

cRabbitPopulationSimulation

Généré par Doxygen 1.8.16



<b>1 cRabbitPopulation</b>	<b>1</b>
<b>2 Index des structures de données</b>	<b>3</b>
2.1 Structures de données	3
<b>3 Index des fichiers</b>	<b>5</b>
3.1 Liste des fichiers	5
<b>4 Documentation des structures de données</b>	<b>7</b>
4.1 Référence de la structure configSimu_t	7
4.1.1 Description détaillée	7
4.1.2 Documentation des champs	7
4.1.2.1 ageDisminish	7
4.1.2.2 initPredator	8
4.1.2.3 littleRabbitSurvRate	8
4.1.2.4 maxBabyPerLitter	8
4.1.2.5 maxLitterPerYear	8
4.1.2.6 maxRabbitYear	8
4.1.2.7 maxYear	8
4.1.2.8 minBabyPerLitter	8
4.1.2.9 minLitterPerYear	9
4.1.2.10 probaLittersBabys	9
4.1.2.11 probaLittersYear	9
4.1.2.12 rabbitSurvRate	9
4.1.2.13 rateDisminish	9
4.2 Référence de la structure link	9
4.3 Référence de la structure rabbit_t	9
<b>5 Documentation des fichiers</b>	<b>11</b>
5.1 Référence du fichier main.c	11
5.1.1 Description détaillée	11
5.1.2 Documentation des fonctions	12
5.1.2.1 calcul_moyenne_long()	12
5.1.2.2 compute_S()	12
5.1.2.3 intervalle_de_confiance()	12
5.1.2.4 main()	13
5.1.2.5 question1()	13
5.1.2.6 question2_list()	13
5.1.2.7 question2_TS()	14
5.2 Référence du fichier rabbitTreeStatus.c	14
5.2.1 Description détaillée	14
5.2.2 Documentation des fonctions	14
5.2.2.1 addBabys()	14
5.2.2.2 CalcSumYears()	15

---

5.2.2.3 checkMallocNull()	15
5.2.2.4 copyRabitTab()	16
5.2.2.5 createRabbitTabZero()	16
5.2.2.6 free_double_tab()	16
5.2.2.7 makeBabys()	17
5.2.2.8 rabbit_simu_iteration()	17
5.2.2.9 realistic_simulation_TS()	18
5.2.2.10 sum_array()	18
5.2.2.11 surviveRabbitYear()	19
5.3 Référence du fichier rabbitTreeStatus.h	19
5.3.1 Description détaillée	20
5.3.2 Documentation des fonctions	20
5.3.2.1 realistic_simulation_TS()	20
5.4 Référence du fichier reverseDistLow.c	20
5.4.1 Description détaillée	21
5.4.2 Documentation des fonctions	21
5.4.2.1 calcCumulProba()	21
5.4.2.2 discDispLoop()	21
5.4.2.3 genRand_discDist()	22
5.4.2.4 probaCalc()	22
<b>Index</b>	<b>25</b>

# Chapitre 1

## cRabbitPopulation

Simule une population de lapins en prenant en compte leur probabilité de survivre ainsi que leur probabilité de donner naissance à l'aide de nombres pseudo-aléatoires.

Projet réalisé dans le cadre de mes études en L2 informatique

Générateur de nombres pseudo-aléatoires utilisé : Mersenne Twister de 2002 par Makoto Matsumoto link ↔  
: <http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/MT2002/emt19937ar.html>



## Chapitre 2

# Index des structures de données

### 2.1 Structures de données

Liste des structures de données avec une brève description :

<a href="#">configSimu_t</a>		
	Configuration d'une simulation . . . . .	7
<a href="#">link</a>	. . . . .	9
<a href="#">rabbit_t</a>	. . . . .	9





## Chapitre 3

# Index des fichiers

### 3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

<b>fibo.h</b>	??
<a href="#">main.c</a>	
Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin	11
<b>mt19937ar.h</b>	??
<b>rabbit.h</b>	??
<b>rabbit_list.h</b>	??
<a href="#">rabbitTreeStatus.c</a>	
Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin avec des arbres d'états	14
<a href="#">rabbitTreeStatus.h</a>	
Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin avec des arbres d'états	19
<a href="#">reverseDistLow.c</a>	
Programmes pour générer des nombre pseudo-aléatoire qui suivent une probabilité donné	20
<b>reverseDistLow.h</b>	??



