cRabbitPopulationSimulation

Généré par Doxygen 1.8.16

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 :

Chapitre 3

Index des fichiers

3.1 Liste des fichiers

Liste de tous les fichiers documentés avec une brève description :

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 :
— rabbitTreeStatus.h

4.2 Référence de la structure link

Champs de données

— rabbit_t rabbit
— struct link * next

4.2.1 Documentation des champs

4.2.1.1 next

```
struct link* next
```

pointeur sur la cellule suivante

4.2.1.2 rabbit

```
rabbit_t rabbit
```

valeur de la cellule, ici un lapin

La documentation de cette structure a été générée à partir du fichier suivant :
— rabbit_list.h

4.3 Référence de la structure link_t

structure qui définit une cellule de liste chainée de lapins

```
#include <rabbit_list.h>
```

4.3.1 Description détaillée

structure qui définit une cellule de liste chainée de lapins

La documentation de cette structure a été générée à partir du fichier suivant :
— rabbit_list.h

4.4 Référence de la structure list_rabbit_t

structure qui définit une liste chainée de lapins

```
#include <rabbit_list.h>
```

4.4.1 Description détaillée

structure qui définit une liste chainée de lapins

La documentation de cette structure a été générée à partir du fichier suivant :
— rabbit_list.h

4.5 Référence de la structure rabbit_t

structure qui définit un lapin

`#include <rabbit_list.h>`

Champs de données

- sexe_t sexe
- int nb_years
- Boolean alive

4.5.1 Description détaillée

structure qui définit un lapin

4.5.2 Documentation des champs

4.5.2.1 alive

`Boolean alive`

variable qui définit si le lapin est vivant

4.5.2.2 nb_years

`int nb_years`

age du lapin

4.5.2.3 sexe

`sexe_t sexe`

sexe du lapin

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 fibo.c

Programmes pour calculer les termes de la suite de Fibonacci.

```
#include "fibo.h"
```

Fonctions

- unsigned long fibo_recursif (int n)
- unsigned long fibo_recurence (int n)
- unsigned long fibo_explicite (int n)

5.1.1 Description détaillée

Programmes pour calculer les termes de la suite de Fibonacci.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.1.2 Documentation des fonctions

5.1.2.1 fibo_explicite()

```
unsigned long fibo_explicite (
            int n )
```

fonction pour obtenir le n-ième terme de la suite de Fibonacci implémentée à l'aide de la formule explicite

Paramètres

n	indice du terme voulu

Renvoie

le terme à la n-ième place dans la suite

5.1.2.2 fibo_recurence()

```
unsigned long fibo_recurence (
             int n )
```

fonction pour obtenir le n-ième terme de la suite de Fibonacci implémentée à l'aide de la formule de récurrence

Paramètres

n	indice du terme voulu

Renvoie

le terme à la n-ième place dans la suite

5.1.2.3 fibo_recursif()

```
unsigned long fibo_recursif (
             int n )
```

fonction pour obtenir le n-ième terme de la suite de Fibonacci implémentée récursivement

Paramètres

n	indice du terme voulu

Renvoie

le terme à la n-ième place dans la suite

5.2 Référence du fichier fibo.h

Programmes pour calculer les termes de la suite de Fibonacci.

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

Fonctions

— unsigned long fibo_recursif (int n)
— unsigned long fibo_recurence (int n)
— unsigned long fibo_explicite (int n)

5.2.1 Description détaillée

Programmes pour calculer les termes de la suite de Fibonacci.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.2.2 Documentation des fonctions

5.2.2.1 fibo_explicite()

```
unsigned long fibo_explicite (
            int n )
```

fonction pour obtenir le n-ième terme de la suite de Fibonacci implémentée à l'aide de la formule explicite

Paramètres

n	indice du terme voulu

Renvoie

le terme à la n-ième place dans la suite

5.2.2.2 fibo_recurence()

```
unsigned long fibo_recurence (
            int n )
```

fonction pour obtenir le n-ième terme de la suite de Fibonacci implémentée à l'aide de la formule de récurrence

