IN505 C++ Project - Huffman Coding

0.3.0

Generated by Doxygen 1.8.13

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

File Index

3.1 File List

Here is a list of all files with brief descriptions:

Chapter 4

Class Documentation

4.1 Sommet Class Reference

The class `Sommet` represents a node of the class `ArbreB`.

`#include <Sommet.hpp>`

Public Member Functions

- Sommet ()
- Sommet (const char &data, const double &freq)
- Sommet (const Sommet &other)
- ~Sommet ()
- Sommet & operator= (const Sommet &other)
- char & get_data ()
- double & get_freq ()
- Sommet * get_left ()
- Sommet * get_right ()
- void set_data (const char &data)
- void set_freq (const double &freq)
- void print (int spacing)

Friends

- class ArbreB

4.1.1 Detailed Description

The class `Sommet` represents a node of the class `ArbreB`.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020/11/17

4.1.2 Constructor & Destructor Documentation

4.1.2.1 Sommet() [1/3]

```
Sommet::Sommet ( )
```

Creates a default object Sommet. Sets m_Data to \0, m_Freq to 0, m_Left and m_Right to nullptr.

4.1.2.2 Sommet() [2/3]

```
Sommet::Sommet (
            const char & data,
            const double & freq )
```

Creates an object Sommet with the specified parameters.

Parameters

data	The character to store.
freq	The frequency of the stored character.

4.1.2.3 Sommet() [3/3]

```
Sommet::Sommet (
            const Sommet & other )
```

Creates a copy of the specified object Sommet.

Parameters

other	The Sommet to copy.

4.1.2.4 ~Sommet()

```
Sommet::~Sommet ( )
```

Frees the memory for of an object Sommet.

4.1.3 Member Function Documentation

4.1.3.1 get_data()

```
char& Sommet::get_data ( )
```

Gets the character.

Returns

A reference of the character.

4.1.3.2 get_freq()

```
double& Sommet::get_freq ( )
```

Gets the character's frequency.

Returns

A reference of the character's frequency.

4.1.3.3 get_left()

```
Sommet* Sommet::get_left ( )
```

Gets the left child.

Returns

A reference of the left child.

4.1.3.4 get_right()

```
Sommet* Sommet::get_right ( )
```

Gets the right child.

Returns

A reference of the right child.

4.1.3.5 operator=()

```
Sommet& Sommet::operator= (
            const Sommet & other )
```

Redefines the behavior of the operator =.

Parameters

other	The object `Sommet` to assign the values from.

Returns

A reference to a copy of `other`.

4.1.3.6 print()

```
void Sommet::print (
            int spacing )
```

Prints an ArbreB.

4.1.3.7 set_data()

```
void Sommet::set_data (
            const char & data )
```

Sets the value of the character.

Parameters

data	The character to assign to `m_Data`.

4.1.3.8 set_freq()

```
void Sommet::set_freq (
            const double & freq )
```

Sets the value of the frequency.

Parameters

freq	The value to assign to `m_Freq`.

4.1.4 Friends And Related Function Documentation

4.1.4.1 ArbreB

```
friend class ArbreB  [friend]
```

The documentation for this class was generated from the following file:

- src/headers/Sommet.hpp

4.2 ArbreB Class Reference

The class `ArbreB` represents a binary tree.

```
#include <ArbreB.hpp>
```

Public Member Functions

- ArbreB ()
- ArbreB (const char &data, const double &freq)
- ArbreB (const Sommet &node)
- ArbreB (const ArbreB &other)
- ~ArbreB ()
- ArbreB & operator= (const ArbreB &other)
- void insert (Sommet &new_node)
- void insert (const char &data, const double &freq)
- bool search (const char &data)
- bool bst_search (const char &data, std::string &path)
- ArbreB & remove (const char &data)
- ArbreB operator+ (const ArbreB &other)
- std::tuple< ArbreB, ArbreB > decompose ()
- Sommet * get_root ()
- void print ()
- std::map< char, std::string > build_huffman_map ()

Friends

- std::ostream & operator<< (std::ostream &stream, ArbreB &tree)

4.2.1 Detailed Description

The class `ArbreB` represents a binary tree.

