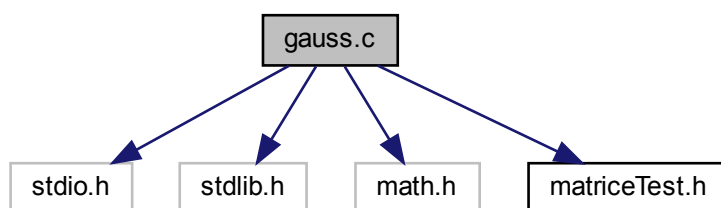
gauss.c	3
gauss.h	7
main.c	
	Contients l'appel à toute les fonctions pour résoudre les systèmes avec les matrices tests proposées	
	10
matriceTest.c	
	Contients l'implémentation de la méthode de Gauss	
	12
matriceTest.h	33

Chapter 2

File Documentation

2.1 gauss.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "matriceTest.h"
Include dependency graph for gauss.c:
```



Macros

- `#define` `VECTEUR_COLONNE` 0
Correspond à l'indice des colonnes d'un vecteur colonne.
- `#define` `TAILLE_MATRICE` 3
Correspond à la taille de la matrice le plus souvent considéré.

Functions

- double `valeurAbsolue` (double `nbr`)
Permet de calculer la valeur absolue d'un flottant.
- int `permutationLigne` (double `**matrice`, unsigned int `taille`, unsigned int `Li`, unsigned int `Lj`)
Permet de permuter 2 lignes d'une même matrice.
- double `** methodeDeGauss` (double `**A`, unsigned int `n`, double `**B`, double `epsilon`)
Fonction qui appliquer la méthode de Gauss.

2.1.1 Macro Definition Documentation

2.1.1.1 TAILLE_MATRICE

```
#define TAILLE_MATRICE 3
```

Correspond à la taille de la matrice le plus souvent considérer.

Definition at line 25 of file gauss.c.

2.1.1.2 VECTEUR_COLONNE

```
#define VECTEUR_COLONNE 0
```

Correspond à l'indice des colonnes d'un vecteur colonne.

Definition at line 19 of file gauss.c.

2.1.2 Function Documentation

2.1.2.1 methodeDeGauss()

```
double** methodeDeGauss (  
    double ** A,  
    unsigned int n,  
    double ** B,  
    double epsilon )
```

Fonction qui appliquer la méthode de Gauss.

Parameters

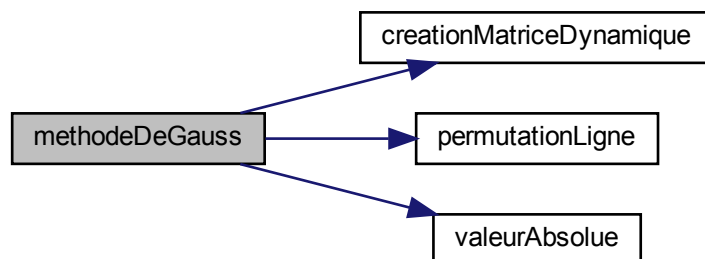
in	<i>A</i>	Matrice du système (qui sera triangulariser).
in	<i>B</i>	Matrice associée au second membre du système.
in	<i>n</i>	Correspond à la taille de A (matrice carrée).
in	<i>epsilon</i>	Correspond à la valeur que l'on se fixe, dans la méthode du pivot maximal , pour considérer un pivot presque nul.

Returns

Renvoie la matrice solution du système, **NULL** si problème.

Definition at line 80 of file gauss.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.1.2.2 permutationLigne()

```

int permutationLigne (
    double ** matrice,
    unsigned int taille,
    unsigned int Li,
    unsigned int Lj )
  
```

Permet de permuter 2 lignes d'une même matrice.

Parameters

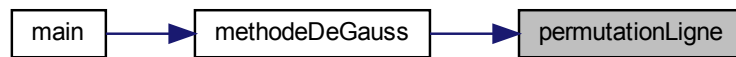
in	<i>matrice</i>	Matrice à afficher.
in	<i>taille</i>	Correspond à la taille de la matrice.
in	<i>Li</i>	Correspond à la ligne à permuter avec Lj .
in	<i>Lj</i>	Correspond à la ligne à permuter avec Li .

Returns

Renvoie 0 si la fonction se déroule normalement jusqu'à la fin.

Definition at line 55 of file gauss.c.

Here is the caller graph for this function:

**2.1.2.3 valeurAbsolue()**

```
double valeurAbsolue (  
    double nbr )
```

Permet de calculer la valeur absolue d'un flottant.

Parameters

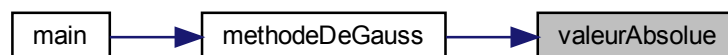
in	nbr	La valeur absolue du nombre à calculer.
----	-----	---

Returns

Renvoie la valeur absolue du nombre passé en argument.

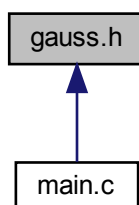
Definition at line 36 of file gauss.c.

Here is the caller graph for this function:



2.2 gauss.h File Reference

This graph shows which files directly or indirectly include this file:



Macros

- #define [VECTEUR_COLONNE](#) 0
- #define [TAILLE_MATRICE](#) 3

Functions

- double [valeurAbsolue](#) (double)
Permet de calculer la valeur absolue d'un flottant.
- int [permutationLigne](#) (double **, unsigned int, unsigned int, unsigned int)
Permet de permuter 2 lignes d'une même matrice.
- double ** [methodeDeGauss](#) (double **, unsigned int, double **, double)
Fonction qui appliquer la méthode de Gauss.

2.2.1 Macro Definition Documentation

2.2.1.1 TAILLE_MATRICE

