Reference Manual

Generated by Doxygen 1.8.13

Contents

1 Test List

File insertion.cpp

inserti

```
15 77 83 86 93
"
==
"83 86 77 15 93
15 77 83 86 93
"
Passed 1 test case with 1 assertion.
```

2 File Index

2.1 File List

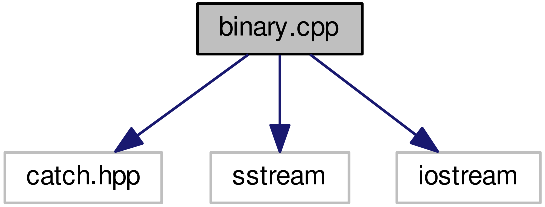
Here is a list of all files with brief descriptions:

3 File Documentation

3.1 binary.cpp File Reference

```
#include "catch.hpp"
#include <sstream>
#include <iostream>
```

Include dependency graph for binary.cpp:



Functions

- int binary_search (int a[ ], int from, int to, int value)
- TEST_CASE ("Binary Search")

3.1.1 Function Documentation

3.1.1.1 binary_search()

```
int binary_search (
            int a[],
            int from,
            int to,
            int value )
```

Finds an element in a sorted array.

Parameters

a	the sorted array with the elements to search
from	the start of the range to search
to	the end of the range to search
value	the value to search for

Returns

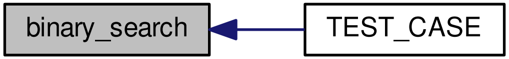
the index of the first match, or -1 if not found

Definition at line 15 of file binary.cpp.

```
{
   if (from > to)
   {
      return -1;
   }

   int mid = (from + to) / 2;
   if (a[mid] == value)
   {
      return mid;
   }
   else if (a[mid] < value)
   {
      return binary_search(a, mid + 1, to, value);
   }
   else
   {
      return binary_search(a, from, mid - 1, value);
   }
}
```

Here is the caller graph for this function:

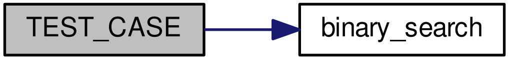


3.1.1.2 TEST_CASE()

```
TEST_CASE (
             "Binary Search" )
```

Definition at line 37 of file binary.cpp.

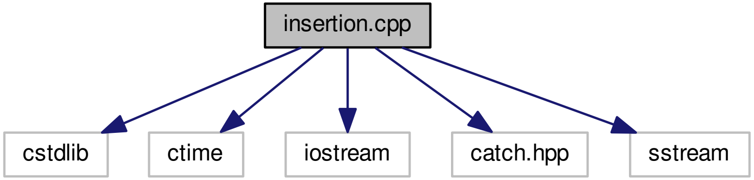
Here is the call graph for this function:



3.2 insertion.cpp File Reference

```
#include <cstdlib>
#include <ctime>
#include <iostream>
#include "catch.hpp"
#include <sstream>
```

Include dependency graph for insertion.cpp:



Functions

- void insertion_sort (int a[ ], int size)
- void print (int a[ ], int size)
- TEST_CASE ("Insertion")

3.2.1 Detailed Description

Test

```
insertion.cpp:68: passed: ss.str() == "83 86 77 15 93 \n15 77 83 86 93 \n" for: "83 86 77 15 93
15 77 83 86 93
"
==
"83 86 77 15 93
15 77 83 86 93
"
Passed 1 test case with 1 assertion.
```

3.2.2 Function Documentation

3.2.2.1 insertion_sort()

```
void insertion_sort (
            int a[],
            int size )
```

Sorts an array, using insertion sort.

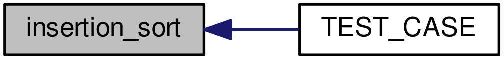
Parameters

a	the array to sort

Definition at line 14 of file insertion.cpp.

```
{
    for (int i = 1; i < size; i++)
    {
        int next = a[i];
        // Move all larger elements up
        int j = i;
        while (j > 0 && a[j - 1] > next)
        {
            a[j] = a[j - 1];
            j--;
        }
        // Insert the element
        a[j] = next;
    }
}
```

Here is the caller graph for this function:



3.2.2.2 print()

```
void print (
            int a[],
            int size )
```

Prints all elements in an array.

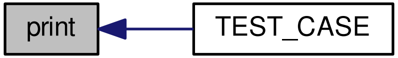
Parameters

a	the array to print
size	the number of elements in a

Definition at line 36 of file insertion.cpp.

```
{
    for (int i = 0; i < size; i++)
    {
        cout << a[i] << " ";
    }
    cout << endl;
}
```

Here is the caller graph for this function:



3.2.2.3 TEST_CASE()

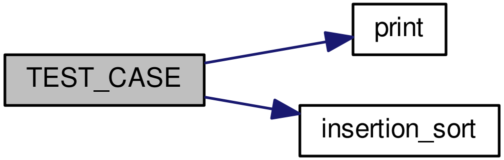
```
TEST_CASE (
             "Insertion"  )
```

Definition at line 51 of file insertion.cpp.

```
{
   const int SIZE = 5;
   int values[SIZE];
   for (int i = 0; i < SIZE; i++)
   {
      values[i] = rand() % 100;
   }
   std::streambuf *b = std::cout.rdbuf(); std::stringstream ss;
    std::streambuf *sb = ss.rdbuf(); std::cout.rdbuf(sb);
    // Now all output will be redirected into ss
   print(values, SIZE);
   insertion_sort(values, SIZE);
   print(values, SIZE);
   // set output back to the terminal
   std::cout.rdbuf(b);

   CHECK(ss.str() == "83 86 77 15 93 \n15 77 83 86 93 \n");
   //return 0;
}
```

Here is the call graph for this function:



Index