

# Poly'novant Stéganographie

v3

Generated by Doxygen 1.8.11



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	BufferBinary Struct Reference . . . . .	5
3.1.1	Field Documentation . . . . .	5
3.1.1.1	binary . . . . .	5
3.1.1.2	indexNextBit . . . . .	5
3.2	Condition Struct Reference . . . . .	5
3.2.1	Field Documentation . . . . .	6
3.2.1.1	channels . . . . .	6
3.2.1.2	magic_Number . . . . .	6
3.2.1.3	magicNumberSize . . . . .	6
3.2.1.4	nbBits . . . . .	6
3.2.1.5	pattern . . . . .	6
3.3	HiddenFileByte Struct Reference . . . . .	6
3.3.1	Field Documentation . . . . .	6
3.3.1.1	BLOC . . . . .	6
3.3.1.2	byte . . . . .	6
3.3.1.3	indexNextByte . . . . .	6
3.3.1.4	tabSize . . . . .	6

3.4	Img Struct Reference	6
3.4.1	Field Documentation	7
3.4.1.1	bitmapImage	7
3.4.1.2	height	7
3.4.1.3	isColor	7
3.4.1.4	width	7
3.5	OptionList Struct Reference	7
3.5.1	Field Documentation	8
3.5.1.1	channels	8
3.5.1.2	erreur	8
3.5.1.3	finOption	8
3.5.1.4	hasfinOption	8
3.5.1.5	hasInOption	8
3.5.1.6	hasOutOption	8
3.5.1.7	inputFile	8
3.5.1.8	magic_number	8
3.5.1.9	magicNumberSize	8
3.5.1.10	numberBitsUsed	8
3.5.1.11	outputFile	8
3.5.1.12	pattern_used	8

<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/image.c File Reference . . . . .	9
4.1.1	Function Documentation . . . . .	9
4.1.1.1	extractByteHiddenFileImage(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	9
4.1.1.2	getGoodFormat(char *filename) . . . . .	10
4.1.1.3	isColorImage(FREE_IMAGE_FORMAT format) . . . . .	10
4.1.1.4	loadImage(char *filename, char *finOption, struct Img *image, char *channels) .	10
4.1.1.5	verifyChannelAndImageFormat(FREE_IMAGE_FORMAT format, char *channels)	11
4.2	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/image.h File Reference . . . . .	11
4.2.1	Function Documentation . . . . .	11
4.2.1.1	extractByteHiddenFileImage(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	11
4.2.1.2	loadImage(char *filename, char *finOption, struct Img *image, char *channels) .	12
4.3	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/main.c File Reference . . . . .	12
4.3.1	Function Documentation . . . . .	13
4.3.1.1	main(int argc, char *argv[]) . . . . .	13
4.4	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/mathlib.c File Reference . . . . .	13
4.4.1	Function Documentation . . . . .	13
4.4.1.1	power(int nombre, int pow) . . . . .	13
4.5	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/mathlib.h File Reference . . . . .	13
4.5.1	Function Documentation . . . . .	13
4.5.1.1	power(int nombre, int pow) . . . . .	13
4.6	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/options↵ Parser.c File Reference . . . . .	14
4.6.1	Function Documentation . . . . .	14
4.6.1.1	isCorrectBitsUsedOption(int bitsUsed) . . . . .	14
4.6.1.2	isCorrectChannel(char *channel) . . . . .	14
4.6.1.3	isCorrectFinOption(char *Fin) . . . . .	14

4.6.1.4	<code>isInFileExisting(char *filename)</code>	14
4.6.1.5	<code>parseOption(char *argv[])</code>	14
4.7	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/options← Parser.h File Reference	15
4.7.1	Function Documentation	15
4.7.1.1	<code>isCorrectBitsUsedOption(int bitsUsed)</code>	15
4.7.1.2	<code>isCorrectFinOption(char *Fin)</code>	15
4.7.1.3	<code>isInFileExisting(char *filename)</code>	15
4.7.1.4	<code>parseOption(char *argv[])</code>	15
4.8	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/output.c File Reference	15
4.8.1	Function Documentation	16
4.8.1.1	<code>createHideFile(char *outputFileName, struct HiddenFileByte *tabByte)</code>	16
4.8.1.2	<code>printOnStandardOutput(struct HiddenFileByte *tabByte)</code>	16
4.9	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/output.h File Reference	16
4.9.1	Function Documentation	17
4.9.1.1	<code>createHideFile(char *outputFileName, struct HiddenFileByte *tabByte)</code>	17
4.9.1.2	<code>printOnStandardOutput(struct HiddenFileByte *tabByte)</code>	17
4.10	D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/pattern.c File Reference	17
4.10.1	Function Documentation	18
4.10.1.1	<code>colorImageCase(struct Img *image, struct BufferBinary *bufferBinary, struct HiddenFileByte *tabByte, int coordX, int coordY, struct Condition *condition, char channel)</code>	18
4.10.1.2	<code>direct(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition)</code>	18
4.10.1.3	<code>external_spiral(struct Img *image, struct HiddenFileByte *tabByte, struct Condi- tion *condition)</code>	18
4.10.1.4	<code>grayImageCase(struct Img *image, struct BufferBinary *bufferBinary, struct HiddenFileByte *tabByte, int coordX, int coordY, struct Condition *condition)</code>	19
4.10.1.5	<code>internal_spiral(struct Img *image, struct HiddenFileByte *tabByte, struct Condi- tion *condition)</code>	19
4.10.1.6	<code>reverse(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition)</code>	19