```
#define TAILLE_MATRICE 3
```

Definition at line 7 of file gauss.h.

2.2.1.2 VECTEUR_COLONNE

```
#define VECTEUR_COLONNE 0
```

Definition at line 6 of file gauss.h.

2.2.2 Function Documentation

2.2.2.1 methodeDeGauss()

```
double** methodeDeGauss (
    double ** A,
    unsigned int n,
    double ** B,
    double epsilon )
```

Fonction qui applique la méthode de Gauss.

Parameters

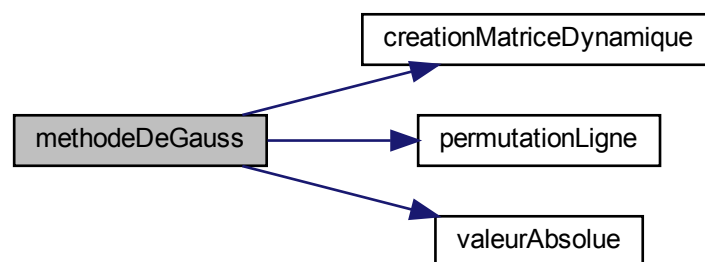
in	<i>A</i>	Matrice du système (qui sera triangulariser).
in	<i>B</i>	Matrice associée au second membre du système.
in	<i>n</i>	Correspond à la taille de A (matrice carrée).
in	<i>epsilon</i>	Correspond à la valeur que l'on se fixe, dans la méthode du pivot maximal , pour considérer un pivot presque nul.

Returns

Renvoie la matrice solution du système, **NULL** si problème.

Definition at line 80 of file gauss.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.2.2.2 permutationLigne()

```
int permutationLigne (  
    double ** matrice,  
    unsigned int taille,  
    unsigned int Li,  
    unsigned int Lj )
```

Permet de permuter 2 lignes d'une même matrice.

Parameters

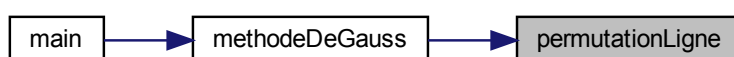
in	<i>matrice</i>	Matrice à afficher.
in	<i>taille</i>	Correspond à la taille de la matrice.
in	<i>Li</i>	Correspond à la ligne à permuter avec Lj .
in	<i>Lj</i>	Correspond à la ligne à permuter avec Li .

Returns

Renvoie 0 si la fonction se déroule normalement jusqu'à la fin.

Definition at line 55 of file gauss.c.

Here is the caller graph for this function:



2.2.2.3 valeurAbsolue()

```
double valeurAbsolue (
    double nbr )
```

Permet de calculer la valeur absolue d'un flottant.

Parameters

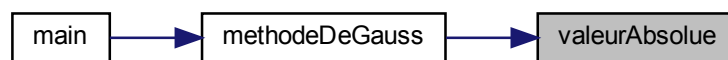
in	<i>nbr</i>	La valeur absolue du nombre à calculer.
----	------------	---

Returns

Renvoie la valeur absolue du nombre passé en argument.

Definition at line 36 of file gauss.c.

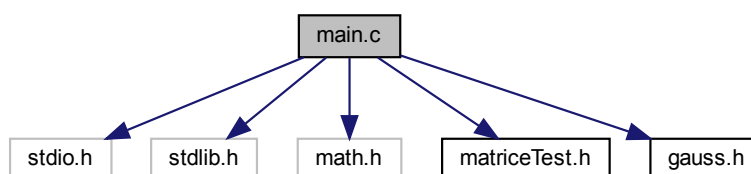
Here is the caller graph for this function:



2.3 main.c File Reference

Contient l'appel à toutes les fonctions pour résoudre les systèmes avec les matrices tests proposées.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include "matriceTest.h"
#include "gauss.h"
Include dependency graph for main.c:
```



Functions

- int `main` ()

2.3.1 Detailed Description

Contients l'appel à toute les fonctions pour résoudre les systèmes avec les matrices tests proposées.

