

Quiz 2, R02

Name _____

SBID# _____

Problem 1 (7 points) Consider a program in java:

```
int i=0;
While(a[i]>k) {
    i=i+1;
}
print( i);
```

Write a MIPS assembly program that implements the above code. Your code should be such that, if we know that $a[4] = k-2$, then the code will not execute more than 23 instructions at most in total.

Assume that the address of array a is stored in base hex address $0x4314DDE1$, and k is stored in register $\$s5$. (You can't use the `.data` segment to load the address of a).

Solution:

```
    lui $t5, 0x4314
    ori $t5, $t5, 0xDDE1
    lw $t0, 0($t5)
    addi $a0, $zero, 0
    ble $t0, $s5, Exit
Loop:    addi $a0, $a0, 1
        addi $t5, $t5, 4
        lw $t0, 0($t5)
        bgt $t0, $s5, Loop
Exit:    li $v0, 1
        syscall
```

Problem 2 (3 points) The following program reads an input string symbol

by symbol, omits the 'a' characters, and prints the output.

```
    la $t0, Input      # $t0 contains address of an input string
Loop: lb $a0, ($t0)    # load byte into $a0
    beq $a0, '¥0', Exit # if $a0 is the end of the string goto Exit
    beq $a0, 'a', Inc   #if $a0 is equal to 'a', do not print.
    li $v0, 11         # syscall 11 prints a character
    syscall            # stored in $a0
Inc:  addi $t0, $t0, 1  # move pointer to the next symbol
    j loop

    Exit:
```

An example of input in the .data segment:

```
Input: .asciiz "abc"
```

Rewrite the program to such that it changes every small 'a' character to a big 'A' character, and prints the outcome string. For example, if the input is "abc" the program should print out "Abc". (Note: be careful not to print the '¥0' symbol.) Your program should work for any input string, not just for the specific string given as an example above.

(Hint: For instructions which accept numerical values as part of their operands (immediate instructions, conditional branches), you could refer to characters by simply writing them between to single quotations 'a' , for example:

```
addi $t5, $t5, 'A'
```

is acceptable and is treated as:

```
addi $t5, $t5, 65
```

But remember that when you do so, then you should print the content of such register as a character in order to get your required result!)

Solution:

```
    la $t0, Input
Loop: lb $a0, 0($t0)
    beq $a0, '¥0', Exit
    beq $a0, 'a', L
```

```
Print:    li $v0, 11
          syscall
          addi $t0, $t0, 1
          j Loop
L:        addi $t5, $t5, 'A'
          move $a0, $t5
          j Print
Exit:
```