4.11 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/pattern.h File Reference . . . . .	20
4.11.1 Function Documentation . . . . .	20
4.11.1.1 colorImageCase(struct Img *image, struct BufferBinary *bufferBinary, struct HiddenFileByte *tabByte, int coordX, int coordY, struct Condition *condition, char channel) . . . . .	20
4.11.1.2 direct(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	21
4.11.1.3 external_spiral(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	21
4.11.1.4 grayImageCase(struct Img *image, struct BufferBinary *bufferBinary, struct HiddenFileByte *tabByte, int coordX, int coordY, struct Condition *condition) . . . . .	21
4.11.1.5 internal_spiral(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	22
4.11.1.6 reverse(struct Img *image, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	22
4.12 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/tools.c File Reference . . . . .	22
4.12.1 Function Documentation . . . . .	23
4.12.1.1 addBinaryBufferInTab(unsigned int n, struct BufferBinary *bufferBinary, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	23
4.12.1.2 binaryToByteTab(struct HiddenFileByte *tabByte, struct BufferBinary *bufferBinary, struct Condition *condition) . . . . .	23
4.12.1.3 verifyMagicNumber(struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	23
4.13 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/tools.h File Reference . . . . .	24
4.13.1 Typedef Documentation . . . . .	24
4.13.1.1 Byte . . . . .	24
4.13.2 Function Documentation . . . . .	24
4.13.2.1 addBinaryBufferInTab(unsigned int n, struct BufferBinary *bufferBinary, struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	24
4.13.2.2 binaryToByteTab(struct HiddenFileByte *tabByte, struct BufferBinary *bufferBinary, struct Condition *condition) . . . . .	25
4.13.2.3 verifyMagicNumber(struct HiddenFileByte *tabByte, struct Condition *condition) . . . . .	25





# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">BufferBinary</a>	5
<a href="#">Condition</a>	5
<a href="#">HiddenFileByte</a>	6
<a href="#">Img</a>	6
<a href="#">OptionList</a>	7



## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">image.c</a>	9
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">image.h</a>	11
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">main.c</a>	12
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">mathlib.c</a>	13
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">mathlib.h</a>	13
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">optionsParser.c</a>	14
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">optionsParser.h</a>	15
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">output.c</a>	15
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">output.h</a>	16
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">pattern.c</a>	17
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">pattern.h</a>	20
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">tools.c</a>	22
D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/ <a href="#">tools.h</a>	24



## Chapter 3

# Data Structure Documentation

### 3.1 BufferBinary Struct Reference

```
#include <tools.h>
```

#### Data Fields

- char [binary](#) [8]
- int [indexNextBit](#)

#### 3.1.1 Field Documentation

3.1.1.1 char [binary](#)[8]

3.1.1.2 int [indexNextBit](#)

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

- D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/[tools.h](#)

### 3.2 Condition Struct Reference

```
#include <image.h>
```

#### Data Fields

- int [nbBits](#)
- char \* [channels](#)
- char \* [magic\\_Number](#)
- int [magicNumberSize](#)
- char \* [pattern](#)

### 3.2.1 Field Documentation

3.2.1.1 char\* channels

3.2.1.2 char\* magic\_Number

3.2.1.3 int magicNumberSize

3.2.1.4 int nbBits

3.2.1.5 char\* pattern

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

- D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/[image.h](#)

## 3.3 HiddenFileByte Struct Reference

```
#include <tools.h>
```

### Data Fields

- char \* [byte](#)
- unsigned long [indexNextByte](#)
- int [BLOC](#)
- unsigned long [tabSize](#)

### 3.3.1 Field Documentation

3.3.1.1 int BLOC

3.3.1.2 char\* byte

3.3.1.3 unsigned long indexNextByte

3.3.1.4 unsigned long tabSize

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

- D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/[tools.h](#)

## 3.4 Img Struct Reference

```
#include <image.h>
```

## Data Fields

- FIBITMAP \* [bitmapImage](#)
- int [height](#)
- int [width](#)
- int [isColor](#)

### 3.4.1 Field Documentation

#### 3.4.1.1 FIBITMAP\* [bitmapImage](#)

#### 3.4.1.2 int [height](#)

#### 3.4.1.3 int [isColor](#)

#### 3.4.1.4 int [width](#)

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

- D:/Users/Kevin/Documents/Cours/Polytech S13/Polytech S5/Projet S5/private/reveal/src/[image.h](#)

## 3.5 OptionList Struct Reference

```
#include <optionsParser.h>
```

## Data Fields

- char \* [inputFile](#)
- int [hasInOption](#)
- char \* [finOption](#)
- int [hasfinOption](#)
- char \* [outputFile](#)
- int [hasOutOption](#)
- int [numberBitsUsed](#)
- char \* [channels](#)
- char \* [pattern\\_used](#)
- char \* [magic\\_number](#)
- int [magicNumberSize](#)
- int [erreur](#)