Author

Valentin PORTAIL et Jérémie VILLEPREUX

Version

1.0

Date

02/10/2022

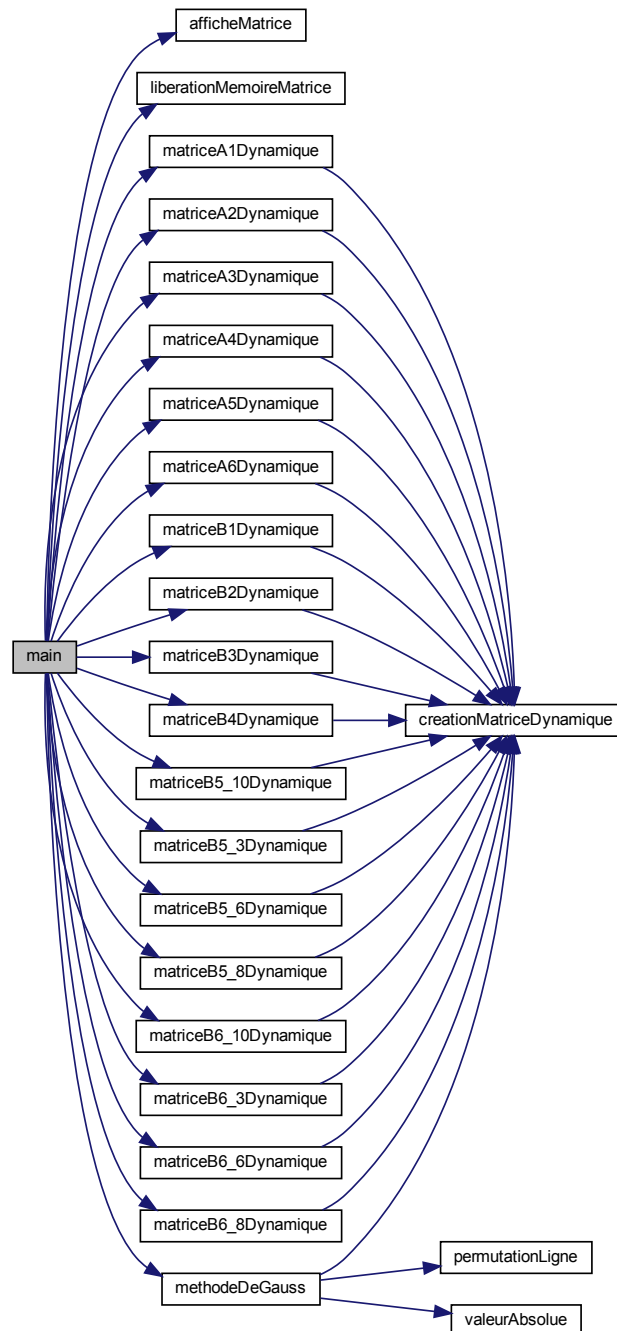
2.3.2 Function Documentation

2.3.2.1 `main()`

```
int main ( )
```

Definition at line 31 of file main.c.

Here is the call graph for this function:



2.4 matriceTest.c File Reference

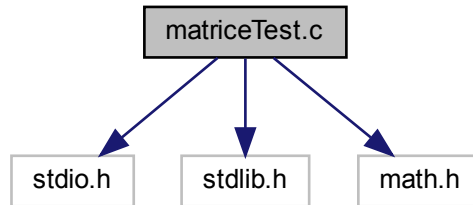
Contient l'implémentation de la méthode de Gauss.

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



```
#include <math.h>
```

Include dependency graph for matriceTest.c:



Macros

- `#define TAILLE_MATRICE 3`
Correspond à la taille de la matrice le plus souvent considérer.

Functions

- `int afficheMatrice (double **tab, unsigned int nbrLig, unsigned int nbrCol)`
Permet d'afficher les éléments d'une matrice.
- `double ** creationMatriceDynamique (unsigned int nbrLig, unsigned int nbrCol)`
Permet de créer une matrice de manière dynamique.
- `int liberationMemoireMatrice (double **matrice, unsigned int nbrLig)`
Permet de libérer la mémoire d'une matrice, qui a été créer dynamiquement.
- `double ** matriceA1Dynamique (void)`
Permet de créer dynamiquement la matrice A1 de l'énoncé.
- `double ** matriceB1Dynamique (void)`
Permet de créer dynamiquement la matrice B1 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A1.
- `double ** matriceA2Dynamique (void)`
Permet de créer dynamiquement la matrice A2 de l'énoncé.
- `double ** matriceB2Dynamique (void)`
Permet de créer dynamiquement la matrice B2 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A2.
- `double ** matriceA3Dynamique (void)`
Permet de créer dynamiquement la matrice A3 de l'énoncé.
- `double ** matriceB3Dynamique (void)`
Permet de créer dynamiquement la matrice B3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A3.
- `double ** matriceA4Dynamique (void)`
Permet de créer dynamiquement la matrice A4 de l'énoncé.
- `double ** matriceB4Dynamique (void)`
Permet de créer dynamiquement la matrice B4 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A4.
- `double ** matriceA5Dynamique (unsigned int taille)`

- Permet de créer dynamiquement la matrice A5 de l'énoncé.*
- double ** `matriceB5_3Dynamique` (void)
 - Permet de créer dynamiquement la matrice B5 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 3.*
- double ** `matriceB5_6Dynamique` (void)
 - Permet de créer dynamiquement la matrice B5 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 6.*
- double ** `matriceB5_8Dynamique` (void)
 - Permet de créer dynamiquement la matrice B5 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 8.*
- double ** `matriceB5_10Dynamique` (void)
 - Permet de créer dynamiquement la matrice B5 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 10.*
- double ** `matriceA6Dynamique` (unsigned int taille)
 - Permet de créer dynamiquement la matrice A6 de l'énoncé.*
- double ** `matriceB6_3Dynamique` (void)
 - Permet de créer dynamiquement la matrice B6 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 3.*
- double ** `matriceB6_6Dynamique` (void)
 - Permet de créer dynamiquement la matrice B6 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 6.*
- double ** `matriceB6_8Dynamique` (void)
 - Permet de créer dynamiquement la matrice B6 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 8.*
- double ** `matriceB6_10Dynamique` (void)
 - Permet de créer dynamiquement la matrice B6 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 10.*

2.4.1 Detailed Description

Contient l'implémentation de la méthode de Gauss.

Contient les matrices test proposées et des fonctions de bases pour créer/afficher ces matrices.

Author

Valentin PORTAIL et Jérémie VILLEPREUX

Version

1.0

Date

02/10/2022

2.4.2 Macro Definition Documentation

2.4.2.1 TAILLE_MATRICE

```
#define TAILLE_MATRICE 3
```

Correspond à la taille de la matrice le plus souvent considérer.

Definition at line 18 of file matriceTest.c.

2.4.3 Function Documentation

2.4.3.1 afficheMatrice()

```
int afficheMatrice (  
    double ** tab,  
    unsigned int nbrLig,  
    unsigned int nbrCol )
```

Permet d'afficher les éléments d'une matrice.

Parameters

in	<i>tab</i>	Matrice à afficher.
in	<i>nbrLig</i>	Correspond au nombre de ligne de la matrice à afficher.
in	<i>nbrCol</i>	Correspond au nombre de colonne de la matrice à afficher.

Returns

Renvoie 0 si la fonction se déroule normalement jusqu'à la fin.