Paramètres

n	indice du terme voulu

Renvoie

le terme à la n-ième place dans la suite

5.2.2.3 fibo_recursif()

```
unsigned long fibo_recursif (
             int n )
```

fonction pour obtenir le n-ième terme de la suite de Fibonacci implémentée récursivement

Paramètres

n	indice du terme voulu

Renvoie

le terme à la n-ième place dans la suite

5.3 Référence du fichier main.c

Programmes pour réaliser une simulation stocastique 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 ()

5.3.1 Description détaillée

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

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.3.2 Documentation des fonctions

5.3.2.1 calcul_moyenne_long()

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

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

Paramètres

tab	tableau
n	taille du tableau

Renvoie

moyenne du tableau

5.3.2.2 compute_S()

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

calcul l'estimateur sans biais de la variance

Paramètres

tab	tableau
n	taille du tableau
moyenne	moyenne du tableau

Renvoie

S

5.3.2.3 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

tab	tableau
nbExp	taille du tableau / nombre d'espériences

Renvoie

void

5.3.2.4 main()

```
int main ( )
```

point d'entré du programe, ici elle appelle question2_TS et initialise le générateur de nombre pseudo-aléatoire

Renvoie

void

5.3.2.5 question1()

```
void question1 ( )
```

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

Renvoie

void

5.3.2.6 question2_list()

```
void question2_list ( )
```

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

Renvoie

void

5.3.2.7 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

5.4 Référence du fichier rabbit.c

Programmes pour gérer les lapins ainsi que la simulation de l'évolution de la population de lapin dans le temps.

```
#include "rabbit.h"
```

Fonctions

- rabbit_t new_rabbit (int years)
- rabbit_t new_rabbit2 (int years, sexe_t sexe)
- sexe_t is_male (rabbit_t rabbit)
- sexe_t is_female (rabbit_t rabbit)
- Boolean give_birth (rabbit_t rabbit)
- Boolean survive (rabbit_t rabbit)
- Boolean is_alive (rabbit_t rabbit)
- list_rabbit_t give_birth_all (list_rabbit_t list_rabbit)
- list_rabbit_t survive_all (list_rabbit_t list_rabbit)
- void realistic_simulation ()

Variables

- int **nb_loop** = 1

5.4.1 Description détaillée

Programmes pour gérer les lapins ainsi que la simulation de l'évolution de la population de lapin dans le temps.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.4.2 Documentation des fonctions

5.4.2.1 give_birth()

```
Boolean give_birth (
            rabbit_t rabbit )
```

permet de savoir si le lapin donne naissance à une portée

Paramètres

rabbit	un lapin

Renvoie

true si le lapin donne naissance

5.4.2.2 give_birth_all()

```
list_rabbit_t give_birth_all (
            list_rabbit_t list_rabbit )
```

ajoute tout les nouveaux nés dans la liste de lapins

Paramètres

list_rabbit	une liste de lapin

Renvoie

la nouvelle liste de lapins après insertion des nouveaux nés

5.4.2.3 is_alive()

```
Boolean is_alive (
            rabbit_t rabbit )
```

retourne la variable bouléene alive du lapin

Paramètres

rabbit	un lapin

Renvoie

true si le lapin est en vie

5.4.2.4 is_female()

```
sexe_t is_female (
            rabbit_t rabbit )
```

permet de savoir si le lapin est une femelle

Paramètres

rabbit	un lapin

Renvoie

true si le lapin est une femelle

5.4.2.5 is_male()

```
sexe_t is_male (
            rabbit_t rabbit )
```

permet de savoir si le lapin est un mâle

Paramètres

rabbit	un lapin

Renvoie

true si le lapin est un mâle

5.4.2.6 new_rabbit()

```
rabbit_t new_rabbit (
            int years )
```

créer un lapin de sexe aléatoire

Paramètres

years	age du lapin

Renvoie

un lapin

5.4.2.7 new_rabbit2()

```
rabbit_t new_rabbit2 (
            int years,
            sexe_t sexe )
```

créer un lapin de sexe défini

Paramètres

years	age du lapin
sexe	sexe du lapin

Renvoie

un lapin

5.4.2.8 realistic_simulation()

```
void realistic_simulation ( )
```

fonction pour réaliser une simulation d'évolution de population de lapin sur plusieurs années