## Chapitre 4

# Documentation des structures de données

### 4.1 Référence de la structure configSimu\_t

configuration d'une simulation

```
#include <rabbitTreeStatus.h>
```

#### Champs de données

- int maxBabyPerLitter
- int minBabyPerLitter
- double littleRabbitSurvRate
- double rabbitSurvRate
- double rateDisminish
- int ageDisminish
- int minLitterPerYear
- int maxLitterPerYear
- double \* probaLittersYear
- double \* probaLittersBabys
- int maxRabbitYear
- int maxYear
- int initPredator

#### 4.1.1 Description détaillée

configuration d'une simulation

#### 4.1.2 Documentation des champs

##### 4.1.2.1 ageDisminish

```
int ageDisminish
```

age où les lapin perde du taux de survie

#### 4.1.2.2 initPredator

```
int initPredator
```

mois a partir duquel les bébé peuvent mourir

#### 4.1.2.3 littleRabbitSurvRate

```
double littleRabbitSurvRate
```

taux de survie d'un bébé lapin

#### 4.1.2.4 maxBabyPerLitter

```
int maxBabyPerLitter
```

nombre Max de bébé par portées

#### 4.1.2.5 maxLitterPerYear

```
int maxLitterPerYear
```

nombre max de portées par ans

#### 4.1.2.6 maxRabbitYear

```
int maxRabbitYear
```

age mas des lapins

#### 4.1.2.7 maxYear

```
int maxYear
```

nombre d'années de simulation

#### 4.1.2.8 minBabyPerLitter

```
int minBabyPerLitter
```

nombre min de bébé par portées

#### 4.1.2.9 minLitterPerYear

```
int minLitterPerYear
```

nombre min de portées par ans

#### 4.1.2.10 probaLittersBabys

```
double* probaLittersBabys
```

probabilités de bébé

#### 4.1.2.11 probaLittersYear

```
double* probaLittersYear
```

probabilités de portées

#### 4.1.2.12 rabbitSurvRate

```
double rabbitSurvRate
```

taux de survie d'un lapin adulte

#### 4.1.2.13 rateDisminish

```
double rateDisminish
```

taux de survie perdu après ageDisminish

La documentation de cette structure a été générée à partir du fichier suivant :

— [rabbitFreeStatus.h](#)

## 4.2 Référence de la structure link

### Champs de données

- [rabbit\\_t](#) **rabbit**
- [struct link](#) \* **next**

La documentation de cette structure a été générée à partir du fichier suivant :

— [rabbit\\_list.h](#)

## 4.3 Référence de la structure rabbit\_t

### Champs de données

- [sexe\\_t](#) **sexe**
- [int](#) **nb\_years**
- [Boolean](#) **alive**

La documentation de cette structure a été générée à partir du fichier suivant :

— [rabbit\\_list.h](#)



## Chapitre 5

# Documentation des fichiers

### 5.1 Référence du fichier main.c

Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin.

```
#include "fibo.h"
#include "rabbit.h"
#include "rabbitTreeStatus.h"
#include <stdio.h>
```

#### Fonctions

- void [question1](#) ()
- double [compute\\_S](#) (unsigned long tab[], int n, double moyenne)
- double [calcul\\_moyenne\\_long](#) (unsigned long tab[], int n)
- void [intervalle\\_de\\_confiance](#) (unsigned long tab[], int nbExp)
- void [question2\\_list](#) ()
- void [question2\\_TS](#) ()
- int [main](#) ()

#### 5.1.1 Description détaillée

Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin.