### 3.5.1 Field Documentation

3.5.1.1 `char* channels`

3.5.1.2 `int erreur`

3.5.1.3 `char* finOption`

3.5.1.4 `int hasfinOption`

3.5.1.5 `int hasInOption`

3.5.1.6 `int hasOutOption`

3.5.1.7 `char* inputFile`

3.5.1.8 `char* magic_number`

3.5.1.9 `int magicNumberSize`

3.5.1.10 `int numberBitsUsed`

3.5.1.11 `char* outputFile`

3.5.1.12 `char* pattern_used`

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

- `D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/optionsParser.h`



## Chapter 4

# File Documentation

### 4.1 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/image.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "image.h"
#include "tools.h"
#include "FreeImage.h"
```

#### Functions

- int [isColorImage](#) (FREE\_IMAGE\_FORMAT format)  
*Verifie si l'image est une image en couleur (avec RGB) ou non.*
- int [verifyChannelAndImageFormat](#) (FREE\_IMAGE\_FORMAT format, char \*channels)  
*Fonction qui verifie si il y a une cohérence entre le format de l'image et les channels (ici Y correspond à Gray)*
- FREE\_IMAGE\_FORMAT [getGoodFormat](#) (char \*filename)  
*Si l'utilisateur n'a pas spécifier un format à l'aide de l'option -Fin, on choisit ici le bon format(parmi les 4 disponibles) correspondant à l'image.*
- int [loadImage](#) (char \*filename, char \*finOption, struct [Img](#) \*image, char \*channels)  
*Chargement d'une image.*
- int [extractByteHiddenFileImage](#) (struct [Img](#) \*image, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)  
*Redirection vers le bon pattern de revelation.*

#### 4.1.1 Function Documentation

4.1.1.1 int [extractByteHiddenFileImage](#) ( struct [Img](#) \* *image*, struct [HiddenFileByte](#) \* *tabByte*, struct [Condition](#) \* *condition* )

Redirection vers le bon pattern de revelation.

## Parameters

<i>IMG</i>	Structure avec les informations de l'image (bitmap, hauteur, largeur)
<i>HIDDENFILEBYTE</i>	Structure représentant le tableau de byte du fichier caché
<i>Condition</i>	<i>Condition</i> de revelation du fichier (nombre de bits sur lequel est codé le fichier, pattern, channels,...)

4.1.1.2 **FREE\_IMAGE\_FORMAT** `getGoodFormat ( char * filename )`

Si l'utilisateur n'a pas spécifier un format à l'aide de l'option -Fin, on choisit ici le bon format(parmi les 4 disponibles) correspondant à l'image.

## Parameters

<i>filename</i>	Image traiter
-----------------	---------------

## Returns

Retourne NULL si l'image correspond à aucun des 4 formats disponible, sinon retourne le bon format

4.1.1.3 **int** `isColorImage ( FREE_IMAGE_FORMAT format )`

Verifie si l'image est une image en couleur (avec RGB) ou non.

## Parameters

<i>format</i>	format de l'image
---------------	-------------------

## Returns

Retourne 1 si c'est une image couleur, 0 sinon

4.1.1.4 **int** `loadImage ( char * filename, char * finOption, struct Img * image, char * channels )`

Chargement d'une image.

Cette methode a pour but de charger une image grâce à son nom, en utilisant le librairie FreeImage On remplit une struct avec le bitmap de l'image ainsi que sa hauteur et largeur.

## Parameters

<i>filename</i>	Chemin absolu ou relatif de l'image à charger
<i>IMG</i>	[description] Struct dans laquelle remplir les informations concernant l'image

### Returns

Si la fonction renvoie 0, alors une erreur s'est produite. Sinon, la fonction renvoie 1

#### 4.1.1.5 int verifyChannelAndImageFormat ( FREE\_IMAGE\_FORMAT *format*, char \* *channels* )

Fonction qui verifie si il y a une cohérence entre le format de l'image et les channels (ici Y correspond à Gray)

### Parameters

<i>format</i>	Format de l'image
<i>channels</i>	Channel sur lesquels decoder

### Returns

Retourne 0 si erreur, 1 sinon

## 4.2 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/image.h File Reference

```
#include "FreeImage.h"
#include "pattern.h"
#include "tools.h"
```

### Data Structures

- struct [Img](#)
- struct [Condition](#)

### Functions

- int [loadImage](#) (char \*filename, char \*finOption, struct [Img](#) \*image, char \*channels)  
*Chargement d'une image.*
- int [extractByteHiddenFileImage](#) (struct [Img](#) \*image, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)  
*Redirection vers le bon pattern de revelation.*

#### 4.2.1 Function Documentation

##### 4.2.1.1 int extractByteHiddenFileImage ( struct [Img](#) \* *image*, struct [HiddenFileByte](#) \* *tabByte*, struct [Condition](#) \* *condition* )

Redirection vers le bon pattern de revelation.

## Parameters

<i>IMG</i>	Structure avec les informations de l'image (bitmap, hauteur, largeur)
<i>HIDDENFILEBYTE</i>	Structure représentant le tableau de byte du fichier caché
<i>Condition</i>	<i>Condition</i> de revelation du fichier (nombre de bits sur lequel est codé le fichier, pattern, channels,...)

4.2.1.2 `int loadImage ( char * filename, char * finOption, struct Img * image, char * channels )`

Chargement d'une image.

Cette methode a pour but de charger une image grâce à son nom, en utilisant le librairie FreeImage On remplit une struct avec le bitmap de l'image ainsi que sa hauteur et largeur.

## Parameters

<i>filename</i>	Chemin absolu ou relatif de l'image à charger
<i>IMG</i>	[description] Struct dans laquelle remplir les informations concernant l'image

## Returns

Si la fonction renvoie 0, alors une erreur s'est produite. Sinon, la fonction renvoie 1

## 4.3 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/main.c

### File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "FreeImage.h"
#include "mathlib.h"
#include "optionsParser.h"
#include "tools.h"
#include "image.h"
#include "output.h"
#include "pattern.h"
```

## Functions

- `int main (int argc, char *argv[ ])`

#### 4.3.1 Function Documentation

4.3.1.1 `int main ( int argc, char * argv [ ] )`

### 4.4 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/mathlib.c File Reference

```
#include "mathlib.h"
```

#### Functions

- `int power (int nombre, int pow)`  
*Calcul d'une puissance.*

#### 4.4.1 Function Documentation

4.4.1.1 `int power ( int nombre, int pow )`

Calcul d'une puissance.