5.4.2.9 survive()

```
Boolean survive (
            rabbit_t rabbit )
```

permet de savoir si le lapin survit jusqu'à l'année suivante

Paramètres

rabbit	un lapin

Renvoie

true si le lapin survit

5.4.2.10 survive_all()

```
list_rabbit_t survive_all (
            list_rabbit_t list_rabbit )
```

change la valeur du bouléen alive pour tous les lapins de la liste

Paramètres

list_rabbit	une liste de lapin

Renvoie

la nouvelle liste de lapins après modification du bouléen

5.5 Référence du fichier rabbit.h

Programmes pour gérer les lapins ainsi que la simulation de l'évolution de la population de lapin dans le temps.

```
#include "fibo.h"
#include "mt19937ar.h"
#include "rabbit_list.h"
#include <stdio.h>
#include <stdlib.h>
```

Fonctions

- rabbit_t new_rabbit (int years)
- rabbit_t new_rabbit2 (int years, sexe_t sexe)
- sexe_t is_male (rabbit_t rabbit)
- sexe_t is_female (rabbit_t rabbit)
- Boolean give_birth (rabbit_t rabbit)
- Boolean survive (rabbit_t rabbit)
- Boolean is_alive (rabbit_t rabbit)
- list_rabbit_t survive_all (list_rabbit_t list_rabbit)
- void realistic_simulation ()

5.5.1 Description détaillée

Programmes pour gérer les lapins ainsi que la simulation de l'évolution de la population de lapin dans le temps.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.5.2 Documentation des fonctions

5.5.2.1 give_birth()

```
Boolean give_birth (
            rabbit_t rabbit )
```

permet de savoir si le lapin donne naissance à une portée

Paramètres

rabbit	un lapin

Renvoie

true si le lapin donne naissance

5.5.2.2 is_alive()

```
Boolean is_alive (
            rabbit_t rabbit )
```

retourne la variable bouléene alive du lapin

Paramètres

rabbit	un lapin

Renvoie

true si le lapin est en vie

5.5.2.3 is_female()

```
sexe_t is_female (
            rabbit_t rabbit )
```

permet de savoir si le lapin est une femelle

Paramètres

rabbit	un lapin

Renvoie

true si le lapin est une femelle

5.5.2.4 is_male()

```
sexe_t is_male (
            rabbit_t rabbit )
```

permet de savoir si le lapin est un mâle

Paramètres

rabbit	un lapin

Renvoie

true si le lapin est un mâle

5.5.2.5 new_rabbit()

```
rabbit_t new_rabbit (
            int years )
```

créer un lapin de sexe aléatoire

Paramètres

years	age du lapin

Renvoie

un lapin

5.5.2.6 new_rabbit2()

```
rabbit_t new_rabbit2 (
            int years,
            sexe_t sexe )
```

créer un lapin de sexe défini

Paramètres

years	age du lapin
sexe	sexe du lapin

Renvoie

un lapin

5.5.2.7 realistic_simulation()

```
void realistic_simulation ( )
```

fonction pour réaliser une simulation d'évolution de population de lapin sur plusieurs années

5.5.2.8 survive()

```
Boolean survive (
            rabbit_t rabbit )
```

permet de savoir si le lapin survit jusqu'à l'année suivante

Paramètres

rabbit	un lapin

Renvoie

true si le lapin survit

5.5.2.9 survive_all()

```
list_rabbit_t survive_all (
            list_rabbit_t list_rabbit )
```

change la valeur du bouléen alive pour tous les lapins de la liste

Paramètres

list_rabbit	une liste de lapin

Renvoie

la nouvelle liste de lapins après modification du bouléen

5.6 Référence du fichier rabbit_list.c

Programmes pour stocker des lapins dans une liste.

```
#include "rabbit_list.h"
```

