ASCII Obfuscation 0.0.1

Generated by Doxygen 1.8.13

Contents

1	1000	DO LIST			1
2	File	File Index			
	2.1	File Lis	st		3
3	File	Docum	entation		5
	3.1	depred	ated_func	tions.c File Reference	5
		3.1.1	Function	Documentation	5
			3.1.1.1	character_bit_shifting()	5
			3.1.1.2	shift_letters()	6
		3.1.2	Variable	Documentation	6
			3.1.2.1	bit_shifted_char_table	6
			3.1.2.2	input_table	6
	3.2	depred	ated_func	tions.h File Reference	6
		3.2.1	Macro D	efinition Documentation	7
			3.2.1.1	SHIFT_VAL	7
		3.2.2	Function	Documentation	7
			3.2.2.1	character_bit_shifting()	7
			3.2.2.2	shift_letters()	7
	3.3	functio	ns.c File F	Reference	8
		3.3.1	Function	Documentation	8
			3.3.1.1	randomize_value()	8
			3.3.1.2	translate_into_obscure()	9
	3.4	functio	ns.h File F	Reference	9

ii CONTENTS

	3.4.1	Function Documentation	9
		3.4.1.1 randomize_value()	9
		3.4.1.2 translate_into_obscure()	10
3.5	invert_	ascii.c File Reference	11
	3.5.1	Function Documentation	11
		3.5.1.1 main()	11
3.6	main.c	File Reference	12
	3.6.1	Function Documentation	12
		3.6.1.1 main()	12
	3.6.2	Variable Documentation	13
		3.6.2.1 title_str	13
3.7	main.h	File Reference	13
	3.7.1	Macro Definition Documentation	14
		3.7.1.1 ever	14
		3.7.1.2 forever	14
		3.7.1.3 UPPERCASE_LOWERCASE_SHIFT	14
	3.7.2	Enumeration Type Documentation	14
		3.7.2.1 error_codes	14
	3.7.3	Variable Documentation	15
		3.7.3.1 consonants	15
		3.7.3.2 message_str	15
		3.7.3.3 vowels	15
3.8	unit_te	sts.c File Reference	16
	3.8.1	Function Documentation	16
		3.8.1.1 main()	16
		3.8.1.2 setUp()	16
		3.8.1.3 tearDown()	17
		3.8.1.4 TEST_randomize_value()	17
3.9	utils.c	File Reference	17
	3.9.1	Variable Documentation	17
		3.9.1.1 consonants	17
		3.9.1.2 message_str	18
		3.9.1.3 vowels	18
Index			19

Chapter 1

Todo List

Global main (int argC, char **argV)

Add a check on return code for 'translate_into_obscure' function (and take action depending on it)

Global randomize_value (char input, char *output, unsigned int *offset)

We have to revise the following lines to ensure they do what they are supposed to

2 Todo List

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

deprecated_functions.c	5
deprecated_functions.h	6
functions.c	8
functions.h	9
invert_ascii.c	11
main.c	
main.h	
unit_tests.c	
utils.c	17

File Index

Chapter 3

File Documentation

3.1 deprecated_functions.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>
#include "deprecated_functions.h"
```

Functions

void shift_letters (void)

Shift letters from upper case to lower case or from lower case to upper case.

void character_bit_shifting (void)

Bit shifts a string of character to make it unreadable.

Variables

- const char input_table [] = "I think you know my point about inline if operations: it only obfuscates the code.\n"
- const char bit_shifted_char_table []

3.1.1 Function Documentation

3.1.1.1 character_bit_shifting()

```
\begin{array}{c} \mbox{void character\_bit\_shifting (} \\ \mbox{void )} \end{array}
```

Bit shifts a string of character to make it unreadable.

Definition at line 89 of file deprecated_functions.c.

References input_table.

3.1.1.2 shift_letters()

Shift letters from upper case to lower case or from lower case to upper case.

Definition at line 40 of file deprecated functions.c.

References SHIFT_VAL.

3.1.2 Variable Documentation

3.1.2.1 bit_shifted_char_table

```
const char bit_shifted_char_table[]
```

Initial value:

Definition at line 18 of file deprecated_functions.c.

3.1.2.2 input_table