##### Parameters

<i>nombre</i>	Nombre sur lequel on souhaite calculer la puissance
<i>pow</i>	Puissance a laquelle on souhaite elevé le nombre

##### Returns

Resultat du calcul

### 4.5 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/mathlib.h File Reference

#### Functions

- `int power (int nombre, int pow)`  
*Calcul d'une puissance.*

#### 4.5.1 Function Documentation

4.5.1.1 `int power ( int nombre, int pow )`

Calcul d'une puissance.

## Parameters

<i>nombre</i>	Nombre sur lequel on souhaite calculer la puissance
<i>pow</i>	Puissance a laquelle on souhaite elevé le nombre

## Returns

Resultat du calcul

## 4.6 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/optionsParser.c File Reference ↩

```
#include "optionsParser.h"
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

## Functions

- int [isInFileExisting](#) (char \*filename)
- int [isCorrectBitsUsedOption](#) (int bitsUsed)
- int [isCorrectFinOption](#) (char \*Fin)
- int [isCorrectChannel](#) (char \*channel)
- struct [OptionList](#) [parseOption](#) (char \*argv[])

*Parseur d'options entré en ligne de commande.*

### 4.6.1 Function Documentation

4.6.1.1 int [isCorrectBitsUsedOption](#) ( int *bitsUsed* )

4.6.1.2 int [isCorrectChannel](#) ( char \* *channel* )

4.6.1.3 int [isCorrectFinOption](#) ( char \* *Fin* )

4.6.1.4 int [isInFileExisting](#) ( char \* *filename* )

4.6.1.5 struct [OptionList](#) [parseOption](#) ( char \* *argv*[] )

Parseur d'options entré en ligne de commande.

Cette fonction analyse la ligne de commande entré par l'utilisateur et associe à chaque options la valeur entrée.

## Parameters

<i>argv</i>	Ligne de commande de l'utilisateur
-------------	------------------------------------

## Returns

Struct OPTIONLIST, contenant toutes les informations de la ligne de commande

## 4.7 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/options Parser.h File Reference

## Data Structures

- struct [OptionList](#)

## Functions

- int [isInFileExisting](#) (char \*filename)
- int [isCorrectBitsUsedOption](#) (int bitsUsed)
- int [isCorrectFinOption](#) (char \*Fin)
- struct [OptionList](#) [parseOption](#) (char \*argv[])

*Parseur d'options entré en ligne de commande.*

### 4.7.1 Function Documentation

4.7.1.1 int [isCorrectBitsUsedOption](#) ( int *bitsUsed* )

4.7.1.2 int [isCorrectFinOption](#) ( char \* *Fin* )

4.7.1.3 int [isInFileExisting](#) ( char \* *filename* )

4.7.1.4 struct [OptionList](#) [parseOption](#) ( char \* *argv*[] )

Parseur d'options entré en ligne de commande.

Cette fonction analyse la ligne de commande entré par l'utilisateur et associe à chaque options la valeur entrée.

## Parameters

<i>argv</i>	Ligne de commande de l'utilisateur
-------------	------------------------------------

## Returns

Struct OPTIONLIST, contenant toutes les informations de la ligne de commande

## 4.8 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/output.c File Reference

```
#include "tools.h"
```

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

## Functions

- void `printOnStandardOutput` (struct `HiddenFileByte` \*tabByte)  
*Ecriture sur la sortie standard.*
- void `createHideFile` (char \*outputFileName, struct `HiddenFileByte` \*tabByte)  
*Ecriture dans un nouveau fichier.*

### 4.8.1 Function Documentation

#### 4.8.1.1 void createHideFile ( char \* outputFileName, struct HiddenFileByte \* tabByte )

Ecriture dans un nouveau fichier.

Si l'utilisateur a spécifié l'option -out, alors on écrit le fichier caché dans un nouveau fichier où l'utilisateur a spécifié le nom

##### Parameters

<code>outputFileName</code>	Nom du fichier sur lequel écrire
<code>HIDDENFILEBYTE</code>	Tableau de byte à écrire dans le nouveau fichier

#### 4.8.1.2 void printOnStandardOutput ( struct HiddenFileByte \* tabByte )

Ecriture sur la sortie standard.

Si l'utilisateur n'a pas spécifié l'option -out, alors cette fonction est appelée, écrivant le fichier caché sur la sortie standard.

##### Parameters

<code>HIDDENFILEBYTE</code>	Tableau de byte à écrire sur stdout.
-----------------------------	--------------------------------------

## 4.9 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/output.h File Reference

### Functions

- void `printOnStandardOutput` (struct `HiddenFileByte` \*tabByte)  
*Ecriture sur la sortie standard.*
- void `createHideFile` (char \*outputFileName, struct `HiddenFileByte` \*tabByte)  
*Ecriture dans un nouveau fichier.*



## 4.9.1 Function Documentation

### 4.9.1.1 void createHideFile ( char \* *outputFileName*, struct HiddenFileByte \* *tabByte* )

Ecriture dans un nouveau fichier.

Si l'utilisateur a spécifié l'option -out, alors on écrit le fichier caché dans un nouveau fichier où l'utilisateur a spécifié le nom

#### Parameters

<i>outputFileName</i>	Nom du fichier sur lequel écrire
<i>HIDDENFILEBYTE</i>	Tableau de byte à écrire dans le nouveau fichier

### 4.9.1.2 void printOnStandardOutput ( struct HiddenFileByte \* *tabByte* )

Ecriture sur la sortie standard.

Si l'utilisateur n'a pas spécifié l'option -out, alors cette fonction est appelée, écrivant le fichier caché sur la sortie standard.

#### Parameters

<i>HIDDENFILEBYTE</i>	Tableau de byte à écrire sur stdout.
-----------------------	--------------------------------------

## 4.10 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/pattern.c File Reference