Fonctions

- list_rabbit_t new_list (void)
- rabbit_t head (list_rabbit_t list_rabbit)
- Boolean empty (list_rabbit_t list_rabbit)
- list_rabbit_t add_head (list_rabbit_t list_rabbit, rabbit_t rabbit)
- list_rabbit_t delete_head (list_rabbit_t list_rabbit)
- list_rabbit_t delete_dead (list_rabbit_t list_rabbit)
- void display_rabbit (rabbit_t rabbit)
- void display_rabbit_all (list_rabbit_t list_rabbit)
- void display_death (list_rabbit_t list_rabbit)
- int size_list (list_rabbit_t list_rabbit)

5.6.1 Description détaillée

Programmes pour stocker des lapins dans une liste.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.6.2 Documentation des fonctions

5.6.2.1 add_head()

```
list_rabbit_t add_head (
            list_rabbit_t list_rabbit,
            rabbit_t rabbit )
```

ajoute un lapin dans la liste

Paramètres

list_rabbit	liste de lapins
rabbit	lapin à ajouter à la liste

Renvoie

la liste avec le lapin ajouté

5.6.2.2 delete_dead()

```
list_rabbit_t delete_dead (
            list_rabbit_t list_rabbit )
```

supprime tout les lapins morts de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

la liste avec les lapins morts supprimés

5.6.2.3 delete_head()

```
list_rabbit_t delete_head (
            list_rabbit_t list_rabbit )
```

supprime le premier lapin de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

la liste avec le lapin supprimé

5.6.2.4 display_death()

```
void display_death (
            list_rabbit_t list_rabbit )
```

affiche la valeur de la variable bouléene alive (pour debug)

Paramètres

list_rabbit	liste de lapins

5.6.2.5 display_rabbit()

```
void display_rabbit (
            rabbit_t rabbit )
```

affiche les informations d'un lapin

Paramètres

rabbit	un lapins

5.6.2.6 display_rabbit_all()

```
void display_rabbit_all (
            list_rabbit_t list_rabbit )
```

affiche les informations de tous les lapins de la liste

Paramètres

list_rabbit	liste de lapins

5.6.2.7 empty()

```
Boolean empty (
            list_rabbit_t list_rabbit )
```

permet de savoir si la liste est vide

Paramètres

list_rabbit	liste de lapins

Renvoie

true si la liste est vide, false sinon

5.6.2.8 head()

```
rabbit_t head (
            list_rabbit_t list_rabbit )
```

permet d'obtenir le premier élement de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

le premier lapin de la liste

5.6.2.9 new_list()

```
list_rabbit_t new_list (
            void  )
```

créer une liste vide

Renvoie

NULL

5.6.2.10 size_list()

```
int size_list (
            list_rabbit_t list_rabbit )
```

compte le nombre d'éléments de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

le nombre d'éléments de la liste

5.7 Référence du fichier rabbit_list.h

Programmes pour stocker des lapins dans une liste.

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

Structures de données

- struct rabbit_t
  structure qui définit un lapin
- struct link

Définitions de type

- typedef struct link **link_t**
- typedef struct link * **list_rabbit_t**

Énumérations

- enum sexe_t { **male**, **female** }
  structure pour définir le sexe d'un lapin
- enum Boolean { **false**, **true** }
  structure pour définir un type boolean

Fonctions

- list_rabbit_t new_list (void)
- rabbit_t head (list_rabbit_t list_rabbit)
- Boolean empty (list_rabbit_t list_rabbit)
- list_rabbit_t add_head (list_rabbit_t list_rabbit, rabbit_t rabbit)
- list_rabbit_t delete_head (list_rabbit_t list_rabbit)
- list_rabbit_t delete_dead (list_rabbit_t list_rabbit)
- void display_rabbit (rabbit_t rabbit)
- void display_rabbit_all (list_rabbit_t list_rabbit)
- void display_death (list_rabbit_t list_rabbit)
- int size_list (list_rabbit_t list_rabbit)

5.7.1 Description détaillée

Programmes pour stocker des lapins dans une liste.