Definition at line 31 of file matriceTest.c.

Here is the caller graph for this function:



2.4.3.2 creationMatriceDynamique()

```
double** creationMatriceDynamique (
    unsigned int nbrLig,
    unsigned int nbrCol )
```

Permet de créer une matrice de manière dynamique.

Parameters

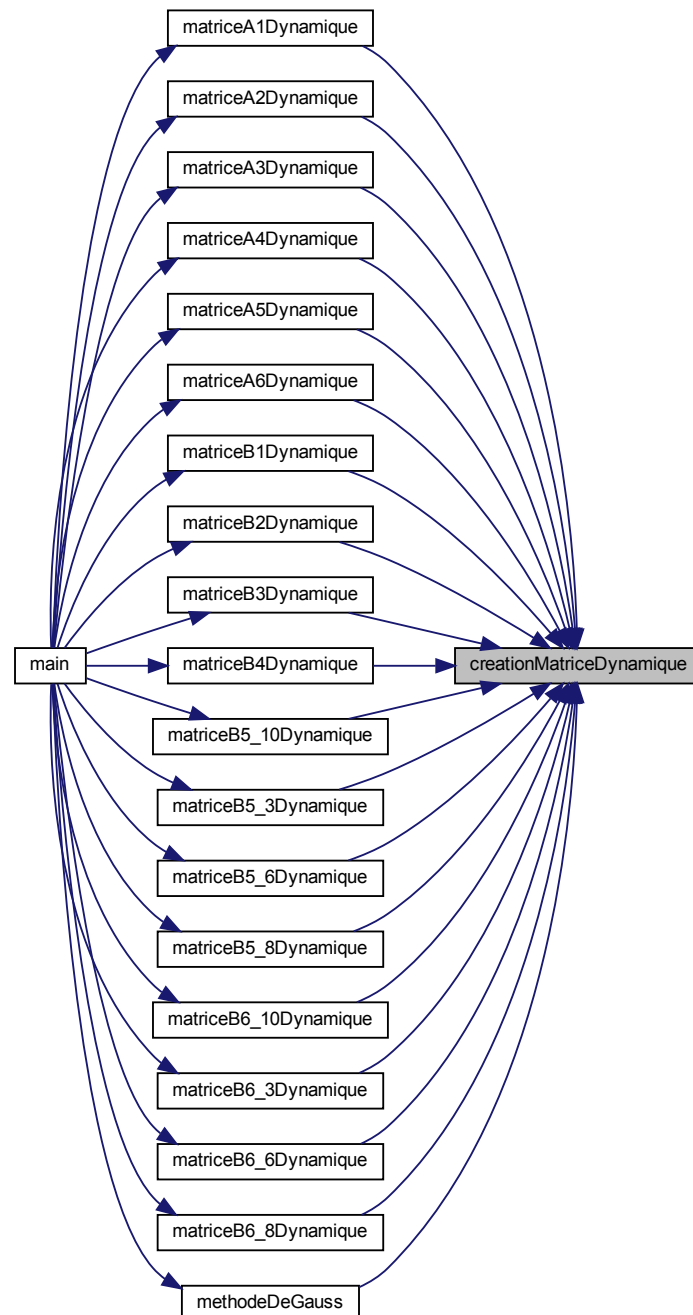
in	<i>nbrLig</i>	Correspond au nombre de ligne que l'on veut donner à la matrice .
in	<i>nbrCol</i>	Correspond au nombre de colonne que l'on veut donner à la matrice.

Returns

Renvoie le pointeur sur la matrice, vaut **NULL** si l'allocation à échouer.

Definition at line 61 of file `matriceTest.c`.

Here is the caller graph for this function:



2.4.3.3 liberationMemoireMatrice()

```

int liberationMemoireMatrice (
    double ** matrice,
    unsigned int nbrLig )

```

Permet de libérer la mémoire d'une matrice, qui a été créer dynamiquement.

Parameters

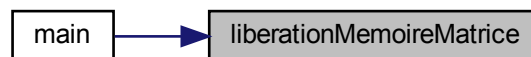
in	<i>matrice</i>	Correspond à la matrice à libérer.
in	<i>nbrLig</i>	Correspond au nombre de ligne de cette matrice.

Returns

Renvoie 0 si la focntion se déroule normalement jusqu'à la fin.

Definition at line 94 of file matriceTest.c.

Here is the caller graph for this function:



2.4.3.4 matriceA1Dynamique()

```
double** matriceA1Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A1 de l'énoncé.

Returns

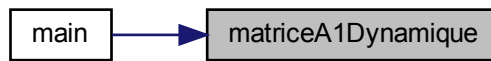
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 131 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.5 matriceA2Dynamique()

```
double** matriceA2Dynamique (
    void )
```

Permet de créer dynamiquement la matrice A2 de l'énoncé.

Returns

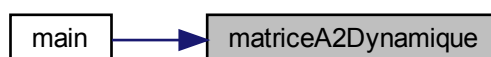
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 176 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.6 matriceA3Dynamique()

```
double** matriceA3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A3 de l'énoncé.