Author

Gabriel Dos Santos

Version

0.2.0

Date

2020/11/17

4.2.2 Constructor & Destructor Documentation

4.2.2.1 ArbreB() [1/4]

```
ArbreB::ArbreB ( )
```

Creates a default object ArbreB. Sets m_Root at nullptr.

4.2.2.2 ArbreB() [2/4]

```
ArbreB::ArbreB (
            const char & data,
            const double & freq )
```

Creates an object ArbreB from the specified parameters.

Parameters

data	The character to store in the root of the ArbreB.
freq	The character's frequency to store in the root of the ArbreB.

4.2.2.3 ArbreB() [3/4]

```
ArbreB::ArbreB (
            const Sommet & node )
```

Creates an object ArbreB from the specified Sommet.

Parameters

node	The Sommet to initialize m_Root from.

4.2.2.4 ArbreB() [4/4]

```
ArbreB::ArbreB (
            const ArbreB & other )
```

Creates a copy of the specified ArbreB.

Parameters

other	The ArbreB to copy.

4.2.2.5 ~ArbreB()

```
ArbreB::~ArbreB ( )
```

Frees the memory of an ArbreB.

4.2.3 Member Function Documentation

4.2.3.1 bst_search()

```
bool ArbreB::bst_search (
            const char & data,
            std::string & path )
```

Searches for the specified character in the object ArbreB. Internaly calls private method `bst_search()`.

Parameters

data	The character to search for.
path	A string that stores the path to the character. '0's mean the path takes a left branch, '1's means it takes a right branch.

Returns

`True` if the character was found, `False` otherwise.

4.2.3.2 build_huffman_map()

```
std::map<char, std::string> ArbreB::build_huffman_map ( )
```

Returns a map holding each character in the ArbreB and its binary code. Internaly calls private method `map_char_to_code()`.

Returns

The map holding the characters and their encoding.

4.2.3.3 decompose()

```
std::tuple<ArbreB, ArbreB> ArbreB::decompose ( )
```

Decomposes one object ArbreB into two.

Returns

An std::tuple that holds two ArbreB. The first one is the left branch of the original tree. The second one is the right branch of the original tree.

4.2.3.4 get_root()

```
Sommet* ArbreB::get_root ( )
```

Returns a pointer to the root of the ArbreB.

Returns

The pointer on the root.

4.2.3.5 insert() [1/2]

```
void ArbreB::insert (
            Sommet & new_node )
```

Inserts a new node in the object ArbreB. Internaly calls private method `insert()`.

Parameters

new_node	The Sommet to insert in the ArbreB.

4.2.3.6 insert() [2/2]

```
void ArbreB::insert (
            const char & data,
            const double & freq )
```

Inserts a new node in the object ArbreB. Internaly calls private method `insert()`.

Parameters

data	The character to insert in the ArbreB.
freq	The frequency of the character to insert.

4.2.3.7 operator+()

```
ArbreB ArbreB::operator+ (
            const ArbreB & other )
```

Overloads the operator + to redefine its behavior. Fuses two ArbreB together to creates a new one. Sets m_Root->m_Left as this, m_Root->m_Right as other. Sets m_Root->m_Data as \0, m_Root->m_Freq as this.m_Root->m_Freq + other.m_Root->m_Freq.

Parameters

other	The ArbreB to fuse.

Returns

The fusion of the two ArbreBs.

4.2.3.8 operator=()

```
ArbreB& ArbreB::operator= (
            const ArbreB & other )
```

Overloads the operator = to redefine its behavior.

Parameters

other	The ArbreB to assign the values from.

Returns

A reference to a copy of other.

4.2.3.9 print()

```
void ArbreB::print ( )
```

Prints an ArbreB.

4.2.3.10 remove()

```
ArbreB& ArbreB::remove (
            const char & data )
```

Removes a Sommet from the object ArbreB. Internaly calls private method remove().

