Proyecto para prueba de Doxygen

Generated by Doxygen 1.8.13

Contents

1 File Index

1.1 File List

Here is a list of all files with brief descriptions:

2 File Documentation

2.1 cripto.cc File Reference

```
#include <iostream>
#include <string>
#include <fstream>
#include <vector>
#include "funciones_cripto.h"
```

Include dependency graph for cripto.cc:

2.2 fibonacci_main.cc File Reference

```
#include <iostream>
#include <cstdlib>
#include "fibonacci_sum.h"
```

Include dependency graph for fibonacci_main.cc:

Functions

- int main (int argc, char ∗argv[ ])

  Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

2.2.1 Function Documentation

2.2.1.1 main()

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

Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

Author

F. de Sande

Date

7.nov.2020 Cada nuevo término de la serie de Fibonacci se genera sumando los dos anteriores. Comenzando con 0 y 1, los primeros 10 términos serán: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 Desarrolle en C++ un programa que calcule la suma de todos los términos de valor par de la serie que sean menores que 1000.

See also

https://docs.google.com/document/d/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8h↩
ReVU0KOSUQ/edit?usp=sharing
stoi http://www.cplusplus.com/reference/string/stoi/ An Object Oriented Version of the program:
https://stackoverflow.com/questions/21360694/sum-of-even-fibonacci-numbers-under-100
Main function

Parameters

in	*argc*	Number of command line parameters
in	*argv*	Vector containing (char∗) the parameters

Definition at line 29 of file fibonacci_main.cc.

References fibonacci_sum(), and Usage().

```
29                                  {
30   Usage(argc, argv);
31   std::string limit = argv[1];
32   const size_t kLimit = stoi(limit);
33   std::cout << "Sum: " << fibonacci_sum(kLimit) << std::endl;
34   return 0;
35 }
```

Here is the call graph for this function:

2.3 fibonacci_sum.cc File Reference

```
#include <iostream>
#include <cstdlib>
#include "fibonacci_sum.h"
```

Include dependency graph for fibonacci_sum.cc:

Functions

- void Usage (int argc, char ∗argv[ ])

  Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.
- size_t fibonacci_sum (const size_t kLimit)

  Devuelve el valor de la suma de todos los términos de valor par de la serie de Fibonacci menores que kLimit.

2.3.1 Function Documentation

2.3.1.1 fibonacci_sum()

```
size_t fibonacci_sum (
            const size_t kLimit )
```

Devuelve el valor de la suma de todos los términos de valor par de la serie de Fibonacci menores que kLimit.

Parameters

in	*kLimit.*	Se suman los términos pares menores que kLimit

Returns

La suma de los términos pares menores que kLimit

Definition at line 52 of file fibonacci_sum.cc.

Referenced by main().

```
52                                      {
53   size_t second_to_last{0},  // Second to last term
54          last{1},            // Last term generated
55          new_term;           // New term of the serie
56   size_t long sum{0};        // Accumulated sum of the terms
57
58   do {
59     new_term = last + second_to_last;
60     if (new_term % 2 == 0) {
61       sum += new_term;
62     }
63     // Uncomment for debug: print each new term
64     // std::cout << "Term: " << new_term << std::endl;
65     second_to_last = last;
66     last = new_term;
67   } while (new_term < kLimit);
68   return sum;
69 }
```

Here is the caller graph for this function:

2.3.1.2 Usage()

```
void Usage (
             int argc,
             char * argv[] )
```

Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

Author

F. de Sande

Date

7.nov.2020 Cada nuevo término de la serie de Fibonacci se genera sumando los dos anteriores. Comenzando con 0 y 1, los primeros 10 términos serán: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 Desarrolle en C++ un programa que calcule la suma de todos los términos de valor par de la serie que sean menores que 1000.

See also

https://docs.google.com/document/d/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8h↩
ReVU0KOSUQ/edit?usp=sharing
stoi http://www.cplusplus.com/reference/string/stoi/ An Object Oriented Version of the program:
https://stackoverflow.com/questions/21360694/sum-of-even-fibonacci-numbers-under-100
Muestra el modo de uso correcto del programa En caso de que el uso no sea el correcto, muestra el mensaje y finaliza la ejecución del programa. El programa precisa un único número natural para su ejecución.

