Question 1

1. The register do not get updated correctly within the program and we are missing a move operator so that we can use the new line subprogram.
2. To fix the problem and ensure correct behavior, you need to ensure that the **PrintNewLine** subprogram receives the correct argument in **$a0** hear is the correct
3. solution:

.text

PrintInt:

# Print string. The string address is already in $a0

li $v0, 4

syscall

# Save the integer value in a temporary register

move $t0, $a1

# Print integer. Move integer value to $a0

move $a0, $t0

li $v0, 1

syscall

# Print a new line character using PrintNewLine

move $a0, $t0 # Restore integer value in $a0

jal PrintNewLine

# Return

jr $ra

Question 2 next page

Questions (2)

Each subprogram should have its own .text first thing when you make the sub-program but there is no .text above the (PrintTab:) line so the solutions is: (also it is better to have a heading before each sub-program)

# Subprogram: PrintNewLine  
# Author: Charles Kann  
# Purpose: to output a new line to the user console  
# Input/Output: None  
# Side effects: A new line char is printed to the console  
.text  
PrintNewLine:  
li $v0, 4  
la $a0, \_\_PNL\_newline  
syscall  
jr $ra  
.data  
\_\_PNL\_newline: .asciiz "\n"

.text  
PrintTab:  
li $v0, 4  
la $a0, tab  
syscall  
jr $ra  
.data  
tab: .asciiz "\t"