CS 137: Assignment #3

Due on Friday, Oct 7, 2022, at 11:59 PM

Submit all programs using the Marmoset Submission and Testing Server located at
https://marmoset.student.cs.uwaterloo.ca/

Victoria Sakhnini

Fall 2022

Notes:

- Use the examples to guide your formatting for your output. Remember to terminate your output with a newline character.
- Integers should be read using `scanf`.
- **You must solve all problems using recursion. You must NOT use loops.**
- **Do not write for/while in your submitted solution, even as a comment.**
- For this assignment and all future assignments, I strongly recommend that you solve the extra practice problems in the course notes before starting working on the assignment.
- You must NOT use MATH Library

Problem 1

Create a C program `functions.c` that contains the following:

a)

```
bool divide(int a, int b);
```

The function returns `true` if a divides b and `false` otherwise.

Assumptions: both a and b are non-zero integers.

Restrictions: You must NOT use any of `%` `/` `*` in your solution.

Definition: a divides b if there is an integer c such that a*c = b

b)

```
int IntegerDivision(int a, int b);
```

The function returns the integer division of a by b.

Assumptions: both a and b are positive.

Restrictions: You must NOT use any of `%` `/` `*` in your solution.

Note: You are to submit this file containing only your implemented function and any additional functions you defined (that is, you must delete the test cases portion and the `main` function). However, **you must keep the required included libraries.**

The following code will help you with testing

```c
#include <stdio.h>
#include <assert.h>
#include <stdbool.h>

// The rest of the Code is here

int main(void){
        assert(divide(2,10));
        assert(divide(2,-10));
        assert(!divide(7,22));
        assert(IntegerDivision(10,5)==2);
        assert(IntegerDivision(151,5)==30);
 }
```

Problem 2

Assume you want to go upstairs of n (>=1) steps. There are three different methods:

1. You walk 1 step up or jump 2 steps up at any time.

2. You walk 1 step up or jump 2 steps, but must keep the same step size 3 times in a row (excluding the ending steps)?

3. You walk 2-step jump or 3-step jump (no 1-step walk-up), and all the jumps have to be precisely 3 times in a row, including the ending jumps?

Example for n=6

1. There are 13 different ways to go up 6 stairs using method 1
2. There are 2 different ways to go up 6 stairs using method 2 (1,1,1,1,1,1 or 2,2,2). Notice that 1,1,1,1,2 is not a valid option, why?
3. There is 1 way to go up 6 stairs using method 3 (2,2,2)

Create a program `jumpstairs.c` that includes the following three functions to implement the three methods above.

```
int jump_stair_v1(int n);

int jump_stair_v2(int n);

int jump_stair_v3(int n);
```

You are to submit this file containing only your implemented function and any additional functions you defined (that is, you must delete the test cases portion and the `main` function). However, **you must keep the required included libraries.**

The following Code will help you with testing

```
1.  int main(void)
2.  {
3.      assert(1==jump_stair_v1(1));
4.      assert(1==jump_stair_v2(1));
5.      assert(0==jump_stair_v3(1));
6.      assert(2==jump_stair_v1(2));
7.      assert(2==jump_stair_v2(2));
8.      assert(0==jump_stair_v3(2));
9.      assert(13==jump_stair_v1(6));
10.     assert(2==jump_stair_v2(6));
11.     assert(1==jump_stair_v3(6));
12.     assert(21==jump_stair_v1(7));
13.     assert(3==jump_stair_v2(7));
14.     assert(0==jump_stair_v3(7));
15.     assert(55==jump_stair_v1(9));
16.     assert(3==jump_stair_v2(9));
17.     assert(1==jump_stair_v3(9));
18.
19.     return 0;
20. }
```

Problem 3

Create a C program `nstars.c` that contains a function

```
void stars(long int n);
```

The function prints * between each two consecutive digits that are equal.

For example: `starts(11102222)` prints `1*1*102*2*2*2`. No `'\n'` is printed by the function at the end of the output

You are to submit this file containing only your implemented function and any additional functions you defined (that is, you must delete the test cases portion and the `main` function). However, **you must keep the required included libraries.**

The following Code will help you with testing

```c
int main(void) {

    stars(11102222);
    printf("\n");
    stars(1234567);
    printf("\n");

    return 0;
}
```

Expected output:

```
1*1*102*2*2*2
1234567
```