```
const char input_table[] = "I think you know my point about inline if operations : it only obfuscates the code.\n"
```

Definition at line 17 of file deprecated_functions.c.

Referenced by character_bit_shifting().

3.2 deprecated_functions.h File Reference

Macros

• #define SHIFT_VAL ('a'-'A')

Functions

void shift_letters (void)

Shift letters from upper case to lower case or from lower case to upper case.

void character_bit_shifting (void)

Bit shifts a string of character to make it unreadable.

3.2.1 Macro Definition Documentation

```
3.2.1.1 SHIFT_VAL
```

```
#define SHIFT_VAL ('a'-'A')
```

Definition at line 6 of file deprecated_functions.h.

Referenced by shift_letters().

3.2.2 Function Documentation

3.2.2.1 character_bit_shifting()

Bit shifts a string of character to make it unreadable.

Definition at line 89 of file deprecated_functions.c.

References input_table.

3.2.2.2 shift_letters()

```
void shift_letters (
     void )
```

Shift letters from upper case to lower case or from lower case to upper case.

Definition at line 40 of file deprecated_functions.c.

References SHIFT_VAL.

3.3 functions.c File Reference

```
#include "functions.h"
#include "main.h"
#include <stdlib.h>
#include <iso646.h>
```

Functions

 $\bullet \ \ \text{int } \underline{\text{randomize_value}} \ (\text{char input, char } * \text{output, unsigned int } * \text{offset}) \\$

function implementation for ascii_obfuscation

int translate_into_obscure (char *input, unsigned int input_length, unsigned char *output, unsigned int *p
 —output_length)

Translation loop function.

3.3.1 Function Documentation

3.3.1.1 randomize_value()

function implementation for ascii_obfuscation

function definition for ascii_obfuscation

functions.c

Note

Created by vince on 08/02/2020.

Todo We have to revise the following lines to ensure they do what they are supposed to

Definition at line 43 of file functions.c.

References RETURN_OK.

Referenced by TEST_randomize_value(), and translate_into_obscure().

3.3.1.2 translate_into_obscure()

Translation loop function.

Translation loop function Parameters:

- [in] input String that contains the original message
- [in] input_length Length of input string
- · [out] output Buffer in which we would write the resulting string
- [out] p_output_length Pointer to length of output string Return : An int value :
 - RETURN OK (0) if everything is OK
 - GENERIC_ERROR (-1) if process ends in error

If 'output' buffer is 'NULL' we might get in trouble trying to set values to a random memory location => We would better exit the function as soon as possible with an error code to inform the caller

If there is no input buffer passed in argument, use default string 'message_str'

Definition at line 86 of file functions.c.

References GENERIC_ERROR, message_str, randomize_value(), and RETURN_OK.

Referenced by main(), and TEST_randomize_value().

3.4 functions.h File Reference

Functions

• int randomize_value (char input, char *output, unsigned int *offset)

 $\textit{function definition for } \texttt{ascii_obfuscation}$

int translate_into_obscure (char *input, unsigned int input_length, unsigned char *output, unsigned int *p
 —output_length)

Translation loop function.

3.4.1 Function Documentation

3.4.1.1 randomize_value()

function definition for ascii_obfuscation

functions.h

Note

Created by vince on 08/02/2020. Make 2 randomized letters out of 1 letter input

Parameters

	in	input	Input character to be randomized
	out	output	Buffer in which we would write the resulting 2 characters
Ī	out	offset	Offset of output characters in output buffer

Returns

An int value:

- RETURN_OK (0) if everything is OK
- GENERIC_ERROR (-1) if process ends in error

 $function \ definition \ for \ {\tt ascii_obfuscation}$

functions.c

Note

Created by vince on 08/02/2020.

Todo We have to revise the following lines to ensure they do what they are supposed to

Definition at line 43 of file functions.c.

References RETURN_OK.

Referenced by TEST_randomize_value(), and translate_into_obscure().

3.4.1.2 translate_into_obscure()

Translation loop function.

Parameters

	in	input	String that contains the original message
	in	input_length	Length of input string
	out	output	Buffer in which we would write the resulting string
ĺ	out	p_output_length	Pointer to length of output string

Returns

An int value:

- RETURN_OK (0) if everything is OK
- · GENERIC ERROR (-1) if process ends in error

Translation loop function Parameters:

- [in] input String that contains the original message
- [in] input_length Length of input string
- · [out] output Buffer in which we would write the resulting string
- [out] p_output_length Pointer to length of output string Return : An int value :
 - RETURN_OK (0) if everything is OK
 - GENERIC_ERROR (-1) if process ends in error

If 'output' buffer is 'NULL' we might get in trouble trying to set values to a random memory location => We would better exit the function as soon as possible with an error code to inform the caller

If there is no input buffer passed in argument, use default string 'message str'

Definition at line 86 of file functions.c.

References GENERIC_ERROR, message_str, randomize_value(), and RETURN_OK.

Referenced by main(), and TEST_randomize_value().

3.5 invert_ascii.c File Reference