Returns

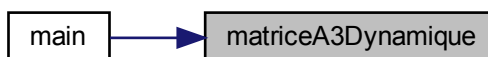
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 219 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.7 matriceA4Dynamique()

```
double** matriceA4Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A4 de l'énoncé.

Returns

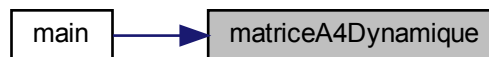
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 263 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**2.4.3.8 matriceA5Dynamique()**

```
double** matriceA5Dynamique (
    unsigned int taille )
```

Permet de créer dynamiquement la matrice A5 de l'énoncé.

Parameters

<code>in</code>	<code><i>taille</i></code>	Correspond à la taille que l'on veut donner à la matrice .
-----------------	----------------------------	--

Returns

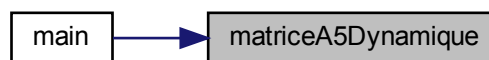
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 309 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.9 matriceA6Dynamique()

```
double** matriceA6Dynamique (
    unsigned int taille )
```

Permet de créer dynamiquement la matrice A6 de l'énoncé.

Parameters

in	<i>taille</i>	Correspond à la taille que l'on veut donner à la matrice .
----	---------------	--

Returns

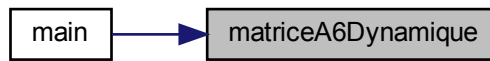
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 432 of file `matriceTest.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.10 `matriceB1Dynamique()`

```
double** matriceB1Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B1 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A1.

Returns

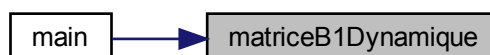
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 159 of file `matriceTest.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.11 matriceB2Dynamique()

```
double** matriceB2Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B2 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A2.

Returns

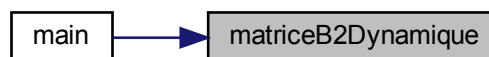
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 201 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.12 matriceB3Dynamique()

```
double** matriceB3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A3.

Returns

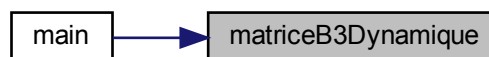
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 245 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**2.4.3.13 matriceB4Dynamique()**

```
double** matriceB4Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B4 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A4.

Returns

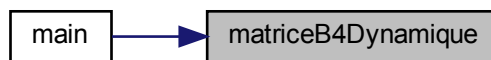
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 288 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.14 matriceB5_10Dynamique()

```
double** matriceB5_10Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 10.

Returns

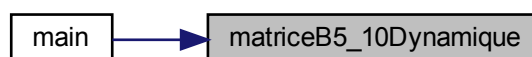
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 404 of file `matriceTest.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.15 matriceB5_3Dynamique()

```
double** matriceB5_3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 3.

Returns

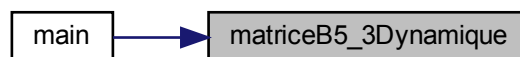
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 342 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.16 matriceB5_6Dynamique()

```
double** matriceB5_6Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 6.

Returns

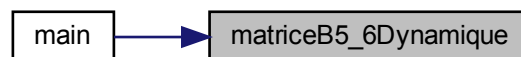
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 360 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.17 matriceB5_8Dynamique()

```
double** matriceB5_8Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 8.

Returns

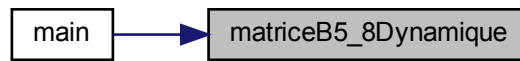
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 381 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.18 matriceB6_10Dynamique()

```
double** matriceB6_10Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 10.

Returns

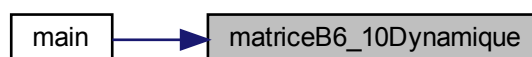
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 527 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.19 matriceB6_3Dynamique()

```
double** matriceB6_3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 3.

Returns

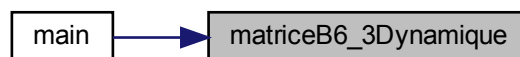
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 465 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.4.3.20 matriceB6_6Dynamique()

```
double** matriceB6_6Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 6.

Returns

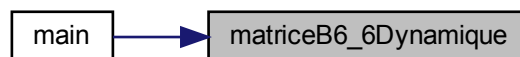
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 483 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**2.4.3.21 matriceB6_8Dynamique()**

```
double** matriceB6_8Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 8.

Returns

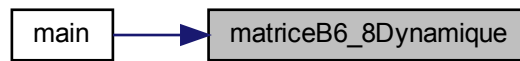
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 504 of file matriceTest.c.

Here is the call graph for this function:

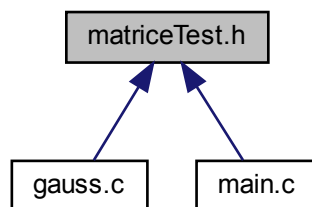


Here is the caller graph for this function:



2.5 matriceTest.h File Reference

This graph shows which files directly or indirectly include this file:



Functions

- int [afficheMatrice](#) (double **, unsigned int, unsigned int)
Permet d'afficher les éléments d'une matrice.
- double ** [creationMatriceDynamique](#) (unsigned int, unsigned int)
Permet de créer une matrice de manière dynamique.
- int [liberationMemoireMatrice](#) (double **, unsigned int)
Permet de libérer la mémoire d'une matrice, qui a été créer dynamiquement.
- double ** [matriceA1Dynamique](#) (void)
Permet de créer dynamiquement la matrice A1 de l'énoncé.
- double ** [matriceA2Dynamique](#) (void)
Permet de créer dynamiquement la matrice A2 de l'énoncé.
- double ** [matriceA3Dynamique](#) (void)
Permet de créer dynamiquement la matrice A3 de l'énoncé.
- double ** [matriceA4Dynamique](#) (void)
Permet de créer dynamiquement la matrice A4 de l'énoncé.
- double ** [matriceA5Dynamique](#) (unsigned int)
Permet de créer dynamiquement la matrice A5 de l'énoncé.
- double ** [matriceA6Dynamique](#) (unsigned int)
Permet de créer dynamiquement la matrice A6 de l'énoncé.

- double ** [matriceB1Dynamique](#) (void)
Permet de créer dynamiquement la matrice B1 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A1.
- double ** [matriceB2Dynamique](#) (void)
Permet de créer dynamiquement la matrice B2 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A2.
- double ** [matriceB3Dynamique](#) (void)
Permet de créer dynamiquement la matrice B3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A3.
- double ** [matriceB4Dynamique](#) (void)
Permet de créer dynamiquement la matrice B4 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A4.
- double ** [matriceB5_3Dynamique](#) (void)
Permet de créer dynamiquement la matrice B5 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 3.
- double ** [matriceB5_6Dynamique](#) (void)
Permet de créer dynamiquement la matrice B5 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 6.
- double ** [matriceB5_8Dynamique](#) (void)
Permet de créer dynamiquement la matrice B5 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 8.
- double ** [matriceB5_10Dynamique](#) (void)
Permet de créer dynamiquement la matrice B5 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 10.
- double ** [matriceB6_3Dynamique](#) (void)
Permet de créer dynamiquement la matrice B6 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 3.
- double ** [matriceB6_6Dynamique](#) (void)
Permet de créer dynamiquement la matrice B6 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 6.
- double ** [matriceB6_8Dynamique](#) (void)
Permet de créer dynamiquement la matrice B6 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 8.
- double ** [matriceB6_10Dynamique](#) (void)
Permet de créer dynamiquement la matrice B6 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 10.

2.5.1 Function Documentation

2.5.1.1 afficheMatrice()