```
#include "pattern.h"
#include "image.h"
#include "tools.h"
```

## Functions

- int [colorImageCase](#) (struct [Img](#) \*image, struct [BufferBinary](#) \*bufferBinary, struct [HiddenFileByte](#) \*tabByte, int coordX, int coordY, struct [Condition](#) \*condition, char channel)  
*Recuperer les informations d'un pixels pour une image en couleur.*
- int [grayImageCase](#) (struct [Img](#) \*image, struct [BufferBinary](#) \*bufferBinary, struct [HiddenFileByte](#) \*tabByte, int coordX, int coordY, struct [Condition](#) \*condition)  
*Recuperer l'information d'un pixel sur une image en niveau de gris.*
- int [direct](#) (struct [Img](#) \*image, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)  
*Pattern de revelation direct.*
- int [reverse](#) (struct [Img](#) \*image, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)  
*Pattern de revelation reverse (inverse du direct)*
- int [external\\_spiral](#) (struct [Img](#) \*image, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)

*Pattern de revaltion external\_spiral. On part en haut à gauche de l'image puis on tourne en spirale dans le sens des aiguilles d'une montre.*

- int [internal\\_spiral](#) (struct [Img](#) \*image, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)

*Pattern de revelation internal\_spiral. Pour cette spirale, on commence du centre de l'image.*

#### 4.10.1 Function Documentation

4.10.1.1 int colorImageCase ( struct [Img](#) \* *image*, struct [BufferBinary](#) \* *bufferBinary*, struct [HiddenFileByte](#) \* *tabByte*, int *coordX*, int *coordY*, struct [Condition](#) \* *condition*, char *channel* )

Recuperer les informations d'un pixels pour une image en couleur.

Cette fonction permet de recuperer l'informations d'un pixel (grâce à ces coordonnées) sur un channel précis

##### Parameters

<i>IMG</i>	Image sur lequel on recupère les informations
<i>BUFFERBINARY</i>	Tableau de binaire
<i>HIDDENFILEBYTE</i>	Tableau de byte, correspondant au fichier caché dans l'image
<i>coordX</i>	Coordonnée sur l'axe des X du pixel
<i>coordY</i>	Coordonnée sur l'axe des Y du pixel
<a href="#">Condition</a>	<a href="#">Condition</a> de révélation du message (nombre de bits, magic_number,...)
<i>channel</i>	Channel sur lequel lire l'information

##### Returns

Return 1 si le magic\_number a été trouvé, 0 sinon

4.10.1.2 int direct ( struct [Img](#) \* *image*, struct [HiddenFileByte](#) \* *tabByte*, struct [Condition](#) \* *condition* )

Pattern de revelation direct.

##### Parameters

<i>IMG</i>	Image sur laquelle on recupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<a href="#">Condition</a>	<a href="#">Condition</a> de révélation

##### Returns

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

4.10.1.3 int external\_spiral ( struct [Img](#) \* *image*, struct [HiddenFileByte](#) \* *tabByte*, struct [Condition](#) \* *condition* )

Pattern de revaltion external\_spiral. On part en haut à gauche de l'image puis on tourne en spirale dans le sens des aiguilles d'une montre.

## Parameters

<i>IMG</i>	Image sur laquelle on récupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

## Returns

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

4.10.1.4 `int grayImageCase ( struct Img * image, struct BufferBinary * bufferBinary, struct HiddenFileByte * tabByte, int coordX, int coordY, struct Condition * condition )`

Recuperer l'information d'un pixel sur une image en niveau de gris.

## Parameters

<i>IMG</i>	Image sur lequel on récupère les informations
<i>BUFFERBINARY</i>	Tableau de binaire
<i>HIDDENFILEBYTE</i>	Tableau de byte, correspondant au fichier caché dans l'image
<i>coordX</i>	Coordonnée sur l'axe des X du pixel
<i>coordY</i>	Coordonnée sur l'axe des Y du pixel
<i>Condition</i>	<i>Condition</i> de révélation du message (nombre de bits, magic_number,...)

## Returns

Return 1 si le magic\_number a été trouvé, 0 sinon

4.10.1.5 `int internal_spiral ( struct Img * image, struct HiddenFileByte * tabByte, struct Condition * condition )`

Pattern de revelation internal\_spiral. Pour cette spirale, on commence du centre de l'image.

## Parameters

<i>IMG</i>	Image sur laquelle on récupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

## Returns

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

4.10.1.6 `int reverse ( struct Img * image, struct HiddenFileByte * tabByte, struct Condition * condition )`

Pattern de revelation reverse (inverse du direct)

## Parameters

<i>IMG</i>	Image sur laquelle on recupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

## Returns

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

## 4.11 D:/Users/Kevin/Documents/Cours/Polytech S13/Polytech S5/Projet S5/private/reveal/src/pattern.h File Reference

