

CMPSC 312

Computer Organization and Architecture
Spring 2017

Due Date: April 19th, 2017 (6:00 p.m.)

Name: _____

Instructions: Please create a MIPS assembly language program to solve the following problem. Submit electronic (Canvas Dropbox) copies of your program source code and **two or more** sample runs to me by the deadline. Submit printouts of your source code and runs to me by the same deadline. Please note that printouts are **required** if you want your submission to be considered for a grade. Documentation requirements follow the problem specification.

1. Write and test a MIPS assembly language program that prompts a user to enter a positive integer, tests if the integer is perfect*, and reports the result of the test. The program begins by prompting the user to enter a positive integer. Next, it prints the proper factors of the integer and the sum of the proper factors, after which it reports if the integer is perfect or not perfect. The program must continue to prompt the user to enter integers to be tested until he or she enters a 0 (the sentinel). Be sure to include error-checking code to deal with the possible entry of negative integers, in the event of which an appropriate error message should be displayed and the user prompted to enter another integer. Your program output should resemble the sample runs at the end of this document.

*An integer number is said to be perfect if it is equal to the sum of its proper factors. The proper factors, or divisors, of a number include all the factors of the number except the number itself. For example, 6 is a perfect number because $1 + 2 + 3 = 6$. 10 is not a perfect number because $1 + 2 + 5 \neq 10$.

Notes:

- The first four perfect numbers are 6, 28, 496, and 8,128. To learn more about perfect numbers, visit <http://mathforum.org/dr.math/faq/faq.perfect.html>.
- Your program **must** use an array to store the factors and it **must** loop.
- The use of procedures is optional.
- Please document **all** sources (Web, in particular) used to complete this project. (Note: I strongly recommend reviewing the Academic Integrity Policy on the CMPSC 312 syllabus.)

Additional Note:

- You may work on this programming project with one of your classmates. If you decide to do so, please submit one electronic copy and one paper copy containing both of your names.

References:

- MIPS resources posted on Canvas (electronic books, lecture slides, demo programs)
-

Documentation Requirements:

- 1) Your program source code file must have a header at the beginning of the program containing the following:
 - Name of author, PSU e-mail address of author, name of course, assignment number and due date, name of file, purpose of program, simulator and operating system used, references
 - Example:

```
# Author:      Wanda Kunkle
# E-mail:      wmk12@psu.edu
# Course:      CMPSC 312
# Assignment:   MIPS Programming Project
# Due date:    4/19/2017
# File:        perfectNumber.s
# Purpose:     This program prompts a user to enter a positive integer, tests
#              if the integer is perfect, and reports the result of the test.
# Simulator:   MARS 4.5
# Operating
# system:      MS Windows 7 Professional
# References:   Class demo programs (include any Web page references here)
```
- 2) You do not have to document every line of your MIPS program, but you may find that doing so helps you to learn the language and write your program. For example:

```
li $v0, 4          # load system call code to print a string
la $a0, prompt     # load address of "prompt"
syscall            # print contents of "prompt"
```

Sample run #1 (*User enters perfect and non-perfect numbers as well as invalid numbers.*):

Please enter a positive integer (0 to quit): 6

The factors are:

1
2
3

The sum is 6

The number is perfect.

Please enter a positive integer (0 to quit): -1

Invalid entry!

Please enter a positive integer (0 to quit): -3

Invalid entry!

Please enter a positive integer (0 to quit): 10

The factors are:

1
2
5

The sum is 8

The number is NOT perfect.

Please enter a positive integer (0 to quit): 28

The factors are:

1
2
4
7
14

The sum is 28

The number is perfect.

Please enter a positive integer (0 to quit): -8

Invalid entry!

Please enter a positive integer (0 to quit): 496

The factors are:

1
2
4
8
16
31
62
124
248

The sum is 496

The number is perfect.

Please enter a positive integer (0 to quit): 24

The factors are:

1
2
3
4
6
8
12

```
The sum is 36
The number is NOT perfect.

Please enter a positive integer (0 to quit): 0

-- program is finished running --
```

Sample run #2 (*User enters one positive integer to test and quits.*):

```
Please enter a positive integer (0 to quit): 8128
The factors are:
1
2
4
8
16
32
64
127
254
508
1016
2032
4064

The sum is 8128
The number is perfect.

Please enter a positive integer (0 to quit): 0

-- program is finished running --
```