Auteur

Aurelien DOUARD et Anthony BERTRAND

Version

0.1

Date

16 mars 2020

5.7.2 Documentation des fonctions

5.7.2.1 add_head()

```
list_rabbit_t add_head (
            list_rabbit_t list_rabbit,
            rabbit_t rabbit )
```

ajoute un lapin dans la liste

Paramètres

list_rabbit	liste de lapins
rabbit	lapin à ajouter à la liste

Renvoie

la liste avec le lapin ajouté

5.7.2.2 delete_dead()

```
list_rabbit_t delete_dead (
            list_rabbit_t list_rabbit )
```

supprime tout les lapins morts de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

la liste avec les lapins morts supprimés

5.7.2.3 delete_head()

```
list_rabbit_t delete_head (
            list_rabbit_t list_rabbit )
```

supprime le premier lapin de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

la liste avec le lapin supprimé

5.7.2.4 display_death()

```
void display_death (
            list_rabbit_t list_rabbit )
```

affiche la valeur de la variable bouléene alive (pour debug)

Paramètres

list_rabbit	liste de lapins

5.7.2.5 display_rabbit()

```
void display_rabbit (
            rabbit_t rabbit )
```

affiche les informations d'un lapin

Paramètres

rabbit	un lapins

5.7.2.6 display_rabbit_all()

```
void display_rabbit_all (
            list_rabbit_t list_rabbit )
```

affiche les informations de tous les lapins de la liste

Paramètres

list_rabbit	liste de lapins

5.7.2.7 empty()

```
Boolean empty (
            list_rabbit_t list_rabbit )
```

permet de savoir si la liste est vide

Paramètres

list_rabbit	liste de lapins

Renvoie

true si la liste est vide, false sinon

5.7.2.8 head()

```
rabbit_t head (
            list_rabbit_t list_rabbit )
```

permet d'obtenir le premier élement de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

le premier lapin de la liste

5.7.2.9 new_list()

```
list_rabbit_t new_list (
            void  )
```

créer une liste vide

Renvoie

NULL

5.7.2.10 size_list()

```
int size_list (
            list_rabbit_t list_rabbit )
```

compte le nombre d'éléments de la liste

Paramètres

list_rabbit	liste de lapins

Renvoie

le nombre d'éléments de la liste

5.8 Référence du fichier rabbitTreeStatus.c

Programmes pour réaliser une simulation stocastique 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 copyRabitTab (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 lenCumul↩ ProbaLitter, 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)

5.8.1 Description détaillée

Programmes pour réaliser une simulation stocastique 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

5.8.2 Documentation des fonctions

5.8.2.1 addBabys()

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

fonction pour ajouter les lapins a la liste des lapins

Paramètres

babys	nombre de bébés a ajouter
rabbit	tableaux d'états de lapins

Renvoie

void

5.8.2.2 CalcSumYears()

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

fonction pour calculer la somme des lapins

Paramètres

tab	tableau
maxOld	age max des lapins

Renvoie

la somme des lapins

5.8.2.3 checkMallocNull()

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

vérifi si un malloc c'est bien passé

Paramètres

tab	tableau

Renvoie

void

5.8.2.4 copyRabitTab()

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

fonction pour copier des tableaux d'états de lapin

Paramètres

src	tableau source
dest	tableau de destination
maxRabbitYear	l'age maximum d'un lapin

Renvoie

void

5.8.2.5 createRabbitTabZero()

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

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

Paramètres

maxRabbitYear	l'age maximum d'un lapin

Renvoie

un tableau de tableaux de unsigned long

5.8.2.6 free_double_tab()

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

fonction pour free un tableau de tableaux de unsigned long

Paramètres

tab	tableau
n	nombre d'éléments dans le tableau

Renvoie

void

5.8.2.7 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

old	age du lapin
cumulProbaLitter	tableau de probabilité cumulé pour le nombre de portée
cumulProbaBabys	tableau de probabilité cumulé pour le nombre de bébés par portée
lenCumulProbaLitter	longeur du tableau de proba de portée
lenCumulProbaBabys	longeur du tableau de proba de bébés
minLitter	nombre minimum de portées
minBaby	nombre minimum de bébés