```
#include "image.h"
```

## Functions

- int **colorImageCase** (struct **Img** \*image, struct **BufferBinary** \*bufferBinary, struct **HiddenFileByte** \*tabByte, int coordX, int coordY, struct **Condition** \*condition, char channel)  
*Recuperer les informations d'un pixels pour une image en couleur.*
- int **grayImageCase** (struct **Img** \*image, struct **BufferBinary** \*bufferBinary, struct **HiddenFileByte** \*tabByte, int coordX, int coordY, struct **Condition** \*condition)  
*Recuperer l'information d'un pixel sur une image en niveau de gris.*
- int **direct** (struct **Img** \*image, struct **HiddenFileByte** \*tabByte, struct **Condition** \*condition)  
*Pattern de revelation direct.*
- int **reverse** (struct **Img** \*image, struct **HiddenFileByte** \*tabByte, struct **Condition** \*condition)  
*Pattern de revelation reverse (inverse du direct)*
- int **external\_spiral** (struct **Img** \*image, struct **HiddenFileByte** \*tabByte, struct **Condition** \*condition)  
*Pattern de revaltion external\_spiral. On part en haut à gauche de l'image puis on tourne en spirale dans le sens des aiguilles d'une montre.*
- int **internal\_spiral** (struct **Img** \*image, struct **HiddenFileByte** \*tabByte, struct **Condition** \*condition)  
*Pattern de revelation internal\_spiral. Pour cette spirale, on commence du centre de l'image.*

### 4.11.1 Function Documentation

4.11.1.1 int **colorImageCase** ( struct **Img** \* *image*, struct **BufferBinary** \* *bufferBinary*, struct **HiddenFileByte** \* *tabByte*, int *coordX*, int *coordY*, struct **Condition** \* *condition*, char *channel* )

Recuperer les informations d'un pixels pour une image en couleur.

Cette fonction permet de recuperer l'informations d'un pixel (grâce à ces coordonnées) sur un channel précis

## Parameters

<i>IMG</i>	Image sur lequel on recupère les informations
<i>BUFFERBINARY</i>	Tableau de binaire
<i>HIDDENFILEBYTE</i>	Tableau de byte, correspondant au fichier caché dans l'image
<i>coordX</i>	Coordonnée sur l'axe des X du pixel
<i>coordY</i>	Coordonnée sur l'axe des Y du pixel
<i>Condition</i>	<i>Condition</i> de révélation du message (nombre de bits, magic_number,...)
<i>channel</i>	Channel sur lequel lire l'information

#### Returns

Return 1 si le magic\_number a été trouvé, 0 sinon

##### 4.11.1.2 int direct ( struct Img \* *image*, struct HiddenFileByte \* *tabByte*, struct Condition \* *condition* )

Pattern de revelation direct.

#### Parameters

<i>IMG</i>	Image sur laquelle on recupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

#### Returns

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

##### 4.11.1.3 int external\_spiral ( struct Img \* *image*, struct HiddenFileByte \* *tabByte*, struct Condition \* *condition* )

Pattern de revaltion external\_spiral. On part en haut à gauche de l'image puis on tourne en spirale dans le sens des aiguilles d'une montre.

#### Parameters

<i>IMG</i>	Image sur laquelle on recupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

#### Returns

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

##### 4.11.1.4 int grayImageCase ( struct Img \* *image*, struct BufferBinary \* *bufferBinary*, struct HiddenFileByte \* *tabByte*, int *coordX*, int *coordY*, struct Condition \* *condition* )

Recuperer l'information d'un pixel sur une image en niveau de gris.

#### Parameters

<i>IMG</i>	Image sur lequel on recupère les informations
<i>BUFFERBINARY</i>	Tableau de binaire
<i>HIDDENFILEBYTE</i>	Tableau de byte, correspondant au fichier caché dans l'image
<i>coordX</i>	Coordonnée sur l'axe des X du pixel
<i>coordY</i>	Coordonnée sur l'axe des Y du pixel
<i>Condition</i>	<i>Condition</i> de révélation du message (nombre de bits, magic_number,...)

**Returns**

Return 1 si le magic\_number a été trouvé, 0 sinon

4.11.1.5 `int internal_spiral ( struct Img * image, struct HiddenFileByte * tabByte, struct Condition * condition )`

Pattern de revelation internal\_spiral. Pour cette spirale, on commence du centre de l'image.

**Parameters**

<i>IMG</i>	Image sur laquelle on récupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

**Returns**

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

4.11.1.6 `int reverse ( struct Img * image, struct HiddenFileByte * tabByte, struct Condition * condition )`

Pattern de revelation reverse (inverse du direct)

**Parameters**

<i>IMG</i>	Image sur laquelle on récupère l'information
<i>HIDDENFILEBYTE</i>	Tableau de byte du message caché
<i>Condition</i>	<i>Condition</i> de révélation

**Returns**

Return 1 si le magic\_number a été trouvé, 0 sinon (erreur)

## 4.12 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/tools.c

### File Reference