Parameters

data	The character to delete.

Returns

A reference of the ArbreB with the removed Sommet.

4.2.3.11 search()

```
bool ArbreB::search (
            const char & data )
```

Searches for the specified character in the object ArbreB. Internaly calls private method `search()`.

Parameters

data	The character to search for.

Returns

`True` if the character was found, `False` otherwise.

4.2.4 Friends And Related Function Documentation

4.2.4.1 operator<<

```
std::ostream& operator<< (
            std::ostream & stream,
            ArbreB & tree )  [friend]
```

Overloads the operator >> and redefines its behavior.

Parameters

stream	The output stream.
tree	The ArbreB.

Returns

The output stream to print to `std::cout`.

The documentation for this class was generated from the following file:

- src/headers/ArbreB.hpp

4.3 Part1 Class Reference

The class `Part1` implements tests to assert that the functions in classes `Sommet` and `ArbreB` have the expected behavior.

`#include <Part1.hpp>`

Public Member Functions

- Part1 ()
- bool should_create_default_Sommet ()
- bool should_create_parameterized_Sommet ()
- bool should_create_copy_Sommet ()
- bool should_set_Sommet_values ()
- bool should_assign_Sommet ()
- bool should_not_link_Sommet_copies ()
- bool should_create_default_ArbreB ()
- bool should_create_parameterized_ArbreB ()
- bool should_create_ArbreB_from_Sommet ()
- bool should_create_copy_ArbreB ()
- bool should_assign_ArbreB ()
- bool should_not_link_ArbreB_copies ()
- bool should_insert_Sommet_into_ArbreB ()
- bool should_update_freq_if_char_already_in_ArbreB ()
- bool should_find_character_c ()
- bool should_not_find_character_z ()
- bool should_remove_leaf ()
- bool should_remove_Sommet_with_one_child ()
- bool should_remove_Sommet_with_two_children ()
- bool should_fuse_two_ArbreB ()
- bool should_decompose_one_ArbreB_into_two ()
- bool should_find_character_y_with_bfs ()
- bool should_not_find_character_s_with_bfs ()

Public Attributes

- unsigned int tests_run
- unsigned int tests_failed

Static Public Attributes

- static unsigned int total_tests

Friends

- class Sommet
- class ArbreB

4.3.1 Detailed Description

The class `Part1` implements tests to assert that the functions in classes `Sommet` and `ArbreB` have the expected behavior.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020/11/17

4.3.2 Constructor & Destructor Documentation

4.3.2.1 Part1()

```
Part1::Part1 ( )
```

Creates an object `Part1`.

4.3.3 Member Function Documentation

4.3.3.1 should_assign_ArbreB()

```
bool Part1::should_assign_ArbreB ( )
```

Asserts that the overload of `operator=` for `ArbreB` assign the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.2 should_assign_Sommet()

```
bool Part1::should_assign_Sommet ( )
```

Asserts that the overload of `operator=` for `Sommet` assign the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.3 should_create_ArbreB_from_Sommet()

```
bool Part1::should_create_ArbreB_from_Sommet ( )
```

Asserts that the constructor of `ArbreB` from a `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.4 should_create_copy_ArbreB()

```
bool Part1::should_create_copy_ArbreB ( )
```

Asserts that the copy constructor of `ArbreB` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.5 should_create_copy_Sommet()

```
bool Part1::should_create_copy_Sommet ( )
```

Asserts that the copy constructor of `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.6 should_create_default_ArbreB()

```
bool Part1::should_create_default_ArbreB ( )
```

Asserts that the default constructor of `ArbreB` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.7 should_create_default_Sommet()

```
bool Part1::should_create_default_Sommet ( )
```

Asserts that the default constructor of `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.8 should_create_parameterized_ArbreB()

```
bool Part1::should_create_parameterized_ArbreB ( )
```

Asserts that the parameterized constructor of `ArbreB` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.9 should_create_parameterized_Sommet()

```
bool Part1::should_create_parameterized_Sommet ( )
```