##### Auteur

Aurelien DOUARD et Anthony BERTRAND

##### Version

0.1

##### Date

16 mars 2020

## 5.1.2 Documentation des fonctions

### 5.1.2.1 calcul\_moyenne\_long()

```
double calcul_moyenne_long (
    unsigned long tab[],
    int n )
```

fonction pour calculer la moyenne d'un tableau de unsigned long

#### Paramètres

<i>tab</i>	tableau
<i>n</i>	taille du tableau

#### Renvoie

moyenne du tableau

### 5.1.2.2 compute\_S()

```
double compute_S (
    unsigned long tab[],
    int n,
    double moyenne )
```

calcul l'estimateur sans biais de la variance

#### Paramètres

<i>tab</i>	tableau
<i>n</i>	taille du tableau
<i>moyenne</i>	moyenne du tableau

#### Renvoie

S

### 5.1.2.3 intervalle\_de\_confiance()

```
void intervalle_de_confiance (
    unsigned long tab[],
    int nbExp )
```

fonction pour calculer et afficher l'intervalle de confiance d'un tableau de unsigned long



## Paramètres

<i>tab</i>	tableau
<i>nbExp</i>	taille du tableau / nombre d'espériences

## Renvoie

void

**5.1.2.4 main()**

```
int main ( )
```

point d'entré du programe, ici elle appelle question2\_TS et initialise le générateur de nombre pseudo-aléatoire

## Renvoie

void

**5.1.2.5 question1()**

```
void question1 ( )
```

calcul le 11 eme élément de la suite de fibonacci avec trois méthodes différentes

## Renvoie

void

**5.1.2.6 question2\_list()**

```
void question2_list ( )
```

fonction pour réaliser une simulation d'évolution de population de lapin avec une structure list

## Renvoie

void

### 5.1.2.7 question2\_TS()

```
void question2_TS ( )
```

fonction pour réaliser une simulation d'évolution de population de lapin avec une structure d'arbre d'états

Renvoie

```
void
```

## 5.2 Référence du fichier rabbitTreeStatus.c

Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin avec des arbres d'états.

```
#include "rabbitTreeStatus.h"
```

### Fonctions

- void [checkMallocNull](#) (void \*tab)
- unsigned long \*\* [createRabbitTabZero](#) (int maxRabbitYear)
- void [copyRabbitTab](#) (unsigned long \*\*src, unsigned long \*\*dest, int maxRabbitYear)
- int [surviveRabbitYear](#) (int old, [configSimu\\_t](#) config)
- unsigned long [makeBabys](#) (int old, double cumulProbaLitter[], double cumulProbaBabys[], int lenCumulProbaLitter, int lenCumulProbaBabys, int minLitter, int minBaby)
- unsigned long [CalcSumYears](#) (unsigned long \*\*tab, int maxOld)
- void [addBabys](#) (unsigned long babys, unsigned long \*\*rabbit)
- void [free\\_double\\_tab](#) (unsigned long \*\*tab, int n)
- unsigned long [sum\\_array](#) (unsigned long array[], int n)
- unsigned long [rabbit\\_simu\\_iteration](#) ([configSimu\\_t](#) config, unsigned long \*\*rabbits, unsigned long \*\*rabbitsTmp, int year, double \*cumulProbaLitter, double \*cumulProbaBabys, int lenCumulProbaLitter, int lenCumulProbaBabys)
- unsigned long [realistic\\_simulation\\_TS](#) ([configSimu\\_t](#) config)

### 5.2.1 Description détaillée

Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin avec des arbres d'états.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

### 5.2.2 Documentation des fonctions

#### 5.2.2.1 addBabys()

```
void addBabys (
    unsigned long  babys,
    unsigned long ** rabbit )
```

fonction pour ajouter les lapins a la liste des lapins

## Paramètres

<i>babys</i>	nombre de bébés a ajouter
<i>rabbit</i>	tableaux d'états de lapins

## Renvoie

void

**5.2.2.2 CalcSumYears()**

```
unsigned long CalcSumYears (
    unsigned long ** tab,
    int maxOld )
```

fonction pour calculer la somme des lapins

## Paramètres

<i>tab</i>	tableau
<i>maxOld</i>	age max des lapins

## Renvoie

la somme des lapins

**5.2.2.3 checkMallocNull()**

```
void checkMallocNull (
    void * tab )
```

vérifi si un malloc c'est bien passé

## Paramètres

<i>tab</i>	tableau
------------	---------

## Renvoie

void

#### 5.2.2.4 copyRabitTab()

```
void copyRabitTab (
    unsigned long ** src,
    unsigned long ** dest,
    int maxRabbitYear )
```

fonction pour copier des tableaux d'états de lapin

##### Paramètres

<i>src</i>	tableau source
<i>dest</i>	tableau de destination
<i>maxRabbitYear</i>	l'age maximum d'un lapin