```
#include "tools.h"
#include "mathlib.h"
```

**Functions**

- `int addBinaryBufferInTab` (unsigned int n, struct *BufferBinary* \*bufferBinary, struct *HiddenFileByte* \*tabByte, struct *Condition* \*condition)  
*Transforme un decimal en binaire et l'ajoute au Binary Buffer.*
- `int binaryToByteTab` (struct *HiddenFileByte* \*tabByte, struct *BufferBinary* \*bufferBinary, struct *Condition* \*condition)  
*Tableau binaire vers tableau de byte.*
- `int verifyMagicNumber` (struct *HiddenFileByte* \*tabByte, struct *Condition* \*condition)  
*Verifie si les derniers byte sont le magic number.*

### 4.12.1 Function Documentation

4.12.1.1 `int addBinaryBufferInTab ( unsigned int n, struct BufferBinary * bufferBinary, struct HiddenFileByte * tabByte, struct Condition * condition )`

Transforme un decimal en binaire et l'ajoute au Binary Buffer.

Cette fonction prend en paramètre un nombre decimal (valeur RGBA ou Gray d'un pixel), extrait le binaire du fichier caché (en fonction du nombre de bits sur lequel celui ci est codé) et l'ajoute au tableau de bits(struct bufferBinary)

#### Parameters

<i>int</i>	Valeur RGBA / Gray du pixel
<i>BUFFERBINARY</i>	Tableau de bits
<i>HIDDENFILEBYTE</i>	Tableau de byte du fichier caché
<i>Condition</i>	Conditions de revelation (pattern, channels,...)

#### Returns

Retourne 0 si on vient de finir de lire le magic\_number, 0 sinon.

4.12.1.2 `int binaryToByteTab ( struct HiddenFileByte * tabByte, struct BufferBinary * bufferBinary, struct Condition * condition )`

Tableau binaire vers tableau de byte.

Cette fonction est appelé lorsque le tableau de bits est plein. On calcule ainsi la valeur en decimal du nombre binaire et on l'ajoute au tableau de byte du fichier caché.

#### Parameters

<i>HIDDENFILEBYTE</i>	Tableau de byte du fichier caché
<i>BUFFERBINARY</i>	Tableau de bits
<i>Condition</i>	<i>Condition</i> de revelation (on s'en sert ici surtout pour le magic number)

#### Returns

Retourne 1 si le magic number a été decouvert, 0 sinon.

4.12.1.3 `int verifyMagicNumber ( struct HiddenFileByte * tabByte, struct Condition * condition )`

Verifie si les derniers byte sont le magic number.

Cette fonction verifie les "magicNumberSize" dernier byte du tableau de byte.

#### Parameters

<i>HIDDENFILEBYTE</i>	Tableau de byte
<i>Condition</i>	<i>Condition</i> de revelation (ici utilisé pour les informations sur le magic number)

## Returns

Retourne 1 si le magic\_number est trouvé, 0 sinon

## 4.13 D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech S5/Projet S5/private/reveal/src/tools.h File Reference