```
int afficheMatrice (
    double ** tab,
    unsigned int nbrLig,
    unsigned int nbrCol )
```

Permet d'afficher les éléments d'une matrice.

Parameters

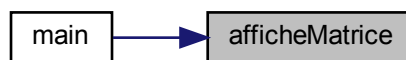
in	<i>tab</i>	Matrice à afficher.
in	<i>nbrLig</i>	Correspond au nombre de ligne de la matrice à afficher.
in	<i>nbrCol</i>	Correspond au nombre de colonne de la matrice à afficher.

Returns

Renvoie 0 si la fonction se déroule normalement jusqu'à la fin.

Definition at line 31 of file matriceTest.c.

Here is the caller graph for this function:

**2.5.1.2 creationMatriceDynamique()**

```
double** creationMatriceDynamique (  
    unsigned int nbrLig,  
    unsigned int nbrCol )
```

Permet de créer une matrice de manière dynamique.

Parameters

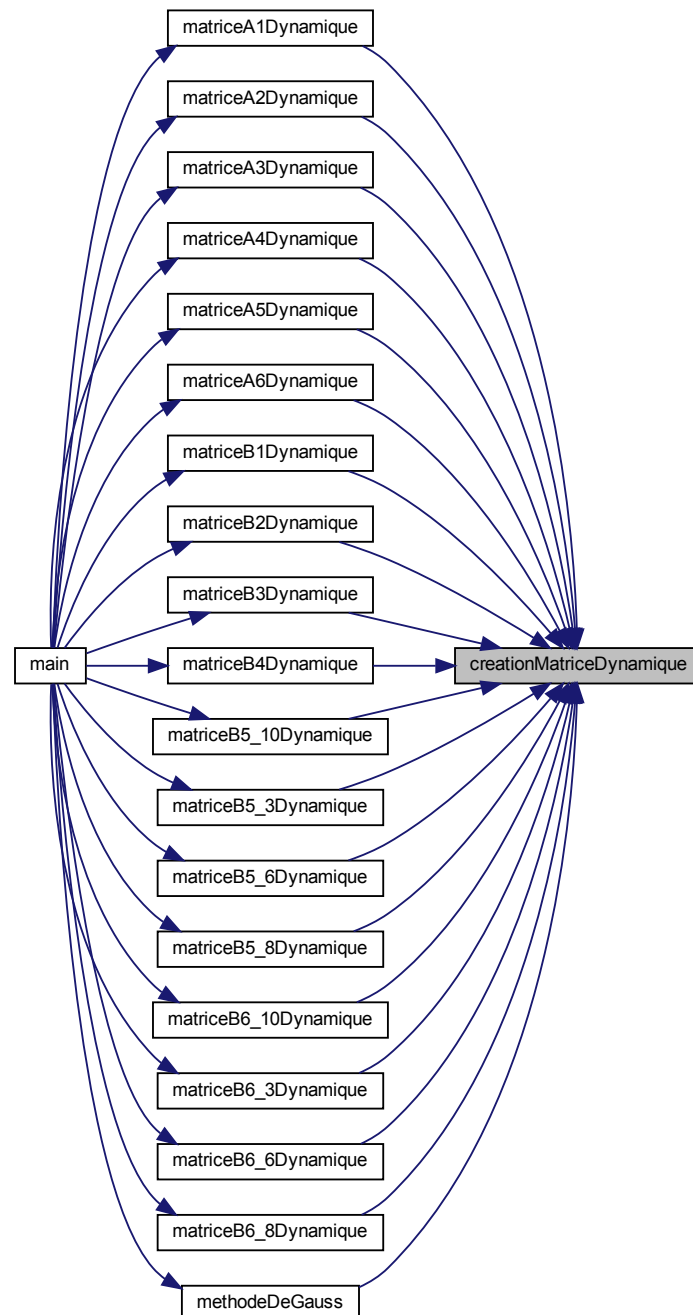
in	<i>nbrLig</i>	Correspond au nombre de ligne que l'on veut donner à la matrice .
in	<i>nbrCol</i>	Correspond au nombre de colonne que l'on veut donner à la matrice.

Returns

Renvoie le pointeur sur la matrice, vaut **NULL** si l'allocation à échouer.

Definition at line 61 of file matriceTest.c.

Here is the caller graph for this function:



2.5.1.3 liberationMemoireMatrice()

```

int liberationMemoireMatrice (
    double ** matrice,
    unsigned int nbrLig )

```


Permet de libérer la mémoire d'une matrice, qui a été créer dynamiquement.

Parameters

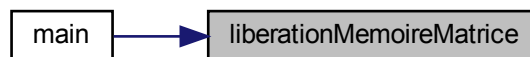
in	<i>matrice</i>	Correspond à la matrice à libérer.
in	<i>nbrLig</i>	Correspond au nombre de ligne de cette matrice.

Returns

Renvoie 0 si la focntion se déroule normalement jusqu'à la fin.

Definition at line 94 of file matriceTest.c.

Here is the caller graph for this function:



2.5.1.4 matriceA1Dynamique()

```
double** matriceA1Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A1 de l'énoncé.

