

# Recursion and Homework 2

Homework 2 Due Friday March 13

```
int fact (int n)
{
    if (n < 1) return (1);
    else return (n * fact(n - 1));
}
```

Unn Fig. 27

Fact:

```
addi $sp, $sp, -8
```

```
sw $ra, 4($sp)
```

```
sw $a0, 0($sp)
```

```
slti $t0, $a0, 1  
beq $t0, $zero, Rec  
li $v0, 1  
addi $sp, $sp, 8  
jr $ra
```

Rec:

addi \$a0, \$a0, -1

jal Fact

lw \$a0, 0(\$sp)

lw \$ra, 4(\$sp)

addi \$sp, \$sp, 8

mul \$v0, \$a0, \$v0

jr \$ra

Fact:

```
addi $sp, $sp, -8
sw $ra, 4($sp)
sw $a0, 0($sp)
slti $t0, $a0, 1
beq $t0, $zero, Rec
li $v0, 1
addi $sp, $sp, 8
jr $ra
```

Rec:

```
addi $a0, $a0, -1
jal Fact
lw $a0, 0($sp)
lw $ra, 4($sp)
addi $sp, $sp, 8
mul $v0, $a0, $v0
jr $ra
```

# Homework 2

- Implement the four subroutines discussed on the following slides and a main code segment that uses the subroutines and let's the user enter a non-negative integer,  $n$ , and prints the  $n$ th Fibonacci number

# readInt

- Implement a subroutine that prompts the user to enter an integer and reads the integer entered by the user. The address of the string to use as the prompt is passed to the subroutine in register \$a0 and the integer entered is returned in register \$v0



# readNonNegInt

- This subroutine uses the readInt subroutine and checks that the value returned by readInt is non-negative. If the value returned by readInt is negative, the subroutine continues to call readInt until a non-negative value is returned. The address of the initial prompt is passed in \$a0 and the address of the prompt used when the user entered a negative value is passed in \$a1

# printInt

- This subroutine prints a label and an integer. The address of the label is passed in \$a0 and the integer is passed in \$a1

# Fibonacci

- This subroutine implements MIPS code for the recursive Fibonacci function shown on the following page. The subroutine returns the  $n$ th Fibonacci number. The value of  $n$  is passed in register  $\$a0$ .
- For this homework the Fibonacci sequence begins with 0. The sequence 0, 1, 1, 2, 3, 5, 8, 13 ... shows the 0th, 1st, 2nd, 3rd, 4th, ... Fibonacci numbers.

```
int fib(int n) {  
    //PRE: n >= 0  
    if (n <= 1) return n;  
    return fib(n-1) + fib(n-2);  
}
```

# Example Executions

## Example 1 (user input in bold)

Which Fibonacci number do you want? **10**

The Fibonacci number is 55

## Example 2

Which Fibonacci number do you want? **-20**

You must enter an integer greater than or equal to 0. Try again. **-10**

You must enter an integer greater than or equal to 0. Try again. **20**

The Fibonacci number is 6765

# Homework 2 Submission

- email me ([tgendreau@uwlax.edu](mailto:tgendreau@uwlax.edu)) only one file called h2.asm
- Put a comment in the first line of the file that includes your name
- You do not need to comment each line of code but you should have comments that explain the basic flow of the code