Hello GitHub Classroom Assignment Instructions

bucs220@github

February 17, 2021

1 Instructions

- Answer the questions individually. Group effort is not allowed. If you utilize code you found online or in print, you must cite its origin within your source code in accordance with the class syllabus' academic honesty and integrity policies.
- Unless otherwise specified, you may include any header from libc and libmath. You may not link to other library code.
- Solutions must be committed to your assignment repository on Github.
- Your code will be evaluated using a CUnit test suite (see **test.c**).
- Useful resources:
  - Common Linux Commands
  - `http://c-faq.com/`
  - `https://cdecl.org`

2 Questions

2.1 Summation of Two Integers

2.1.1 Description

Write the function `long sum(int a, int b)` such that given integers a, b, the function returns the sum of these integers.

2.1.2 Examples

See Table 1

Input		Output
a	b	
0	0	0
1	2	3
5	-3	2
-5	3	-2

Table 1: Inputs and expected outputs for `long sum(int a, int b)`.

2.2 Fibonacci Sequence

2.2.1 Description

The Fibonacci sequence is the sequence of numbers defined by the following linear recurrence relation:

$$\{F_i\}_{i=0}^{\infty} = \begin{cases} 0 & F_0 \\ 1 & F_1 \\ F_{i-1} + F_{i-2} & F_i \end{cases} \tag{1}$$

See OEIS sequence A000045. Write the function `bool isFib(int n)` such that given an integer n, the function returns true if n is a term in the Fibonacci sequence, and false otherwise.

2.2.2 Examples

See Table 2

Input	Output
n	
0	true
1	true
2	true
3	true
4	false

Table 2: Inputs and expected outputs for `bool isFib(int n)`.

2.3 FizzBuzz

2.3.1 Description

FizzBuzz is a game where players take turns reciting the next natural number with the following rules:

- If the number is divisible by 3, the player says *FIZZ*

- If the number is divisible by 5, the player says *BUZZ*
- If the number is divisible by both 3 and 5, the player says *FIZZBUZZ*
- Otherwise, the player says the number.

Write a function `char* fizzbuzz(int i, char* dest, size_t n)`, such that given an integer i, and a buffer $dest$ of size n, the player's response is written into the buffer and null-terminated. The response is guaranteed to fit into the buffer, including the null terminator. The integer is not guaranteed to be positive, and thus any negative sign must also be written into the buffer.

2.3.2 Examples

Assume there is a buffer located on the stack at address `0x7fffdd35c940`. See Table 3.

Input			Output
i	dest	n	ASCII Contents of dest (without quotes)
3	0x7fffdd35c940	9	”FIZZ␀␀␀␀␀”
5	0x7fffdd35c940	9	”BUZZ␀␀␀␀␀”
15	0x7fffdd35c940	9	”FIZZBUZZ␀”
16	0x7fffdd35c940	9	”16␀␀␀␀␀␀␀”
-45	0x7fffdd35c940	12	”-45␀␀␀␀␀␀␀␀␀”
0	0x7fffdd35c940	9	”FIZZBUZZ␀”

Table 3: Inputs and expected outputs for `char* fizzbuzz(int i, char* dest, size_t n)`.