Returns

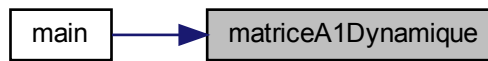
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 131 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.5 matriceA2Dynamique()

```
double** matriceA2Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A2 de l'énoncé.

Returns

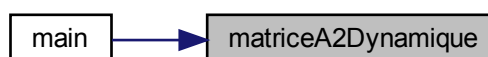
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 176 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.6 matriceA3Dynamique()

```
double** matriceA3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A3 de l'énoncé.

Returns

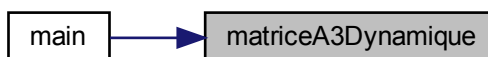
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 219 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.7 matriceA4Dynamique()

```
double** matriceA4Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice A4 de l'énoncé.

Returns

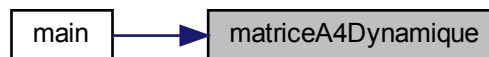
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 263 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**2.5.1.8 matriceA5Dynamique()**

```
double** matriceA5Dynamique (
    unsigned int taille )
```

Permet de créer dynamiquement la matrice A5 de l'énoncé.

Parameters

<code>in</code>	<code><i>taille</i></code>	Correspond à la taille que l'on veut donner à la matrice .
-----------------	----------------------------	--

Returns

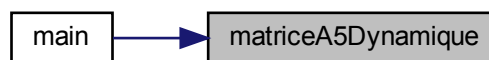
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 309 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.9 matriceA6Dynamique()

```
double** matriceA6Dynamique (
    unsigned int taille )
```

Permet de créer dynamiquement la matrice A6 de l'énoncé.

Parameters

in	<i>taille</i>	Correspond à la taille que l'on veut donner à la matrice .
----	---------------	--

Returns

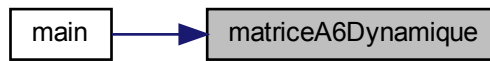
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 432 of file `matriceTest.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.10 `matriceB1Dynamique()`

```
double** matriceB1Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B1 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A1.

Returns

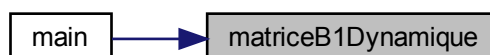
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 159 of file `matriceTest.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.11 matriceB2Dynamique()

```
double** matriceB2Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B2 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A2.

Returns

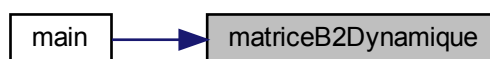
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 201 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.12 matriceB3Dynamique()

```
double** matriceB3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A3.

Returns

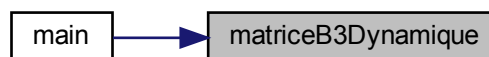
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 245 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**2.5.1.13 matriceB4Dynamique()**

```
double** matriceB4Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B4 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A4.

Returns

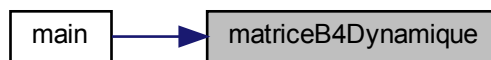
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 288 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.14 matriceB5_10Dynamique()

```
double** matriceB5_10Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 10.

Returns

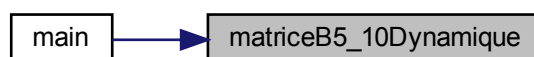
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 404 of file `matriceTest.c`.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.15 matriceB5_3Dynamique()

```
double** matriceB5_3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 3.

Returns

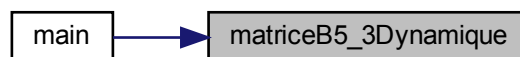
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 342 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.16 matriceB5_6Dynamique()

```
double** matriceB5_6Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 6.

Returns

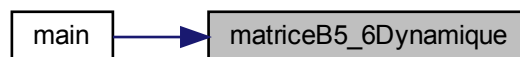
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 360 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.17 matriceB5_8Dynamique()

```
double** matriceB5_8Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B5 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A5 en taille 8.

Returns

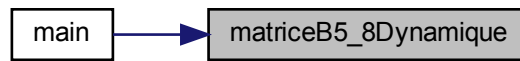
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 381 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.18 matriceB6_10Dynamique()

```
double** matriceB6_10Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 10 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 10.

Returns

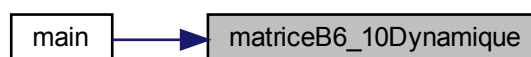
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 527 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.19 matriceB6_3Dynamique()

```
double** matriceB6_3Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 3 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 3.

Returns

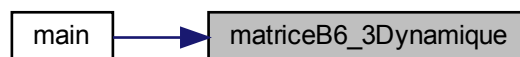
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 465 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



2.5.1.20 matriceB6_6Dynamique()

```
double** matriceB6_6Dynamique (  
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 6 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 6.

Returns

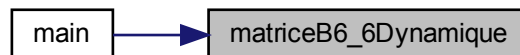
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 483 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:

**2.5.1.21 matriceB6_8Dynamique()**

```
double** matriceB6_8Dynamique (
    void )
```

Permet de créer dynamiquement la matrice B6 en taille 8 de l'énoncé. Obtenu en faisant la somme des lignes de la matrice A6 en taille 8.