Asserts that the parameterized constructor of `Sommet` initializes the object correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.10 should_decompose_one_ArbreB_into_two()

```
bool Part1::should_decompose_one_ArbreB_into_two ( )
```

Asserts that decomposing an `ArbreB` returns a tuple holding two `ArbreB`s with the expected values.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.11 should_find_character_c()

```
bool Part1::should_find_character_c ( )
```

Asserts that the character `c` is found in an `ArbreB` that contains it.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.12 should_find_character_y_with_bfs()

```
bool Part1::should_find_character_y_with_bfs ( )
```

Asserts that using the method `search()` (BFS algorithm), the character `c` is found in an `ArbreB` that contains it.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.13 should_fuse_two_ArbreB()

```
bool Part1::should_fuse_two_ArbreB ( )
```

Asserts that the overload of `operator+` for `ArbreB` fuses two `ArbreB`s into one and has the expected values at its root.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.14 should_insert_Sommet_into_ArbreB()

```
bool Part1::should_insert_Sommet_into_ArbreB ( )
```

Asserts that a `Sommet` is correctly inserted into an `ArbreB`. This method tests for both `insert(const Sommet&)` and `insert(const char&, const double&)`.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.15 should_not_find_character_s_with_bfs()

```
bool Part1::should_not_find_character_s_with_bfs ( )
```

Asserts that using the method `search()` (BFS algorithm), the character `s` is not found in an `ArbreB` that does not contain it.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.16 should_not_find_character_z()

```
bool Part1::should_not_find_character_z ( )
```

Asserts that the character `z` is not found in an `ArbreB` that does not contain it.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.17 should_not_link_ArbreB_copies()

```
bool Part1::should_not_link_ArbreB_copies ( )
```

Asserts that the copy of an object `ArbreB` is not linked with the original.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.18 should_not_link_Sommet_copies()

```
bool Part1::should_not_link_Sommet_copies ( )
```

Asserts that the copy of an object `Sommet` is not linked with the original.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.19 should_remove_leaf()

```
bool Part1::should_remove_leaf ( )
```

Asserts that removing a `Sommet` that is a leaf deletes it correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.20 should_remove_Sommet_with_one_child()

```
bool Part1::should_remove_Sommet_with_one_child ( )
```

Asserts that removing a `Sommet` that has only one child (left or right) deletes it correctly and replaces it with its child.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.21 should_remove_Sommet_with_two_children()

```
bool Part1::should_remove_Sommet_with_two_children ( )
```

Asserts that removing a `Sommet` that has two children deletes it correctly and replaces it with its inorder succesor.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.22 should_set_Sommet_values()

```
bool Part1::should_set_Sommet_values ( )
```

Asserts that the setters for `Sommet` set the values correctly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.3.23 should_update_freq_if_char_already_in_ArbreB()

```
bool Part1::should_update_freq_if_char_already_in_ArbreB ( )
```

Asserts that inserting a `Sommet` that already is in the `ArbreB` updates the `m_Freq` field of that `Sommet` accordingly.

Returns

`True` if the test passed, `false` if it failed.‘

4.3.4 Friends And Related Function Documentation

4.3.4.1 ArbreB

```
friend class ArbreB  [friend]
```

4.3.4.2 Sommet

```
friend class Sommet  [friend]
```

4.3.5 Member Data Documentation

4.3.5.1 tests_failed

```
unsigned int Part1::tests_failed
```

Represents the number of tests failed.

4.3.5.2 tests_run

```
unsigned int Part1::tests_run
```

Represents the number of tests ran.

4.3.5.3 total_tests

```
unsigned int Part1::total_tests  [static]
```

Represents the total number of tests.

