1 ## 2 ## This program is written to find out the number of vowels and the number of

non-vowels in a given string 3 ## and use procedure call to check if a letter is vowel or not. 4 ## 5 ## - It will have initilised string name "example" that 6 ## holds the given example 7 ## - then call a procedure, vowelp to check if it is a vowel. 8 ## - It will return a value to the calling function, 9 ## 0 means consonent, and 1 means vowel, 10 ## - and then print number of vowels and consonents. 11 ## 12 ## v0 - used for syscalls also holds 0 or 1 after the call to "vowelp" 13 ## t0 - holds example 14 ## t1 - vowels counter 15 ## t2 - consonents counter 16 ## t3 - holds the returned value from the procedure call 17 ## a0 - holds strings also to give letter to vowelp 18 ## 19 ## 20 21 ################################################# 22 # # 23 # text segment # 24 # # 25 ################################################# 26 27 .text 28 .globl \_\_start 29 \_\_start: # execution starts here 30 31 32 la $t0, example # loading address of string 33 li $t1, 0 # vowel count 34 li $t2, 0 # consonent count 35 36 loop: 37 lb $a0,($t0) # loading single character from example 38 beqz $a0, End # if end of string then jimp to End 39 addi $t0,$t0, 1 # or else move character by character in example 40 jal vowelp # procedure call 41 move $t3,$v0 # saving value temporarily 42 bnez $t3, plusVowel # If return value is 1 than the character is vowel 43 j plusConso # else jump to plusConso 44 45 plusConso: 46 add $t2,1 #increment consonent count 47 j loop #reenter loop 48 49 plusVowel: 50 addi $t1,$t1,1 #increment vowel count 51 j loop #reenter loop 52 53 End: 54 la $a0, ans1 #syscall to print 55 li $v0, 4 #result 56 syscall 57 58 move $a0,$t1 59 li $v0, 1 #print number of vowels 60 syscall 61 62 la $a0, endl #syscall to print out 63 li $v0, 4 #a new line 64 syscall 65 66 la $a0, ans2 #syscall to print out 67 li $v0, 4 #result 68 syscall

69 70 move $a0,$t2 71 li $v0,1 #print number of consonents 72 syscall 73 74 la $a0,endl #syscall to print out 75 li $v0,4 #a new line 76 syscall 77 78 li $v0,10 79 syscall # Bye! 80 81 ############################################################# 82 # # 83 # FUNCTION: vowelp # 84 # # 85 # - checks for vowels if they are vowel or not # 86 # # 87 ############################################################# 88 89 vowelp: 90 91 li $v0,0 92 beq $a0,'A',yes 93 beq $a0,'a',yes 94 beq $a0,'E',yes 95 beq $a0,'e',yes 96 beq $a0,'I',yes 97 beq $a0,'i',yes 98 beq $a0,'O',yes 99 beq $a0,'o',yes 100 beq $a0,'U',yes 101 beq $a0,'u',yes 102 jr $ra 103 yes: 104 addi $v0,$v0, 1 #If its vowel than return 1 105 jr $ra 106 107 108 ################################################# 109 # # 110 # data segment # 111 # # 112 ################################################# 113 114 .data 115 example: .asciiz "I am going to run this program with the real hardware in two

weeks." 116 ans1: .asciiz "Vowels are : " 117 endl: .asciiz "\n" 118 ans2: .asciiz "Consonents are : " 119 120 # end of file num-vowel.s