

Quiz 2, R01

Name _____

SBID# _____

Problem 1 (3 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 $a[4] > k$, then the code will not execute more than 25 instructions at most in total.

Assume that the address of array a is stored in register $\$t1$, and k is stored in register $\$s5$.

Solution:

```
    addi $a0, $zero, 0
    lw $t0, 0($t1)
    bge $t0, $s5, Exit
Loop:    addi $a0, $a0, 1
        addi $t1, $t1, 4
        lw $t0, 0($t1)
        blt $t0, $s5, Loop
Exit:    li $v0, 1
        syscall
```

Problem 2(3points) Suppose that we have a base address with HEX value, 0x3DE10089. And in that address we have stored a variable t . Write a program that prints out the value of $51 \times t$ without using multiplication instructions. (You can't use the .data segment to load the address of t).

Solution:

```

    lui $t0, 0x3DE1
    ori $t0, $t0, 0x0089
    lw $t1, 0($t0)
    add $t2, $zero, $zero
    addi $t4, $zero, 51
    addi $t3, $zero, 0
Loop:   add $t3, $t3, $t1
        addi $t2, $t2, 1
        blt $t2, $t4, Loop
        move $a0, $t3
        li $v0, 1
        syscall
Exit:

```

Problem 3 (4 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:

```

Exit:

An example of input in the .data segment:

Input: .asciiz "abc"

Rewrite the program to print the reversed input string, changing every 'a' into a 'b' character. For example, if the input is "abc" the program should print out "cbb". (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.

Solution:

```

    la $t0, Input
    add $t1, $t0, $zero
Loop: lb $t2, 0($t1)

```

```
    beq $t2, '¥0', Print
    addi $t1, $t1, 1
    j Loop
Print:    addi $t1, $t1, -1
    blt $t1, $t0 Exit
    lb $a0, ($t1)
    beq $a0, 'a', Chgb
    li $v0, 11
    syscall
    j Print
Chgb: addi $t5, $zero, 'b'
    move $a0, $t5
    li $v0, 11
    syscall
    j Print
Exit:
```