Parameters

in	*argc*	Number of command line parameters
in	*argv*	Vector containing (char∗) the parameters

Definition at line 33 of file fibonacci_sum.cc.

Referenced by main().

```
33                                      {
34   if (argc != 2) {
35     std::cout << argv[0] << ": Falta un número natural como parámetro" << std::endl;
36     std::cout << "Pruebe " << argv[0] << " --help para más información" << std::endl;
37     exit(EXIT_SUCCESS);
38   }
39   std::string parameter{argv[1]};
40   if (parameter == "--help") {
41     std::cout << kHelpText << std::endl;
42     exit(EXIT_SUCCESS);
43   }
44 }
```

Here is the caller graph for this function:

2.4 fibonacci_sum.h File Reference

`#include <iostream>`
Include dependency graph for fibonacci_sum.h: This graph shows which files directly or indirectly include this file:

Functions

- void Usage (int argc, char ∗argv[ ])

  Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.
- size_t fibonacci_sum (const size_t kLimit)

  Devuelve el valor de la suma de todos los términos de valor par de la serie de Fibonacci menores que kLimit.

Variables

- const std::string kHelpText

  Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

2.4.1 Function Documentation

2.4.1.1 fibonacci_sum()

```
size_t fibonacci_sum (
            const size_t kLimit )
```

Devuelve el valor de la suma de todos los términos de valor par de la serie de Fibonacci menores que kLimit.

Parameters

in	*kLimit.*	Se suman los términos pares menores que kLimit

Returns

La suma de los términos pares menores que kLimit

Definition at line 52 of file fibonacci_sum.cc.

Referenced by main().

```
52                                          {
53   size_t second_to_last{0},  // Second to last term
54          last{1},          // Last term generated
55          new_term;         // New term of the serie
56   size_t long sum{0};       // Accumulated sum of the terms
57
58   do {
59     new_term = last + second_to_last;
60     if (new_term % 2 == 0) {
61       sum += new_term;
62     }
63     // Uncomment for debug: print each new term
64     // std::cout << "Term: " << new_term << std::endl;
65     second_to_last = last;
66     last = new_term;
67   } while (new_term < kLimit);
68   return sum;
69 }
```

Here is the caller graph for this function:

2.4.1.2 Usage()

```
void Usage (
              int argc,
              char * argv[] )
```

Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

Author

F. de Sande

Date

7.nov.2020 Cada nuevo término de la serie de Fibonacci se genera sumando los dos anteriores. Comenzando con 0 y 1, los primeros 10 términos serán: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 Desarrolle en C++ un programa que calcule la suma de todos los términos de valor par de la serie que sean menores que 1000.

See also

https://docs.google.com/document/d/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8h↩
ReVU0KOSUQ/edit?usp=sharing
stoi http://www.cplusplus.com/reference/string/stoi/ An Object Oriented Version of the program:
https://stackoverflow.com/questions/21360694/sum-of-even-fibonacci-numbers-under-100
Muestra el modo de uso correcto del programa En caso de que el uso no sea el correcto, muestra el mensaje y finaliza la ejecución del programa. El programa precisa un único número natural para su ejecución.

Parameters

in	*argc*	Number of command line parameters
in	*argv*	Vector containing (char∗) the parameters

Definition at line 33 of file fibonacci_sum.cc.

```
33                                    {
34   if (argc != 2) {
35     std::cout << argv[0] << ": Falta un número natural como parámetro" << std::endl;
36     std::cout << "Pruebe " << argv[0] << " --help para más información" << std::endl;
37     exit(EXIT_SUCCESS);
38   }
39   std::string parameter{argv[1]};
40   if (parameter == "--help") {
41     std::cout << kHelpText << std::endl;
42     exit(EXIT_SUCCESS);
43   }
44 }
```

2.4.2 Variable Documentation

2.4.2.1 kHelpText

```
const std::string kHelpText
```

Initial value:

```
= "Este programa calcula la suma de todos los términos pares de la \
serie de Fibonacci que sean menores que un valor, que el usuario \
ha de introducir por línea de comandos para la ejecución del programa"
```

Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

Author

F. de Sande

Date

7.nov.2020 This file declares the "Help Text" constant and two functions

Definition at line 15 of file fibonacci_sum.h.

Referenced by Usage().