```
#include "mathlib.h"
#include "image.h"
```

## Data Structures

- struct [BufferBinary](#)
- struct [HiddenFileByte](#)

## Typedefs

- typedef unsigned char [Byte](#)

## Functions

- int [addBinaryBufferInTab](#) (unsigned int n, struct [BufferBinary](#) \*bufferBinary, struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)  
*Transforme un decimal en binaire et l'ajoute au Binary Buffer.*
- int [binaryToByteTab](#) (struct [HiddenFileByte](#) \*tabByte, struct [BufferBinary](#) \*bufferBinary, struct [Condition](#) \*condition)  
*Tableau binaire vers tableau de byte.*
- int [verifyMagicNumber](#) (struct [HiddenFileByte](#) \*tabByte, struct [Condition](#) \*condition)  
*Verifie si les derniers byte sont le magic number.*

### 4.13.1 Typedef Documentation

#### 4.13.1.1 typedef unsigned char Byte

### 4.13.2 Function Documentation

#### 4.13.2.1 int addBinaryBufferInTab ( unsigned int n, struct BufferBinary \* bufferBinary, struct HiddenFileByte \* tabByte, struct Condition \* condition )

Transforme un decimal en binaire et l'ajoute au Binary Buffer.

Cette fonction prend en paramètre un nombre decimal (valeur RGBA ou Gray d'un pixel), extrait le binaire du fichier caché (en fonction du nombre de bits sur lequel celui ci est codé) et l'ajoute au tableau de bits(struct bufferBinary)



## Parameters

<i>int</i>	Valeur RGBA / Gray du pixel
<i>BUFFERBINARY</i>	Tableau de bits
<i>HIDDENFILEBYTE</i>	Tableau de byte du fichier caché
<i>Condition</i>	Conditions de revelation (pattern, channels,...)

## Returns

Retourne 0 si on vient de finir de lire le magic\_number, 0 sinon.

#### 4.13.2.2 int binaryToByteTab ( struct HiddenFileByte \* *tabByte*, struct BufferBinary \* *bufferBinary*, struct Condition \* *condition* )

Tableau binaire vers tableau de byte.

Cette fonction est appelé lorsque le tableau de bits est plein. On calcule ainsi la valeur en decimal du nombre binaire et on l'ajoute au tableau de byte du fichier caché.

## Parameters

<i>HIDDENFILEBYTE</i>	Tableau de byte du fichier caché
<i>BUFFERBINARY</i>	Tableau de bits
<i>Condition</i>	<i>Condition</i> de revelation (on s'en sert ici surtout pour le magic number)

## Returns

Retourne 1 si le magic number a été decouvert, 0 sinon.

#### 4.13.2.3 int verifyMagicNumber ( struct HiddenFileByte \* *tabByte*, struct Condition \* *condition* )

Verifie si les derniers byte sont le magic number.

Cette fonction verifie les "magicNumberSize" dernier byte du tableau de byte.

## Parameters

<i>HIDDENFILEBYTE</i>	Tableau de byte
<i>Condition</i>	<i>Condition</i> de revelation (ici utilisé pour les informations sur le magic number)

## Returns

Retourne 1 si le magic\_number est trouvé, 0 sinon



# Index

addBinaryBufferInTab  
tools.c, [23](#)  
tools.h, [24](#)

BLOC  
HiddenFileByte, [6](#)

binary  
BufferBinary, [5](#)

binaryToByteTab  
tools.c, [23](#)  
tools.h, [25](#)

bitmapImage  
Img, [7](#)

BufferBinary, [5](#)  
binary, [5](#)  
indexNextBit, [5](#)

Byte  
tools.h, [24](#)

byte  
HiddenFileByte, [6](#)

channels  
Condition, [6](#)  
OptionList, [8](#)

colorImageCase  
pattern.c, [18](#)  
pattern.h, [20](#)

Condition, [5](#)  
channels, [6](#)  
magic\_Number, [6](#)  
magicNumberSize, [6](#)  
nbBits, [6](#)  
pattern, [6](#)

createHideFile  
output.c, [16](#)  
output.h, [17](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/image.c, [9](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/image.h, [11](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/main.c, [12](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/mathlib.c, [13](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/mathlib.h, [13](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/options↔  
Parser.c, [14](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/options↔  
Parser.h, [15](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/output.c, [15](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/output.h, [16](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/pattern.c, [17](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/pattern.h, [20](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/tools.c, [22](#)

D:/Users/Kevin/Documents/Cours/Polytech SI3/Polytech  
S5/Projet S5/private/reveal/src/tools.h, [24](#)

direct  
pattern.c, [18](#)  
pattern.h, [21](#)

erreur  
OptionList, [8](#)

external\_spiral  
pattern.c, [18](#)  
pattern.h, [21](#)

extractByteHiddenFileImage  
image.c, [9](#)  
image.h, [11](#)

finOption  
OptionList, [8](#)

getGoodFormat  
image.c, [10](#)

grayImageCase  
pattern.c, [19](#)  
pattern.h, [21](#)

hasInOption  
OptionList, [8](#)

hasOutOption  
OptionList, [8](#)

hasfinOption  
OptionList, [8](#)

height  
Img, [7](#)

HiddenFileByte, [6](#)  
BLOC, [6](#)  
byte, [6](#)  
indexNextByte, [6](#)  
tabSize, [6](#)

- image.c
  - extractByteHiddenFileImage, 9
  - getGoodFormat, 10
  - isColorImage, 10
  - loadImage, 10
  - verifyChannelAndImageFormat, 11
- image.h
  - extractByteHiddenFileImage, 11
  - loadImage, 12
- Img, 6
  - bitmapImage, 7
  - height, 7
  - isColor, 7
  - width, 7
- indexNextBit
  - BufferBinary, 5
- indexNextByte
  - HiddenFileByte, 6
- inputFile
  - OptionList, 8
- internal\_spiral
  - pattern.c, 19
  - pattern.h, 22
- isColor
  - Img, 7
- isColorImage
  - image.c, 10
- isCorrectBitsUsedOption
  - optionsParser.c, 14
  - optionsParser.h, 15
- isCorrectChannel
  - optionsParser.c, 14
- isCorrectFinOption
  - optionsParser.c, 14
  - optionsParser.h, 15
- isInFileExisting
  - optionsParser.c, 14
  - optionsParser.h, 15
- loadImage
  - image.c, 10
  - image.h, 12
- magic\_Number
  - Condition, 6
- magic\_number
  - OptionList, 8
- magicNumberSize
  - Condition, 6
  - OptionList, 8
- main
  - main.c, 13
- main.c
  - main, 13
- mathlib.c
  - power, 13
- mathlib.h
  - power, 13
- nbBits
  - Condition, 6
- numberBitsUsed
  - OptionList, 8
- OptionList, 7
  - channels, 8
  - erreur, 8
  - finOption, 8
  - hasInOption, 8
  - hasOutOption, 8
  - hasfinOption, 8
  - inputFile, 8
  - magic\_number, 8
  - magicNumberSize, 8
  - numberBitsUsed, 8
  - outputFile, 8
  - pattern\_used, 8
- optionsParser.c
  - isCorrectBitsUsedOption, 14
  - isCorrectChannel, 14
  - isCorrectFinOption, 14
  - isInFileExisting, 14
  - parseOption, 14
- optionsParser.h
  - isCorrectBitsUsedOption, 15
  - isCorrectFinOption, 15
  - isInFileExisting, 15
  - parseOption, 15
- output.c
  - createHideFile, 16
  - printOnStandardOutput, 16
- output.h
  - createHideFile, 17
  - printOnStandardOutput, 17
- outputFile
  - OptionList, 8
- parseOption
  - optionsParser.c, 14
  - optionsParser.h, 15
- pattern
  - Condition, 6
- pattern.c
  - colorImageCase, 18
  - direct, 18
  - external\_spiral, 18
  - grayImageCase, 19
  - internal\_spiral, 19
  - reverse, 19
- pattern.h
  - colorImageCase, 20
  - direct, 21
  - external\_spiral, 21
  - grayImageCase, 21
  - internal\_spiral, 22
  - reverse, 22
- pattern\_used
  - OptionList, 8

- power
  - mathlib.c, [13](#)
  - mathlib.h, [13](#)
- printOnStandardOutput
  - output.c, [16](#)
  - output.h, [17](#)
- reverse
  - pattern.c, [19](#)
  - pattern.h, [22](#)
- tabSize
  - HiddenFileByte, [6](#)
- tools.c
  - addBinaryBufferInTab, [23](#)
  - binaryToByteTab, [23](#)
  - verifyMagicNumber, [23](#)
- tools.h
  - addBinaryBufferInTab, [24](#)
  - binaryToByteTab, [25](#)
  - Byte, [24](#)
  - verifyMagicNumber, [25](#)
- verifyChannelAndImageFormat
  - image.c, [11](#)
- verifyMagicNumber
  - tools.c, [23](#)
  - tools.h, [25](#)
- width
  - Img, [7](#)