```
#include <stdio.h>
```

Functions

• int main (void)

3.5.1 Function Documentation

3.5.1.1 main()

```
int main ( void )
```

Definition at line 3 of file invert_ascii.c.

3.6 main.c File Reference

```
#include <stdio.h>
#include <string.h>
#include "functions.h"
#include "main.h"
```

Functions

• int main (int argC, char **argV)

Main program function.

Variables

• const char title_str[]

Title string to ebe displayed in console.

3.6.1 Function Documentation

3.6.1.1 main()

```
int main (
    int argC,
    char ** argV)
```

Main program function.

Parameters

in	argC	Argument number	
in	argV	Pointer onto input argument strings	

Returns

RETURN_OK (0) in case of successful execution

Conditional behavior of program:

- if there are arguments in the program call, just translates the input string and exit (or print error message)
- if there is no argument, go to the infinite loop to use the program until asked to quit

3.7 main.h File Reference

Remarks

show how many arguments are passed to the program (following line is only here for debug)

- Just print the input string for now (translation part of the program is not finished yet)
 - · Perhaps keep a print of which option has been chosen in the end ('encode' or 'decode' for example)

Todo Add a check on return code for 'translate_into_obscure' function (and take action depending on it)

Definition at line 50 of file main.c.

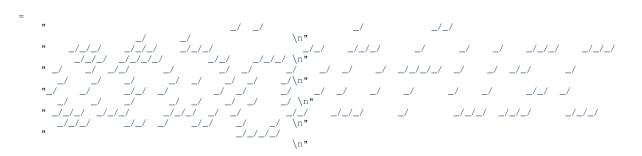
References forever, RETURN_OK, title_str, and translate_into_obscure().

3.6.2 Variable Documentation

```
3.6.2.1 title_str
```

```
const char title_str[]
```

Initial value:



Title string to ebe displayed in console.

ascii obfuscation

Definition at line 20 of file main.c.

Referenced by main().

3.7 main.h File Reference

Macros

- #define UPPERCASE_LOWERCASE_SHIFT ('a'-'A')

 ascii_obfuscation program header file for specific definition share
- #define ever (;;)
- #define forever for ever

Enumerations

enum error_codes { GENERIC_ERROR = -1, RETURN_OK = 0 }
 Define custom/personal error codes to avoid using magic numbers.

Variables

• const char vowels [6]

Global utility elements for ascii_obfuscation

• const char consonants [20]

Consonants table.

• const char message_str [0xFF]

Default test message string for character manipulation.

3.7.1 Macro Definition Documentation

```
3.7.1.1 ever
```

```
#define ever (;;)
```

Definition at line 24 of file main.h.

3.7.1.2 forever

```
#define forever for ever
```

Definition at line 25 of file main.h.

Referenced by main().

3.7.1.3 UPPERCASE_LOWERCASE_SHIFT

```
#define UPPERCASE_LOWERCASE_SHIFT ('a'-'A')
ascii_obfuscation program header file for specific definition share
main.h
```

Note

Created by vince on 09/02/2020.difference between uppercase and lower case letters

Definition at line 21 of file main.h.

3.7.2 Enumeration Type Documentation

3.7.2.1 error_codes

```
enum error_codes
```

Define custom/personal error codes to avoid using magic numbers.

3.7 main.h File Reference

Enumerator

GENERIC_ERROR	General purpose error ; no particular details to share.
RETURN_OK	All good!

Definition at line 32 of file main.h.

3.7.3 Variable Documentation

3.7.3.1 consonants

const char consonants[20]

Consonants table.

Definition at line 20 of file utils.c.

3.7.3.2 message_str

```
const char message_str[0xFF]
```

Default test message string for character manipulation.

Definition at line 24 of file utils.c.

Referenced by translate_into_obscure().

3.7.3.3 vowels

const char vowels[6]

 ${\bf Global\ utility\ elements\ for\ ascii_obfuscation}$

utils.c

Note

Created by vince on 09/02/2020.vowels table

Definition at line 18 of file utils.c.

3.8 unit_tests.c File Reference

```
#include "unity.h"
#include "functions.h"
#include <string.h>
#include <iso646.h>
```

Functions

void setUp (void)

 $\textit{Unit test implementation for } ascii_obfuscation$

- void tearDown (void)
- void TEST_randomize_value (void)
- int main (void)

3.8.1 Function Documentation

```
3.8.1.1 main()
```

```
int main (
     void )
```

Definition at line 63 of file unit_tests.c.

References TEST_randomize_value().