Returns

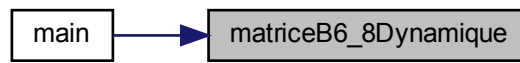
Renvoie le pointeur sur cette matrice, **NULL** si l'allocation à échouer.

Definition at line 504 of file matriceTest.c.

Here is the call graph for this function:



Here is the caller graph for this function:



Index

afficheMatrice
 matriceTest.c, [15](#)
 matriceTest.h, [34](#)

creationMatriceDynamique
 matriceTest.c, [15](#)
 matriceTest.h, [35](#)

gauss.c, [3](#)
 methodeDeGauss, [4](#)
 permutationLigne, [5](#)
 TAILLE_MATRICE, [4](#)
 valeurAbsolue, [6](#)
 VECTEUR_COLONNE, [4](#)

gauss.h, [7](#)
 methodeDeGauss, [8](#)
 permutationLigne, [9](#)
 TAILLE_MATRICE, [7](#)
 valeurAbsolue, [9](#)
 VECTEUR_COLONNE, [7](#)

liberationMemoireMatrice
 matriceTest.c, [18](#)
 matriceTest.h, [36](#)

main
 main.c, [11](#)

main.c, [10](#)
 main, [11](#)

matriceA1Dynamique
 matriceTest.c, [19](#)
 matriceTest.h, [37](#)

matriceA2Dynamique
 matriceTest.c, [20](#)
 matriceTest.h, [38](#)

matriceA3Dynamique
 matriceTest.c, [20](#)
 matriceTest.h, [38](#)

matriceA4Dynamique
 matriceTest.c, [21](#)
 matriceTest.h, [39](#)

matriceA5Dynamique
 matriceTest.c, [22](#)
 matriceTest.h, [40](#)

matriceA6Dynamique
 matriceTest.c, [23](#)
 matriceTest.h, [41](#)

matriceB1Dynamique
 matriceTest.c, [24](#)
 matriceTest.h, [42](#)

matriceB2Dynamique
 matriceTest.c, [24](#)
 matriceTest.h, [42](#)

matriceB3Dynamique
 matriceTest.c, [25](#)
 matriceTest.h, [43](#)

matriceB4Dynamique
 matriceTest.c, [26](#)
 matriceTest.h, [44](#)

matriceB5_10Dynamique
 matriceTest.c, [27](#)
 matriceTest.h, [45](#)

matriceB5_3Dynamique
 matriceTest.c, [27](#)
 matriceTest.h, [45](#)

matriceB5_6Dynamique
 matriceTest.c, [28](#)
 matriceTest.h, [46](#)

matriceB5_8Dynamique
 matriceTest.c, [29](#)
 matriceTest.h, [47](#)

matriceB6_10Dynamique
 matriceTest.c, [30](#)
 matriceTest.h, [48](#)

matriceB6_3Dynamique
 matriceTest.c, [30](#)
 matriceTest.h, [48](#)

matriceB6_6Dynamique
 matriceTest.c, [31](#)
 matriceTest.h, [49](#)

matriceB6_8Dynamique
 matriceTest.c, [32](#)
 matriceTest.h, [50](#)

matriceTest.c, [12](#)
 afficheMatrice, [15](#)
 creationMatriceDynamique, [15](#)
 liberationMemoireMatrice, [18](#)
 matriceA1Dynamique, [19](#)
 matriceA2Dynamique, [20](#)
 matriceA3Dynamique, [20](#)
 matriceA4Dynamique, [21](#)
 matriceA5Dynamique, [22](#)
 matriceA6Dynamique, [23](#)
 matriceB1Dynamique, [24](#)
 matriceB2Dynamique, [24](#)
 matriceB3Dynamique, [25](#)
 matriceB4Dynamique, [26](#)
 matriceB5_10Dynamique, [27](#)
 matriceB5_3Dynamique, [27](#)

- matriceB5_6Dynamique, [28](#)
- matriceB5_8Dynamique, [29](#)
- matriceB6_10Dynamique, [30](#)
- matriceB6_3Dynamique, [30](#)
- matriceB6_6Dynamique, [31](#)
- matriceB6_8Dynamique, [32](#)
- TAILLE_MATRICE, [14](#)
- matriceTest.h, [33](#)
 - afficheMatrice, [34](#)
 - creationMatriceDynamique, [35](#)
 - liberationMemoireMatrice, [36](#)
 - matriceA1Dynamique, [37](#)
 - matriceA2Dynamique, [38](#)
 - matriceA3Dynamique, [38](#)
 - matriceA4Dynamique, [39](#)
 - matriceA5Dynamique, [40](#)
 - matriceA6Dynamique, [41](#)
 - matriceB1Dynamique, [42](#)
 - matriceB2Dynamique, [42](#)
 - matriceB3Dynamique, [43](#)
 - matriceB4Dynamique, [44](#)
 - matriceB5_10Dynamique, [45](#)
 - matriceB5_3Dynamique, [45](#)
 - matriceB5_6Dynamique, [46](#)
 - matriceB5_8Dynamique, [47](#)
 - matriceB6_10Dynamique, [48](#)
 - matriceB6_3Dynamique, [48](#)
 - matriceB6_6Dynamique, [49](#)
 - matriceB6_8Dynamique, [50](#)
- methodeDeGauss
 - gauss.c, [4](#)
 - gauss.h, [8](#)
- permutationLigne
 - gauss.c, [5](#)
 - gauss.h, [9](#)
- TAILLE_MATRICE
 - gauss.c, [4](#)
 - gauss.h, [7](#)
 - matriceTest.c, [14](#)
- valeurAbsolue
 - gauss.c, [6](#)
 - gauss.h, [9](#)
- VECTEUR_COLONNE
 - gauss.c, [4](#)
 - gauss.h, [7](#)