2.5 funciones_cripto.cc File Reference

```
#include <iostream>
#include <string>
#include <cstring>
#include "funciones_cripto.h"
```

Include dependency graph for funciones_cripto.cc:

Functions

- void Usage (int argc, char ∗argv[ ])

  Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.
- string XOR (string texto)
- string CesarEncriptar (string texto, int clave)
- string CesarDesencriptar (string texto, int clave)

2.5.1 Function Documentation

2.5.1.1 CesarDesencriptar()

```
string CesarDesencriptar (
              string texto,
              int clave )
```

Definition at line 70 of file funciones_cripto.cc.

Referenced by main().

```
70                                                {
71   string resultado = "";
72   for(int i = 0; i <texto.length(); i++){
73     if(isalpha(texto[i])){
74       if(isupper(texto[i])){
75       resultado = resultado + char((int(texto[i]- clave - 'A') % 26) + 'A');
76     } else {
77       resultado = resultado + char((int(texto[i]- clave - 'a') % 26) + 'a');
78       }
79     }
80   }
81   return resultado;
82 }
```

Here is the caller graph for this function:

2.5.1.2 CesarEncriptar()

```
string CesarEncriptar (
              string texto,
              int clave )
```

Definition at line 56 of file funciones_cripto.cc.

Referenced by main().

```
56                                                {
57   string resultado = "";
58   for(int i = 0; i <texto.length(); i++){
59       if(isalpha(texto[i])){
60         if(isupper(texto[i])){
61           resultado = resultado + char((int(texto[i]+ clave - 'A') % 26) + 'A');
62         }   else {
63           resultado = resultado + char((int(texto[i]+ clave - 'a') % 26) + 'a');
64             }
65       }
66   }
67   return resultado;
68 }
```

Here is the caller graph for this function:

2.5.1.3 Usage()

```
void Usage (
            int argc,
            char * argv[] )
```

Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

Author

F. de Sande

Date

7.nov.2020 Cada nuevo término de la serie de Fibonacci se genera sumando los dos anteriores. Comenzando con 0 y 1, los primeros 10 términos serán: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 Desarrolle en C++ un programa que calcule la suma de todos los términos de valor par de la serie que sean menores que 1000.

See also

https://docs.google.com/document/d/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8h↩ ReVU0KOSUQ/edit?usp=sharing
stoi http://www.cplusplus.com/reference/string/stoi/ An Object Oriented Version of the program:
https://stackoverflow.com/questions/21360694/sum-of-even-fibonacci-numbers-under-100
Muestra el modo de uso correcto del programa En caso de que el uso no sea el correcto, muestra el mensaje y finaliza la ejecución del programa. El programa precisa un único número natural para su ejecución.

Parameters

in	*argc*	Number of command line parameters
in	*argv*	Vector containing (char∗) the parameters

Definition at line 23 of file funciones_cripto.cc.

References kHelpText.

```
23                                  {
24   if(argc <= 1){
25     cout << argv[0] << " -- Cifrado de ficheros" << endl;
26     cout << "Modo de uso: " << argv[0] << " fichero_entrada fichero_salida método password operación" <<
      endl;
27     cout << "Pruebe " << argv[0] << " --help para más información" << endl;
28     exit(EXIT_SUCCESS);
29   }
30   string parametro{argv[1]};
31   if(parametro == "--help"){
32     cout << argv[0] << " -- Cifrado de ficheros" << endl;
33     cout << "Modo de uso: " << argv[0] << " fichero_entrada fichero_salida método password operación" <<
      endl << endl;
34     cout << "fichero_entrada: Fichero a codificar" << endl;
35     cout << "fichero_salida:  Fichero codificaco" << endl;
36     cout << "método:          Indica el método de encriptado" << endl;
37     cout << "                   1: Cifrado XOR" << endl;
38     cout << "                   2: Cifrado César" << endl;
39     cout << "password:        Letra (mayúscula) secreta en el caso del método 1, Valor de K en el método 2"
      << endl;
40     cout << "operación:       Operación a realizar en el fichero" << endl;
```

```
41     cout << "                   +: encriptar el fichero" << endl;
42     cout << "                   -: desencariptar el fichero" << endl;
43     exit(EXIT_SUCCESS);
44   }
45 }
```

2.5.1.4 XOR()

```
string XOR (
             string texto )
```

Definition at line 47 of file funciones_cripto.cc.

