

Quiz 2, R03

Name _____

SBID# _____

Problem 1 (3 points) Consider a program in java:

```
int i=2;
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 13 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, 2
    lw $t0, 0($t1)
    ble $t0, $s5, Exit
Loop:    addi $a0, $a0, 1
        addi $t1, $t1, 4
        lw $t0, 0($t1)
        bgt $t0, $s5, Loop
Exit:    li $v0, 1
        syscall
```

Problem 2 (4 points) The following program reads an input string symbol by symbol, 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
    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:

Rewrite the program to load the value of variable t whose address is 0x3DE10089, if t is greater or equal to the length of your input string, print 1; otherwise print the value of t.

Solution:

```

        lui $t5, 3DE1
        ori $t5, $t5, 0x0089
        lw $t6, 0($t5)
        la $t0, Input
        add $t1, $t0, $zero
        addi $t2, $zero, 0
Loop:  lb $t2, ($t1)
        beq $t2, '¥0', chk
        addi $t1, $t1, 1
        addi $t2, $t2, 1
        j Loop
chk:   bge $t2, $t6, L
        move $a0, $t6
        li $v0, 1
        syscall
        j Exit
L:     addi $t7, $t7, 1
        move $a0, $t7
        li $v0, 1
        syscall
Exit:

```

Problem 3(3points) Write a program that reads an input string symbol by symbol, omits every 'a' character, and prints the reversed outcome.

An example of input in the .data segment:

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

Your program should print out "cb".

(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
    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 $a0,0($t1)
    beq $a0,'a',Omit
    li $v0,11
    syscall
    j Print
Omit: j Print          #omit 'a'
Exit:
```