The documentation for this class was generated from the following file:

- src/headers/Part1.hpp

4.4 AppWindow Class Reference

The class `AppWindow` represents the GUI that dislays the program.

```
#include <AppWindow.hpp>
```

Inheritance diagram for AppWindow:

Collaboration diagram for AppWindow:

Public Slots

- void run_compression ()
- void run_uncompression ()
- void clear_text ()

Public Member Functions

- AppWindow ()
- ~AppWindow ()

4.4.1 Detailed Description

The class `AppWindow` represents the GUI that dislays the program.

Author

Gabriel Dos Santos

4.4.2 Constructor & Destructor Documentation

4.4.2.1 AppWindow()

```
AppWindow::AppWindow ( )
```

Creates a new App Window object.

4.4.2.2 ~AppWindow()

```
AppWindow::~AppWindow ( )
```

Destroys the App Window object

4.4.3 Member Function Documentation

4.4.3.1 clear_text

```
void AppWindow::clear_text ( )  [slot]
```

Clears the text in the text boxes.

4.4.3.2 run_compression

```
void AppWindow::run_compression ( )  [slot]
```

Compresses the text in the input text box.

4.4.3.3 run_uncompression

```
void AppWindow::run_uncompression ( )  [slot]
```

Uncompresses the text in the input box. It will raise an error if no text has been compressed before or if the text could not be fully uncompressed.

The documentation for this class was generated from the following file:

- src/headers/AppWindow.hpp

Chapter 5

File Documentation

5.1 src/headers/Sommet.hpp File Reference

Implementation of the class Sommet.

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

Classes

- class Sommet

  The class `Sommet` represents a node of the class `ArbreB`.

5.1.1 Detailed Description

Implementation of the class Sommet.

Author

Gabriel Dos Santos, Raphael Marouani

Version

0.2.0

Date

2020-12-14

5.2 src/headers/ArbreB.hpp File Reference

Implementation of the class ArbreB.

```
#include <string>
#include <tuple>
#include <map>
#include "Sommet.hpp"
```

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

Classes

- class ArbreB

  The class `ArbreB` represents a binary tree.

5.2.1 Detailed Description

Implementation of the class ArbreB.

Author

Gabriel Dos Santos, Raphael Marouani

Version

0.2.0

Date

2020-12-14

5.3 src/headers/Part1.hpp File Reference

Implementation of the class Part1.

```
#include "../headers/ArbreB.hpp"
```

Include dependency graph for Part1.hpp:

Classes

- class Part1

  The class `Part1` implements tests to assert that the functions in classes `Sommet` and `ArbreB` have the expected behavior.

5.3.1 Detailed Description

Implementation of the class Part1.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020-12-14

5.4 src/headers/Part2.hpp File Reference

Collection of functions used for the second part of C++ Project for module IN505.

```
#include <fstream>
#include <vector>
#include "ArbreB.hpp"
```

Include dependency graph for Part2.hpp:

Functions

- unsigned int find (std::vector< ArbreB > &vec, char &content)
- ArbreB find_lowest (std::vector< ArbreB > &btrees)
- std::string parse_file_to_string (char *filename)
- std::vector< ArbreB > build_btree_vector (std::string &content)
- ArbreB build_huffman_tree (std::vector< ArbreB > &btrees)
- std::string compress_to_bin (std::map< char, std::string > map, std::string file_content)
- void print_input (std::string input)
- void print_map (std::map< char, std::string > map)
- void print_output (std::string output)

5.4.1 Detailed Description

Collection of functions used for the second part of C++ Project for module IN505.

Author

Gabriel Dos Santos

Version

0.1.0

5.4.2 Function Documentation

5.4.2.1 build_btree_vector()

```
std::vector<ArbreB> build_btree_vector (
              std::string & content )
```

Constructs a vector of ArbreBs from the specified string. For each character present in the string, creates an ArbreB in the vector. The ArbreB is initialized with the read character if it is unknown and a frequency of 1. If the read character is already know, increments its frequency by one. When the whole string is read, transforms the frequency of each character to a percentage.

Parameters

content	The string to build the vector from.

Returns

A vector of ArbreB for each character of the string and their frequency.

