

EECS 2500 -- Linear Data Structures

Project 1 -- Stack Implementation and Use

Binary and Decimal Numeric Palindromes

Description:

A palindrome is a word (or, in this case, a number) that reads the same forwards and backwards. In this project you will find numbers that are palindromes in base 10 and check if their binary equivalents are palindromes too. For example, 33 is a decimal palindrome. If you push the digits on a stack they read the same as if they are in order. In this case I want you to store the palindrome in a list and a stack and compare the digits one at a time. Then, assuming it is a palindrome in base 10, I want you to convert it to binary and check if the binary equivalent is also a palindrome. If the number is both a decimal and binary palindrome you will output the number and its binary equivalent on a line to report it.

Details:

The input to your program is a pair of integers that give a range of decimal numbers to check for palindromes. If either of the numbers is negative or the first number is greater than the second number you should print an error message and exit the program. Your program should then check the integers from the first number up to and including the last number looking for numeric palindromes that are palindromes in both base 10 and base 2. If your program locates one, it should output the palindrome in both base 10 and base 2 on a line. This should continue until you have checked the entire range. To convert a number to binary you can use the remainder method that you probably used in digital logic design:

Decimal number: 23

$23 \% 2 = 1$	$23 / 2 = 11$
$11 \% 2 = 1$	$11 / 2 = 5$
$5 \% 2 = 1$	$5 / 2 = 2$
$2 \% 2 = 0$	$2 / 2 = 1$
$1 \% 2 = 1$	$1 / 2 = 0$

Stop -- you've reached zero.

Binary value = 1 0 1 1 1 (noticeably not a palindrome)

Decimal number: 33

$33 \% 2 = 1$	$33 / 2 = 16$
$16 \% 2 = 0$	$16 / 2 = 8$
$8 \% 2 = 0$	$8 / 2 = 4$
$4 \% 2 = 0$	$4 / 2 = 2$
$2 \% 2 = 0$	$2 / 2 = 1$
$1 \% 2 = 1$	$1 / 2 = 0$

Stop

Binary value = 1 0 0 0 0 1 (a palindrome!)

NOTE: You *must* use a stack *you* implement to check your numbers for this project. The stack must be a separate class. The stack class does **not** need to be a generic.

Sample Input and Output 1:

```
Input Range > 1 33
Palindromes:
Base 10      Binary
1            1
3            11
5            101
7            111
9            1001
33           100001
```

Sample Input and Output 2:

```
Input Range > 33 1
Error: First number is larger than second number. Try Again.
Input Range > -1 10
Error: Invalid value entered. All values must be greater than zero. Try Again.
Input Range > 0 1
Error: Invalid value entered. All values must be greater than zero. Try Again.
Input Range > 90 120
Palindromes:
Base 10      Binary
99           1100011
```

Hints:

Work in small pieces.

Do input and check error cases.

Set up code to check a decimal number to see if it is a palindrome.

Set up code to convert a decimal number to binary

Set up code to check a binary number to see if it is a palindrome.

Make a loop to check the entire range from the input given to your program.

Submission:

Your file(s) should be submitted in the location provided on BlackBoard. Your program must have its main method in a class named Project1.