

Proyecto para prueba de Doxygen

Generated by Doxygen 1.9.1

<b>1 File Index</b>	<b>1</b>
<b>1 File Index</b>	<b>1</b>
1.1 File List . . . . .	1
<b>2 File Documentation</b>	<b>1</b>
2.1 fibonacci_main.cc File Reference . . . . .	1
2.2 fibonacci_sum.cc File Reference . . . . .	1
2.2.1 Function Documentation . . . . .	2
2.3 fibonacci_sum.h File Reference . . . . .	3
2.3.1 Function Documentation . . . . .	3
<b>Index</b>	<b>7</b>

## 1 File Index

### 1.1 File List

Here is a list of all files with brief descriptions:

<a href="#">fibonacci_main.cc</a>	<b>1</b>
<a href="#">fibonacci_sum.cc</a>	<b>1</b>
<a href="#">fibonacci_sum.h</a>	<b>3</b>

## 2 File Documentation

### 2.1 fibonacci\_main.cc File Reference

```
#include <iostream>
#include <cstdlib>
#include "fibonacci_sum.h"
Include dependency graph for fibonacci_main.cc:
```

### 2.2 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 [FibonacciSum](#) (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.*

## 2.2.1 Function Documentation

### 2.2.1.1 FibonacciSum() `size_t FibonacciSum (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	<i>kLimit</i> .	Se suman los términos pares menores que kLimit
----	-----------------	--

#### Returns

La suma de los términos pares menores que kLimit

Definition at line 55 of file fibonacci\_sum.cc.

```

55 {
56     size_t second_to_last{0}, // Second to last term
57         last{1},             // Last term generated
58         new_term;             // New term of the serie
59     size_t long sum{0};        // Accumulated sum of the terms
60
61     do {
62         new_term = last + second_to_last;
63         if (new_term % 2 == 0) {
64             sum += new_term;
65         }
66         // Uncomment for debug: print each new term
67         // std::cout << "Term: " << new_term << std::endl;
68         second_to_last = last;
69         last = new_term;
70     } while (new_term < kLimit);
71     return sum;
72 }
```

Referenced by main().

Here is the caller graph for this function:

### 2.2.1.2 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/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8h4ReVU0KOSUQ/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-10>

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	<i>argc</i>	Number of command line parameters
in	<i>argv</i>	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         const std::string kHelpText = "Este programa calcula la suma de todos los términos pares de la \
42 serie de Fibonacci que sean menores que un valor, que el usuario \
43 ha de introducir por línea de comandos para la ejecución del programa";
44         std::cout << kHelpText << std::endl;
45         exit(EXIT_SUCCESS);
46     }
47 }
```

Referenced by main().

Here is the caller graph for this function:

## 2.3 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 [FibonacciSum](#) (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 FibonacciSum()** size\_t FibonacciSum (  
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	<i>kLimit</i> .	Se suman los términos pares menores que kLimit
----	-----------------	--

#### Returns

La suma de los términos pares menores que kLimit

Definition at line 55 of file fibonacci\_sum.cc.

```
55                                     {
56     size_t second_to_last{0}, // Second to last term
57         last{1},             // Last term generated
58         new_term;             // New term of the serie
59     size_t long sum{0};        // Accumulated sum of the terms
60
61     do {
62         new_term = last + second_to_last;
63         if (new_term % 2 == 0) {
64             sum += new_term;
65         }
66         // Uncomment for debug: print each new term
67         // std::cout << "Term: " << new_term << std::endl;
68         second_to_last = last;
69         last = new_term;
70     } while (new_term < kLimit);
71     return sum;
72 }
```

Referenced by main().

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

This file declares the "Help Text" constant and two functions

#### 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/1-3hTIVf8tPrbn9u0vs0Cm2IGyX1XBgv8hReVU0KOSUQ/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-10>

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	<i>argc</i>	Number of command line parameters
in	<i>argv</i>	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         const std::string kHelpText = "Este programa calcula la suma de todos los términos pares de la \
42 serie de Fibonacci que sean menores que un valor, que el usuario \
43 ha de introducir por línea de comandos para la ejecución del programa";
44         std::cout << kHelpText << std::endl;
45         exit(EXIT_SUCCESS);
46     }
47 }
```

Referenced by main().

Here is the caller graph for this function:

## Index

- fibonacci\_main.cc, [1](#)
- fibonacci\_sum.cc, [1](#)
  - FibonacciSum, [2](#)
  - Usage, [2](#)
- fibonacci\_sum.h, [3](#)
  - FibonacciSum, [3](#)
  - Usage, [4](#)
- FibonacciSum
  - fibonacci\_sum.cc, [2](#)
  - fibonacci\_sum.h, [3](#)
- Usage
  - fibonacci\_sum.cc, [2](#)
  - fibonacci\_sum.h, [4](#)