##### Renvoie

void

#### 5.2.2.5 createRabbitTabZero()

```
unsigned long** createRabbitTabZero (
    int maxRabbitYear )
```

crée un tableau pour stocker des états de lapin, initialisé avec des zero

##### Paramètres

<i>maxRabbitYear</i>	l'age maximum d'un lapin
----------------------	--------------------------

##### Renvoie

un tableau de tableaux de unsigned long

#### 5.2.2.6 free\_double\_tab()

```
void free_double_tab (
    unsigned long ** tab,
    int n )
```

fonction pour free un tableau de tableaux de unsigned long

##### Paramètres

<i>tab</i>	tableau
<i>n</i>	nombre d'éléments dans le tableau

**Renvoie**

void

**5.2.2.7 makeBabys()**

```
unsigned long makeBabys (
    int old,
    double cumulProbaLitter[],
    double cumulProbaBabys[],
    int lenCumulProbaLitter,
    int lenCumulProbaBabys,
    int minLitter,
    int minBaby )
```

fonction pour savoir combien de lapins va faire un lapin donné pendant une année

**Paramètres**

<i>old</i>	age du lapin
<i>cumulProbaLitter</i>	tableau de probabilité cumulé pour le nombre de portée
<i>cumulProbaBabys</i>	tableau de probabilité cumulé pour le nombre de bébés par portée
<i>lenCumulProbaLitter</i>	longueur du tableau de proba de portée
<i>lenCumulProbaBabys</i>	longueur du tableau de proba de bébés
<i>minLitter</i>	nombre minimum de portées
<i>minBaby</i>	nombre minimum de bébés

**Renvoie**

1 ou 0

**5.2.2.8 rabbit\_simu\_iteration()**

```
unsigned long rabbit_simu_iteration (
    configSimu_t config,
    unsigned long ** rabbits,
    unsigned long ** rabbitsTmp,
    int year,
    double * cumulProbaLitter,
    double * cumulProbaBabys,
    int lenCumulProbaLitter,
    int lenCumulProbaBabys )
```

fonction qui réalise une itération de la simulation d'évolution de population de lapin

**Paramètres**

<i>config</i>	configuration de la simulation
---------------	--------------------------------

**Paramètres**

<i>rabbits</i>	tableau d'états de lapins
<i>rabbitsTmp</i>	tableau d'états de lapins temporaire
<i>year</i>	année de l'iteration
<i>cumulProbaLitter</i>	tableau de probabilité cumulé pour le nombre de portée
<i>cumulProbaBabys</i>	tableau de probabilité cumulé pour le nombre de bébés par portée
<i>lenCumulProbaLitter</i>	longueur du tableau de proba de portée
<i>lenCumulProbaBabys</i>	longueur du tableau de proba de bébés
<i>n</i>	nombre d'éléments dans le tableau

**Renvoie**

nombre de bébés à ajouter

**5.2.2.9 realistic\_simulation\_TS()**

```
unsigned long realistic_simulation_TS (
    configSimu_t config )
```

fonction pour réaliser une simulation d'évolution de population de lapin sur plusieurs années avec les configurations données

**Paramètres**

<i>config</i>	configuration de la simulation
---------------	--------------------------------

**Renvoie**

nombre de lapin final

**5.2.2.10 sum\_array()**

```
unsigned long sum_array (
    unsigned long array[],
    int n )
```

fonction pour faire la somme d'un tableaux de unsigned long

**Paramètres**

<i>array</i>	tableau
<i>n</i>	nombre d'éléments dans le tableau

**Renvoie**

somme

**5.2.2.11 surviveRabbitYear()**

```
int surviveRabbitYear (
    int old,
    configSimu_t config )
```

fonction qui dit si un lapin survie ou non (renvoie 1 ou 0)

