Unit 8 Problem Set

Due on Monday, May 7, before the start of the final exam. If you submit this early (before midnight on May 4), however, your TA may be able to grade your work prior to the final exam.

Before starting on this assignment, you should download and install the latest version of a Mac or Windows (or Linux) version of QtSpim from http://sourceforge.net/projects/spimsimulator/files/

[1] 30 points

Write a MIPS/SPIM assembly language program that prints the smallest and largest values found in a non-empty table of **N** word-sized integers. The address of the first entry in your table should be named **table**, and **N** should be a defined constant.

Test your program out on the following two test cases plus any additional cases that you deem worthwhile:

Test Case #1 (with N equal to 9)

```
table:    3
          -1
          6
          5
          7
          -3
          -15
          18
          2
```

Test Case #2 (with N equal to 1)

```
table:        3
```

[2] 30 points

Modify the **palindrome.asm** program (in the *unit8* directory) so that it ignores whitespace, capitalization and punctuation. Your program must thus be able to recognize the following strings as valid palindromes:

- `Madam, I'm Adam!!!!!!!????`
- `Ed, I saw Harpo Marx ram Oprah W. aside.`
- `A man, a plan, a canal – Panama!`
- `Go hang a salami; I'm a lasagna hog.`
- `"Naomi, sex at noon taxes," I moan.`
- `1 2 321`

Do not "preprocess" the string in advance. Modify the code shown in lecture so that the loop correctly ignores whitespace and punctuation — and deals with upper- and lowercase characters. Check out **http://www.derf.net/palindromes/old.palindrome.html** for more cute ideas ...

[3] 15 points *for graduate-credit students only*

Undergraduate-credit students may answer this problem for "extra credit."

A "perfect number" is a positive integer greater than 1 which is equal to the sum of its *divisors* (except for the original number itself). For example, **6** is perfect, because it is equal to 1 + 2 + 3, and it so happens that 1, 2 and 3 are the only integer values smaller than 6 that divide into 6 "evenly." On the other hand, 12 is not perfect, since $12 \neq 1 + 2 + 3 + 4 + 6$.

Write a MIPS/SPIM assembly language program that will test all integers between **5** and **500**, and print out only the "perfect" ones. By the way, there only 3 numbers that should be output, one of which is **6**.

For 3 points of *additional credit*, construct your solution in such a way that the main program repeatedly calls on a "subroutine" named **perfect**; this subroutine is passed a single integer argument that it checks for the property of being "perfect." A true/false answer of some sort gets returned by this subroutine.