Renvoie

1 ou 0

5.8.2.8 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

config	configuration de la simulation
rabbits	tableau d'états de lapins
rabbitsTmp	tableau d'états de lapins temporaire
year	année de l'iteration
cumulProbaLitter	tableau de probabilité cumulé pour le nombre de portée
cumulProbaBabys	tableau de probabilité cumulé pour le nombre de bébés par portée
lenCumulProbaLitter	longeur du tableau de proba de portée
lenCumulProbaBabys	longeur du tableau de proba de bébés

Renvoie

nombre de bébés à ajouter

5.8.2.9 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

config	configuration de la simulation

Renvoie

nombre de lapin final

5.8.2.10 sum_array()

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

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

Paramètres

array	tableau
n	nombre d'éléments dans le tableau

Renvoie

somme

5.8.2.11 surviveRabbitYear()

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

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

Paramètres

old	age du lapin
config	configuration de la simulation

Renvoie

1 ou 0

5.9 Référence du fichier rabbitTreeStatus.h

Programmes pour réaliser une simulation stocastique 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)

5.9.1 Description détaillée

Programmes pour réaliser une simulation stocastique 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

5.9.2 Documentation des fonctions

5.9.2.1 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

config	configuration de la simulation

Renvoie

nombre de lapin final

5.10 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 génére 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.

5.10.1 Description détaillée

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

Auteur

Aurelien DOUARD

Version

0.1

Date

16 février 2020

5.10.2 Documentation des fonctions

5.10.2.1 calcCumulProba()

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

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

Paramètres

tabProb	tableau remplit avec les probabilité
len	la taile du tableau

Renvoie

un tableau avec la probabilité cumulé de chaque élément

5.10.2.2 discDispLoop()

```
int * discDispLoop (
            double cumulProba[],
            int nbClass,
            int nbExp )
```

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

Paramètres

nbExp	nombre de tirages
nbClass	nombre de classes différentes du dans le tableau
cumulProba	tableau remplit avec la probabilité cumulé pour chaque classes

Renvoie

tableau remplit avec fréquences

5.10.2.3 genRand_discDist()

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

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

Paramètres

nbClass	nombre de classes différentes du dans le tableau
cumulProba	tableau remplit avec la probabilité cumulé pour chaque classes
rand	nombre pseudo-aléatoire

5.10.2.4 probaCalc()

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

Fonction qui calcule la probabilité de chaque élément du tableau.

Paramètres

tab	tableau remplit avec les evenements
len	la taile du tableau

Renvoie

un tableau avec la probabilité de chaque élément

5.11 Référence du fichier reverseDistLow.h

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

```
#include "mt19937ar.h"
#include <stdio.h>
#include <stdlib.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 * 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.
- int genRand_discDist (double rand, double cumulProba[ ], int nbClass)
  Fonction qui génére un nombre qui suit une probabilité donné à partir d'un nombre pseudo-aléatoire.

5.11.1 Description détaillée

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

Auteur

Aurelien DOUARD

Version

0.1

Date

16 février 2020

5.11.2 Documentation des fonctions

5.11.2.1 calcCumulProba()

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

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

Paramètres

tabProb	tableau remplit avec les probabilité
len	la taile du tableau

Renvoie

un tableau avec la probabilité cumulé de chaque élément

5.11.2.2 discDispLoop()

```
int* discDispLoop (
            double cumulProba[],
            int nbClass,
            int nbExp )
```

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

Paramètres

nbExp	nombre de tirages
nbClass	nombre de classes différentes du dans le tableau
cumulProba	tableau remplit avec la probabilité cumulé pour chaque classes

Renvoie

tableau remplit avec fréquences

5.11.2.3 genRand_discDist()

```
int genRand_discDist (
            double rand,
```

```
            double cumulProba[],
            int nbClass )
```

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

Paramètres

nbClass	nombre de classes différentes du dans le tableau
cumulProba	tableau remplit avec la probabilité cumulé pour chaque classes
rand	nombre pseudo-aléatoire

5.11.2.4 probaCalc()

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

Fonction qui calcule la probabilité de chaque élément du tableau.

Paramètres

tab	tableau remplit avec les evenements
len	la taile du tableau

Renvoie

un tableau avec la probabilité de chaque élément

Index