Amrita Vishwa Vidyapeetham

Amrita School of Engineering, Coimbatore

Department of Computer Science and Engineering

Practice Question

Sub Code: 19CS211 **Sub Title: COA**

Roll No: CB.EN.U4CSE19453 **Name: R.ABHINAV**

Practice Lab **Date: 26-04-2021**

---

1) Write a program in MIPS that contains a procedure which takes one argument - an integer greater than or equal to zero which specifies which element of the Fibonacci sequence is to be returned.The procedure must be recursive and return the correct Fibonacci value. Your final Fibonacci value should be stored in the register $t1.

Code: (done factorial)

```
# Here is a recursive implementation of factorial, first in C, then in assembly:
# int factorial (int n){
#     if (n < 2) return 1;
#     return (n * factorial (n-1));  /* n! = n * (n-1)! */
# }



.text
.globl main

factorial:
   bgtz  $a0, doit
   li   $v0, 1      # base case, 0! = 1
   jr   $ra
doit:
   sub  $sp,8          # stack frame
   sw   $s0,0($sp)     # will use for argument n
   sw   $ra,4($sp)     #  return address

   move $s0, $a0       #  save argument
```

```
    sub  $a0, 1          #  n-1
    jal  factorial       #  v0 = (n-1)!
    mul  $v0,$s0,$v0     #     n*(n-1)!

    lw   $s0,($sp)       # restore registers from stack
    lw   $ra,4($sp)
    add  $sp,8
    jr   $ra
main:
    li   $a0, 7          # set the argument for the factorial function to 7
    sub  $sp, 4          # create the stack frame
    sw   $ra,0($sp)      # save the return address
    jal  factorial       # call factorial
    move $t1, $v0        # save the return value
    lw   $ra,0($sp)      # restore the orignal return address
    add  $sp,4
    jr   $ra
```

2) Write a MIPS program that given a number N and N integers can print the integers in a sorted order using Bubble Sort. Bubble Sort algorithm involves swapping of two numbers. Write a procedure for swapping two numbers separately and use it in the sort function.

Code :

```
.text
   .globl main
main:     la  $a0, Array
loop:     lw  $t0, 0($a0)
          lw  $t1, 4($a0)
          blt $t1, $t0, swap
          addi   $a0, $a0, 4
          j   loop
swap:     sw  $t0, 4($a0)
          sw  $t1, 0($a0)
          li  $a0, 0
          j   loop

          .data

Array:    .word   14, 12, 13, 5, 9, 11, 3, 6, 7, 10, 2, 4, 8, 1
```

3) Write a MIPS program to convert a user given integer to a binary number. Consider both the positive and negative integers. In case of a negative integer, the output has to be in 2's complement form. Print the binary number as a string.

Practice problems :

1.

```
# A demonstration of some simple MIPS instructions
# used to test QtSPIM

        # Declare main as a global function
        .globl main

        # All program code is placed after the
        # .text assembler directive
        .text

# The label 'main' represents the starting point
main:
        li $t2, 25          # Load immediate value (25)
        lw $t3, value       # Load the word stored in value (see bottom)
        add $t4, $t2, $t3   # Add
        sub $t5, $t2, $t3   # Subtract
        sw $t5, Z           #Store the answer in Z (declared at the bottom)

        # Exit the program by means of a syscall.
        # There are many syscalls - pick the desired one
        # by placing its code in $v0. The code for exit is "10"
        li $v0, 10 # Sets $v0 to "10" to select exit syscall
        syscall # Exit

        # All memory structures are placed after the
        # .data assembler directive
        .data

        # The .word assembler directive reserves space
        # in memory for a single 4-byte word (or multiple 4-byte words)
        # and assigns that memory location an initial value
        # (or a comma separated list of initial values)
value:  .word 12
Z:      .word 0
```