5.4.2.2 build_huffman_tree()

```
ArbreB build_huffman_tree (
              std::vector< ArbreB > & btrees )
```

Builds the Huffman tree for a given vector of ArbreBs. Fuses the ArbreBs of the vector together following the Huffman algorithm.

Parameters

btrees	The vector of ArbreBs to fuse.

Returns

An Huffman tree for the given vector of ArbreB.

5.4.2.3 compress_to_bin()

```
std::string compress_to_bin (
              std::map< char, std::string > map,
              std::string file_content )
```

Compresses a given string to its binary representation. Reads the string and for each character (key), appends the corresponding binary value to be returned.

Parameters

map	A map of characters (keys) and their binary representations (values).
file_content	The string to compress.

Returns

The compressed string.

5.4.2.4 find()

```
unsigned int find (
            std::vector< ArbreB > & vec,
            char & content )
```

Checks if an ArbreB holding the specified character is present in the vector. Uses a boolean expression to break the for loop early if the characcter is found.

Parameters

vec	A reference to the vector to search in.
content	The character to search for.

Returns

The index of the ArbreB holding the character + 1 if found, 0 otherwise.

5.4.2.5 find_lowest()

```
ArbreB find_lowest (
            std::vector< ArbreB > & btrees )
```

Finds the ArbreB with the lowest frequency in the specified vector. Creates a copy of the ArbreB with the lowest frequency. Deletes the original from the vector.

Parameters

btrees	A reference of the vector to search in.

Returns

A copy of the ArbreB with the lowest frequency.

5.4.2.6 parse_file_to_string()

```
std::string parse_file_to_string (
              char * filename )
```

Parses a given text file into a string. Reads each character of the file and appends it to a string.

Parameters

filename	The name of the file to parse.

Returns

A string holding the content of the file.

5.4.2.7 print_input()

```
void print_input (
              std::string input )
```

Utility function that simply prints the input text to the terminal.

Parameters

input	A string holding the content of a text file.

5.4.2.8 print_map()

```
void print_map (
              std::map< char, std::string > map )
```

Utility function that prints the binary representation of each character in a string.

Parameters

map	A map holding characters as keys and their binary representations as values.

5.4.2.9 print_output()

```
void print_output (
              std::string output )
```

Utility function that simply prints the compressed input text as binary to the terminal.

Parameters

output	A string holding the binary representation of the content of a text file.

5.5 src/headers/AppWindow.hpp File Reference

Implementation of the class AppWindow for GUI with Qt5.

```
#include <QtCore/QObject>
#include <QtWidgets/QWidget>
#include <QtWidgets/QApplication>
#include <QtWidgets/QGridLayout>
#include <QtWidgets/QPushButton>
#include <QtWidgets/QTextEdit>
#include "Part2.hpp"
#include "Part3.hpp"
```

Include dependency graph for AppWindow.hpp:

5.6 src/headers/Part3.hpp File Reference

Collection of functions for the third part of the project.

```
#include "Part2.hpp"
```

Include dependency graph for Part3.hpp: This graph shows which files directly or indirectly include this file:

Functions

- bool is_huffman_code (std::string &input)
- std::string uncompress_binary (std::string &input, ArbreB &huffman)

5.6.1 Detailed Description

Collection of functions for the third part of the project.

Author

Gabriel Dos Santos

Version

0.1.0

Date

2020-12-21

5.6.2 Function Documentation

5.6.2.1 is_huffman_code()

```
bool is_huffman_code (
            std::string & input )
```

Verifies that the specified input is in binary format.

Parameters

input	The input string.

Returns

True if the input is binary, false otherwise.

5.6.2.2 uncompress_binary()

```
std::string uncompress_binary (
              std::string & input,
              ArbreB & huffman )
```

Uncompresses a binary string to its ASCII format. Performs a inorder traversal of the specified Huffman tree to uncompress the input binary.

Parameters

input	The input string in binary format.
huffman	The huffman tree to traverse to uncompress the input string.

Returns

std::string The uncompressed string in ASCII format.

Index