# Author: Taras Derewecki

# E-mail: TVD5244@psu.edu

# Course: CMPSC 312

# Assignment: MIPS Programming Project

# Due Date: 04/19/2017

# File: Perfect\_Numbers.asm

# 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: Windows 10 Professional Edition

# References: Class demo programs, (include any Web page references here)

.data

prompt: .asciiz "Enter a positive integer (0 to quit): "

invalid: .asciiz "Invalid Entry!\n"

factors: .asciiz "The factors are: "

sum: .asciiz "The Sum is: "

isNotPerfect: .asciiz "The number is NOT perfect.\n"

isPerfect: .asciiz "The number is perfect.\n"

newline: .asciiz "\n"

factorArray: .space 200

.text

.globl main

main:

start:

li $v0, 4 # Load system call code to print a string

la $a0, prompt # Load address of "prompt"

syscall # Print contents of "prompt"

li $v0, 5 # load system call code to read an integer from the keyboard

syscall # Read the integer into $v0

beqz $v0, exit

bltz $v0, loopNegative

move $s1, $v0 # Copy contents of $v0 into $s1

li $s4, 0 # The sum of all proper divisors of A

li $s5, 1 # Start at 1 to check for divisors

la $s3, factorArray # load address of array

la $a3, factorArray

j loop1

loopNegative:

li $v0, 4 # Load system call code to print a string

la $a0, invalid # Load address of "invalid"

syscall # Print contents of "invalid"

j start

loop1:

bgeu $s5, $s1, eval # while $s5 < $s1

rem $t0, $s1, $s5 # $t0 = $s1 % $s5

bne $t0, $0, loop2 #if rem not equal to zero jump to loop2

addu $s4, $s4, $s5 # $s4 += $s5

sw $s5, ($s3) # store a factor in the array

addi $s3, $s3, 4 #increment address in array

loop2: addi $s5, $s5, 1 # $s5++

j loop1 # endwhile

eval:

li $v0, 4

la $a0, factors # Print factors

syscall

la $a0, newline # load address of "newline"

syscall # print "\n"

addi $t7, $0, 0 #$t7 = 0

sw $t7, ($s3) # storing '0' at the end of array

# Print elements in the factor array

loop3: lw $a0, 0($a3) # load element from array

beq $a0, 0, exit2 # if reached end of array

li $v0, 1 # print factor

syscall

addi $a3, $a3, 4

li $v0, 4

la $a0, newline # load address of "newline"

syscall

j loop3

exit2:

li $v0, 4

la $a0, newline # load address of "newline"

syscall

li $v0, 4 # Load system call code to print a string

la $a0, sum # load address of "sum"

syscall

move $a0, $s4 #print sum of factors

li $v0,1

syscall

li $v0, 4

la $a0, newline # load address of "newline"

syscall

sub $t6, $s1, $s4

bnez $t6 NOT

li $v0, 4 # Load system call code to print a string

la $a0, isPerfect

syscall

j start

NOT:

li $v0, 4 # Load system call code to print a string

la $a0, isNotPerfect

syscall

j start

exit:

li $v0, 10

syscall

**SAMPLE RUN: 1**

Enter a positive integer (0 to quit): 6

The factors are:

1

2

3

The Sum is: 6

The number is perfect.

Enter a positive integer (0 to quit): -1

Invalid Entry!

Enter a positive integer (0 to quit): -3

Invalid Entry!

Enter a positive integer (0 to quit): 10

The factors are:

1

2

5

The Sum is: 8

The number is NOT perfect.

Enter a positive integer (0 to quit): 28

The factors are:

1

2

4

7

14

The Sum is: 28

The number is perfect.

Enter a positive integer (0 to quit): -8

Invalid Entry!

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.

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.

Enter a positive integer (0 to quit): 0

-- program is finished running –

**SAMPLE RUN: 2**

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.

Enter a positive integer (0 to quit): 0

-- program is finished running --