

Proyecto para prueba de Doxygen

Generated by Doxygen 1.8.17

<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
2.1.1 Function Documentation . . . . .	2
2.2 fibonacci_sum.cc File Reference . . . . .	3
2.2.1 Function Documentation . . . . .	4
2.3 fibonacci_sum.h File Reference . . . . .	6
2.3.1 Function Documentation . . . . .	7
2.3.2 Variable Documentation . . . . .	9
<b>Index</b>	<b>11</b>

# 1 File Index

## 1.1 File List

Here is a list of all files with brief descriptions:

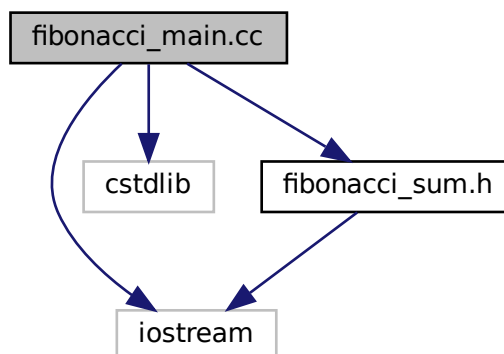
<a href="#">fibonacci_main.cc</a>	<b>1</b>
<a href="#">fibonacci_sum.cc</a>	<b>3</b>
<a href="#">fibonacci_sum.h</a>	<b>6</b>

## 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:



## Functions

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

*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.*

### 2.1.1 Function Documentation

**2.1.1.1 main()**

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

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.

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

Author

F. de Sande

Date

7.nov.2020

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:

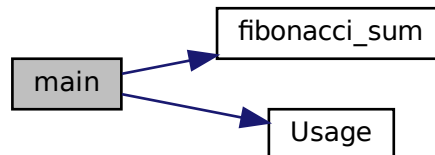
<https://stackoverflow.com/questions/21360694/sum-of-even-fibonacci-numbers-under-10>  
exit

Definition at line 25 of file fibonacci\_main.cc.

```
25     {  
26         Usage(argc, argv);  
27         std::string limit = argv[1];  
28         const size_t kLimit = stoi(limit);  
29         std::cout << "Sum: " << fibonacci_sum(kLimit) << std::endl;  
30         return 0;  
31     }
```

References fibonacci\_sum(), and Usage().

Here is the call graph for this function:



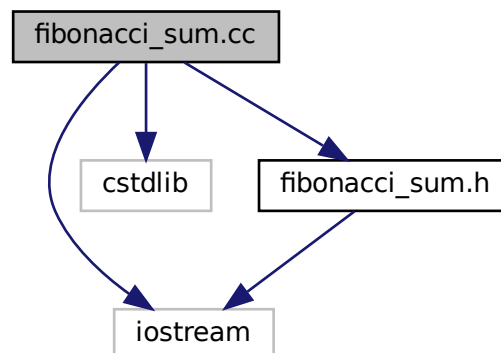
## 2.2 fibonacci\_sum.cc File Reference

```
#include <iostream>
```

```
#include <cstdlib>
```

```
#include "fibonacci_sum.h"
```

Include dependency graph for fibonacci\_sum.cc:



## Functions

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

*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.*

- `size_t fibonacci_sum (const size_t kLimit)`

### 2.2.1 Function Documentation

#### 2.2.1.1 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	<i>kLimit.</i>	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.

```
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 }
```

Referenced by main().

Here is the caller graph for this function:



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

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.

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

#### Author

F. de Sande

#### Date

7.nov.2020

#### 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         std::cout << kHelpText << std::endl;
42         exit(EXIT_SUCCESS);
43     }
44 }
```

References kHelpText.

Referenced by main().

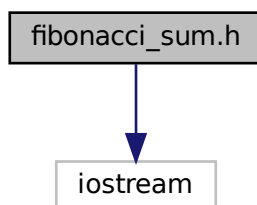
Here is the caller graph for this function:



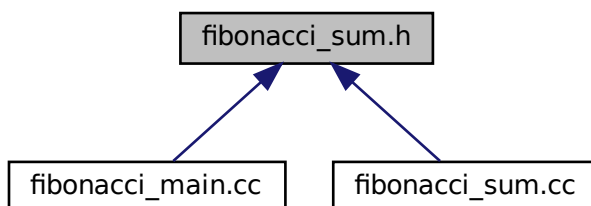
## 2.3 fibonacci\_sum.h File Reference

```
#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[])

*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.*

- size\_t `fibonacci_sum` (const size\_t kLimit)

## Variables

- const std::string [kHelpText](#)

*Definitions.*

### 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	<i>kLimit.</i>	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.

```
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 }
```

Referenced by main().

Here is the caller graph for this function:





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

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.

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

#### Author

F. de Sande

#### Date

7.nov.2020

#### 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] << ": Faltan 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 }
```

References kHelpText.

Referenced by main().

Here is the caller graph for this function:



## 2.3.2 Variable Documentation

### 2.3.2.1 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"
```

Definitions.

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

#### Author

F. de Sande

#### Date

7.nov.2020

Definition at line 15 of file fibonacci\_sum.h.

Referenced by Usage().



## Index

- fibonacci\_main.cc, [1](#)
  - main, [2](#)
- fibonacci\_sum
  - fibonacci\_sum.cc, [4](#)
  - fibonacci\_sum.h, [7](#)
- fibonacci\_sum.cc, [3](#)
  - fibonacci\_sum, [4](#)
  - Usage, [4](#)
- fibonacci\_sum.h, [6](#)
  - fibonacci\_sum, [7](#)
  - kHelpText, [9](#)
  - Usage, [7](#)
- kHelpText
  - fibonacci\_sum.h, [9](#)
- main
  - fibonacci\_main.cc, [2](#)
- Usage
  - fibonacci\_sum.cc, [4](#)
  - fibonacci\_sum.h, [7](#)