**Paramètres**

<i>old</i>	age du lapin
<i>config</i>	configuration de la simulation

**Renvoie**

1 ou 0

**5.3 Référence du fichier rabbitTreeStatus.h**

Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin avec des arbres d'états.

```
#include "mt19937ar.h"
#include "reverseDistLow.h"
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
```

**Structures de données**

— struct `configSimu_t`  
configuration d'une simulation

**Fonctions**

— unsigned long `realistic_simulation_TS` (`configSimu_t` config)

### 5.3.1 Description détaillée

Programmes pour réaliser une simulation stochastique de l'évolution d'une population de lapin avec des arbres d'états.

#### Auteur

Aurelien DOUARD et Anthony BERTRAND

#### Version

0.1

#### Date

16 mars 2020

### 5.3.2 Documentation des fonctions

#### 5.3.2.1 realistic\_simulation\_TS()

```
unsigned long realistic_simulation_TS (  
    configSimu_t config )
```

fonction pour réaliser une simulation d'évolution de population de lapin sur plusieurs années avec les configurations données

#### Paramètres

<i>config</i>	configuration de la simulation
---------------	--------------------------------

#### Renvoie

nombre de lapin final

## 5.4 Référence du fichier reverseDistLow.c

Programmes pour générer des nombre pseudo-aléatoire qui suivent une probabilité donné

```
#include "reverseDistLow.h"
```

### Fonctions

- double \* [probaCalc](#) (int tab[], int len)  
*Fonction qui calcule la probabilité de chaque élément du tableau.*
- void **discDist** (double rand, int tab[], double cumulProba[], int nbClass)



- int `genRand_discDist` (double rand, double cumulProba[ ], int nbClass)  
*Fonction qui génère un nombre qui suit une probabilité donné à partir d'un nombre pseudo-aléatoire.*
- int \* `discDispLoop` (double cumulProba[ ], int nbClass, int nbExp)  
*Fonction qui générer des nombre qui suivent une probabilité pour x tirage.*
- double \* `calcCumulProba` (double tabProb[ ], int len)  
*Fonction qui calcule la probabilité cumulé de chaque élément du tableau.*

### 5.4.1 Description détaillée

Programmes pour générer des nombre pseudo-aléatoire qui suivent une probabilité donné

#### Auteur

Aurelien DOUARD

#### Version

0.1

#### Date

16 février 2020

### 5.4.2 Documentation des fonctions

#### 5.4.2.1 calcCumulProba()

```
double * calcCumulProba (
    double tabProb[ ],
    int len )
```

Fonction qui calcule la probabilité cumulé de chaque élément du tableau.

#### Paramètres

<i>tabProb</i>	tableau rempli avec les probabilité
<i>len</i>	la taile du tableau

#### Renvoie

un tableau avec la probabilité cumulé de chaque élément

#### 5.4.2.2 discDispLoop()

```
int * discDispLoop (
    double cumulProba[ ],
```

```
int nbClass,  
int nbExp )
```

Fonction qui générer des nombre qui suivent une probabilité pour x tirage.

#### Paramètres

<i>nbExp</i>	nombre de tirages
<i>nbClass</i>	nombre de classes différentes du dans le tableau
<i>cumulProba</i>	tableau rempli avec la probabilité cumulé pour chaque classes

#### Renvoie

tableau rempli avec fréquences

#### 5.4.2.3 genRand\_discDist()

```
int genRand_discDist (  
    double rand,  
    double cumulProba[],  
    int nbClass )
```

Fonction qui génère un nombre qui suit une probabilité donné à partir d'un nombre pseudo-aléatoire.

#### Paramètres

<i>nbClass</i>	nombre de classes différentes du dans le tableau
<i>cumulProba</i>	tableau rempli avec la probabilité cumulé pour chaque classes
<i>rand</i>	nombre pseudo-aléatoire

#### 5.4.2.4 probaCalc()

```
double * probaCalc (  
    int tab[],  
    int len )
```

Fonction qui calcule la probabilité de chaque élément du tableau.

