CS 137: Assignment #3

Due on Friday, Oct 4, 2024, 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 2024

Notes:

- Use the examples to guide your formatting for your output. Remember to terminate your output with a newline character unless the question asks something different.
- **You must solve all problems using recursion. You must NOT use loops.**
- For this and all future assignments, I strongly recommend that you solve the extra practice problems in the course notes before working on the assignment.
- You must NOT use MATH Library

Problem 1

Create a C file `pairs.c` that contains the function `void numberPairs(int n);` which takes a positive integer `n` and prints the integer where every digit is paired up next to itself. There should not be a sequence printed of more than two of the same digits in a row. If a digit appears by itself with no duplicate on either side, duplicate that digit when printing. The function does **NOT** print `\n` at the end of the line.

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.**

Examples:

```
numberPairs(1234 )          prints  11223344
numberPairs(338888886)      prints  338866
numberPairs(555555555)      prints  55
numberPairs(1)              prints  11
```

Problem 2

A number is called a narcissistic number when the sum of each digit raised to the power of the number of digits is equal to the number itself. For example,
153 is a narcissistic number because $1^3+5^3+3^3=1+125+27=153$

Create a C program `narcissistic.c` that includes the function `bool narcissist(int n);` which takes an integer `n>0` and returns `true` if `n` is a narcissistic number; otherwise, it returns `false`.

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>

bool narcissist(int n){
    …..
}

int main(void) {
    assert(narcissist(1));
    assert(narcissist(9));
    assert(narcissist(153));
    assert(narcissist(370));
    assert(narcissist(92727));
    assert(narcissist(548834));
    assert(!narcissist(10));
    assert(!narcissist(92));
    assert(!narcissist(1535));
    assert(!narcissist(1234));
    assert(!narcissist(92726));
    assert(!narcissist(93083));

    return 0;
}

```

Problem 3

Create a C program `treeprint.c` that includes the function `void tree(int n);` which takes an integer `n>0` and prints a tree ASCII art picture.

- The height of the tree is `2n+1`.
- The tree is made of an isosceles triangle with a height of `n+1` and a base of `2n+1`.
- The trunk of the tree is a rectangle with a length of `n` and a width of half of `n` if `n` is an even number or half of `n+1` if `n` is an odd number.
- The tree should be vertically symmetric. If the width of the trunk is an even number, we would add 1 to the width. Please note that when the width is 1, you only need to print one single vertical line as trunk.
- The top of the triangle should be printed by `*` characters.
- The sides of triangle should be printed by `+` characters.
- The trunk of the tree should be printed by `|` characters.
- Each line starts with a **.** character and ends with a **.** character except the base of the triangle, which starts with `*` character and ends with `*` character.

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.**

Examples:

Calling `tree(1)` prints:

```
.*.
*+*
.|.
```

Calling `tree(2)` prints:

```
. * .
.+ +.
*+++*
. | .
. | .
```

Calling `tree(4)` prints:

```
.   *   .
.  + +  .
. +   + .
.+     +.
*+++++++*
.  | |  .
.  | |  .
.  | |  .
.  | |  .
```

Calling `tree(5)` prints:

```
.    *    .
.   + +   .
.  +   +  .
. +     + .
.+       +.
*+++++++++*
.   | |   .
.   | |   .
.   | |   .
.   | |   .
.   | |   .
```