```
3.8.1.2 setUp()
```

```
void setUp (
     void )
```

Unit test implementation for ascii_obfuscation

unit_tests.c

Note

Created by vince on 08/02/2020.

Definition at line 21 of file unit_tests.c.

3.9 utils.c File Reference

3.8.1.3 tearDown()

```
void tearDown (
     void )
```

Definition at line 26 of file unit_tests.c.

3.8.1.4 TEST_randomize_value()

Definition at line 35 of file unit_tests.c.

References randomize_value(), and translate_into_obscure().

Referenced by main().

3.9 utils.c File Reference

```
#include "main.h"
```

Variables

const char vowels [6] = { 'a', 'e', 'i', 'o', 'u', 'y' }

Global utility elements for ascii_obfuscation

• const char consonants [20]

Consonants table.

• const char message_str [0xFF]

Default test message string for character manipulation.

3.9.1 Variable Documentation

3.9.1.1 consonants

```
const char consonants[20]
```

Initial value:

```
= { 'b', 'c', 'd', 'f', 'g', 'h', 'j', 'k', 'l', 'm', 'n', 'p', 'q', 'r', 's', 't', 'v', 'w', 'x', 'z' }
```

Consonants table.

Definition at line 20 of file utils.c.

3.9.1.2 message_str

```
const char message_str[0xFF]
```

Initial value:

```
= \ ^{"}I think you know my point about inline if operations : it only obfuscates the code.\0"
```

Default test message string for character manipulation.

Definition at line 24 of file utils.c.

Referenced by translate_into_obscure().

3.9.1.3 vowels

```
const char vowels[6] = { 'a', 'e', 'i', 'o', 'u', 'y' }
```

Global utility elements for ascii_obfuscation

utils.c

Note

Created by vince on 09/02/2020.vowels table

Definition at line 18 of file utils.c.

Index

bit_shifted_char_table	message_str, 15
deprecated_functions.c, 6	UPPERCASE_LOWERCASE_SHIFT, 14
	vowels, 15
character_bit_shifting	message_str
deprecated_functions.c, 5	main.h, 15
deprecated_functions.h, 7	utils.c, 17
consonants	
main.h, 15	randomize_value
utils.c, 17	functions.c, 8
	functions.h, 9
deprecated_functions.c, 5	
bit_shifted_char_table, 6	SHIFT_VAL
character_bit_shifting, 5	deprecated_functions.h, 7
input_table, 6	setUp
shift_letters, 5	unit_tests.c, 16
deprecated_functions.h, 6	shift_letters
character_bit_shifting, 7	deprecated_functions.c, 5
SHIFT_VAL, 7	deprecated_functions.h, 7
shift_letters, 7	
	TEST_randomize_value
error_codes	unit_tests.c, 17
main.h, 14	tearDown
ever	unit_tests.c, 16
main.h, 14	title_str
	main.c, 13
forever	translate_into_obscure
main.h, 14	functions.c, 8
functions.c, 8	functions.h, 10
randomize_value, 8	
translate_into_obscure, 8	UPPERCASE_LOWERCASE_SHIFT
functions.h, 9	main.h, 14
randomize_value, 9	unit_tests.c, 16
translate_into_obscure, 10	main, 16
	setUp, 16
input_table	TEST_randomize_value, 17
deprecated_functions.c, 6	tearDown, 16
invert_ascii.c, 11	utils.c, 17
main, 11	consonants, 17
	message_str, 17
main	vowels, 18
invert_ascii.c, 11	
main.c, 12	vowels
unit_tests.c, 16	main.h, 15
main.c, 12	utils.c, 18
main, 12	
title_str, 13	
main.h, 13	
consonants, 15	
error_codes, 14	
ever, 14	
forever, 14	