Referenced by main().

```
47                          {
48     char clave = 'F';                  //Solo funciona con letras mayúsculas y no usa la clave introducida
       por línea de comandos.
49     string salida = texto;
50
51     for (int i = 0; i < salida.size(); i++)
52         salida[i] = salida[i] ^ clave;
53     return salida;
54 }
```

Here is the caller graph for this function:

2.6 funciones_cripto.h File Reference

```
#include <iostream>
#include <string>
```

Include dependency graph for funciones_cripto.h: This graph shows which files directly or indirectly include this file:

Functions

- void Usage (int argc, char *argv[])

  Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.
- string XOR (string texto)
- string CesarEncriptar (string texto, int clave)
- string CesarDesencriptar (string texto, int clave)

2.6.1 Function Documentation

2.6.1.1 CesarDesencriptar()

```
string CesarDesencriptar (
             string texto,
             int clave )
```

Definition at line 70 of file funciones_cripto.cc.

Referenced by main().

```
70                                                {
71   string resultado = "";
72   for(int i = 0; i <texto.length(); i++){
73     if(isalpha(texto[i])){
74       if(isupper(texto[i])){
75       resultado = resultado + char((int(texto[i]- clave - 'A') % 26) + 'A');
76     } else {
77       resultado = resultado + char((int(texto[i]- clave - 'a') % 26) + 'a');
78       }
79     }
80   }
81   return resultado;
82 }
```

Here is the caller graph for this function:

2.6.1.2 CesarEncriptar()

```
string CesarEncriptar (
             string texto,
             int clave )
```

Definition at line 56 of file funciones_cripto.cc.

Referenced by main().

```
56                                                {
57   string resultado = "";
58   for(int i = 0; i <texto.length(); i++){
59       if(isalpha(texto[i])){
60         if(isupper(texto[i])){
61           resultado = resultado + char((int(texto[i]+ clave - 'A') % 26) + 'A');
62         }   else {
63           resultado = resultado + char((int(texto[i]+ clave - 'a') % 26) + 'a');
64             }
65       }
66   }
67   return resultado;
68 }
```

Here is the caller graph for this function:

2.6.1.3 Usage()

```
void Usage (
             int argc,
             char * argv[] )
```

Universidad de La Laguna Escuela Superior de Ingeniería y Tecnología Grado en Ingeniería Informática Informática Básica.

Author

F. de Sande

Date

7.nov.2020 Cada nuevo término de la serie de Fibonacci se genera sumando los dos anteriores. Comenzando con 0 y 1, los primeros 10 términos serán: 0, 1, 1, 2, 3, 5, 8, 13, 21, 34 Desarrolle en C++ un programa que calcule la suma de todos los términos de valor par de la serie que sean menores que 1000.

See also

https://docs.google.com/document/d/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8h↩
ReVU0KOSUQ/edit?usp=sharing
stoi http://www.cplusplus.com/reference/string/stoi/ An Object Oriented Version of the program:
https://stackoverflow.com/questions/21360694/sum-of-even-fibonacci-numbers-under-100
Muestra el modo de uso correcto del programa En caso de que el uso no sea el correcto, muestra el mensaje y finaliza la ejecución del programa. El programa precisa un único número natural para su ejecución.

Parameters

in	*argc*	Number of command line parameters
in	*argv*	Vector containing (char∗) the parameters

Definition at line 33 of file fibonacci_sum.cc.

References kHelpText.

Referenced by main().

```
33                                    {
34   if (argc != 2) {
35     std::cout << argv[0] << ": Falta un número natural como parámetro" << std::endl;
36     std::cout << "Pruebe " << argv[0] << " --help para más información" << std::endl;
37     exit(EXIT_SUCCESS);
38   }
39   std::string parameter{argv[1]};
40   if (parameter == "--help") {
41     std::cout << kHelpText << std::endl;
42     exit(EXIT_SUCCESS);
43   }
44 }
```

Here is the caller graph for this function:

2.6.1.4 XOR()

```
string XOR (
             string texto )
```

Definition at line 47 of file funciones_cripto.cc.

Referenced by main().

```
47                          {
48     char clave = 'F';                  //Solo funciona con letras mayúsculas y no usa la clave introducida
       por línea de comandos.
49     string salida = texto;
50
51     for (int i = 0; i < salida.size(); i++)
52         salida[i] = salida[i] ^ clave;
53     return salida;
54 }
```

Here is the caller graph for this function:

Index