#### Paramètres

<i>tab</i>	tableau rempli avec les evenements
<i>len</i>	la taile du tableau

**Renvoie**

un tableau avec la probabilité de chaque élément



# Index

addBabys  
    rabbitTreeStatus.c, 14  
ageDisminish  
    configSimu\_t, 7  
  
calcCumulProba  
    reverseDistLow.c, 21  
CalcSumYears  
    rabbitTreeStatus.c, 15  
calcul\_moyenne\_long  
    main.c, 12  
checkMallocNull  
    rabbitTreeStatus.c, 15  
compute\_S  
    main.c, 12  
configSimu\_t, 7  
    ageDisminish, 7  
    initPredator, 7  
    littleRabbitSurvRate, 8  
    maxBabyPerLitter, 8  
    maxLitterPerYear, 8  
    maxRabbitYear, 8  
    maxYear, 8  
    minBabyPerLitter, 8  
    minLitterPerYear, 8  
    probaLittersBabys, 9  
    probaLittersYear, 9  
    rabbitSurvRate, 9  
    rateDisminish, 9  
copyRabitTab  
    rabbitTreeStatus.c, 15  
createRabbitTabZero  
    rabbitTreeStatus.c, 16  
  
discDispLoop  
    reverseDistLow.c, 21  
  
free\_double\_tab  
    rabbitTreeStatus.c, 16  
  
genRand\_discDist  
    reverseDistLow.c, 22  
  
initPredator  
    configSimu\_t, 7  
intervalle\_de\_confiance  
    main.c, 12  
  
link, 9  
littleRabbitSurvRate  
    configSimu\_t, 8  
  
main  
    main.c, 13  
main.c, 11  
    calcul\_moyenne\_long, 12  
    compute\_S, 12  
    intervalle\_de\_confiance, 12  
    main, 13  
    question1, 13  
    question2\_list, 13  
    question2\_TS, 13  
makeBabys  
    rabbitTreeStatus.c, 17  
maxBabyPerLitter  
    configSimu\_t, 8  
maxLitterPerYear  
    configSimu\_t, 8  
maxRabbitYear  
    configSimu\_t, 8  
maxYear  
    configSimu\_t, 8  
minBabyPerLitter  
    configSimu\_t, 8  
minLitterPerYear  
    configSimu\_t, 8  
  
probaCalc  
    reverseDistLow.c, 22  
probaLittersBabys  
    configSimu\_t, 9  
probaLittersYear  
    configSimu\_t, 9  
  
question1  
    main.c, 13  
question2\_list  
    main.c, 13  
question2\_TS  
    main.c, 13  
  
rabbit\_simu\_iteration  
    rabbitTreeStatus.c, 17  
rabbit\_t, 9  
rabbitSurvRate  
    configSimu\_t, 9  
rabbitTreeStatus.c, 14  
    addBabys, 14  
    CalcSumYears, 15  
    checkMallocNull, 15  
    copyRabitTab, 15  
    createRabbitTabZero, 16

- free\_double\_tab, [16](#)
- makeBabys, [17](#)
- rabbit\_simu\_iteration, [17](#)
- realistic\_simulation\_TS, [18](#)
- sum\_array, [18](#)
- surviveRabbitYear, [19](#)
- rabbitTreeStatus.h, [19](#)
  - realistic\_simulation\_TS, [20](#)
- rateDisminish
  - configSimu\_t, [9](#)
- realistic\_simulation\_TS
  - rabbitTreeStatus.c, [18](#)
  - rabbitTreeStatus.h, [20](#)
- reverseDistLow.c, [20](#)
  - calcCumulProba, [21](#)
  - discDispLoop, [21](#)
  - genRand\_discDist, [22](#)
  - probaCalc, [22](#)
- sum\_array
  - rabbitTreeStatus.c, [18](#)
- surviveRabbitYear
  - rabbitTreeStatus.c, [19](#)