

Assignment 5

CSIS 2810- Computer Architecture

Due: Once we finish Chapter 2

Important Notes:

- Solutions turned in must be your own. Please, mention references (if any) at the end of each question.
1. Nested Procedures. The following is a pseudo code for performing the Fibonacci series up to n terms using nested functions. **(20 points)**

BEGIN

FUNCTION sum(a, b):

 DECLARE s

 s = a + b

 RETURN s

END FUNCTION

FUNCTION bonacci(n):

 DECLARE a = 0

 DECLARE b = 1

 FOR i FROM 0 TO n-1 DO:

 DECLARE temp

 temp = sum(a, b)

 a = b

 b = temp

 END FOR

 RETURN a

END FUNCTION

MAIN:

 PROMPT "Enter the number of terms: "

 READ n

 DECLARE nth_term = bonacci(n)

 PRINT "The nth term is: " + nth_term

END MAIN

END

Explanation:

1. **sum(a, b)**: Computes the sum of two integers.
2. **bonacci(n)**: Simulates the Fibonacci-like sequence where each number is the sum of the two preceding ones. It loops *n* times to calculate the *n*th term.
3. **main**: Reads an integer *n* (the number of terms), calls the `bonacci` function, and prints the *n*th term of the sequence.

2. **Recursive Function.** Write an MIPS program and execute it in MARS to recursively reverse an array of integers. Submit your `.asm` file with comments included. The following is the pseudo code for the program. **(20 points)**

BEGIN

FUNCTION reverseArray(arr, start, end):

 DECLARE temp

 IF start >= end THEN

 RETURN

 END IF

 // Swap the elements

 temp = arr[start]

 arr[start] = arr[end]

 arr[end] = temp

 // Recursive call to reverse the rest of the

array

 reverseArray(arr, start + 1, end - 1)

END FUNCTION

FUNCTION printArray(arr):

 FOR i FROM 0 TO LENGTH(arr) - 1

DO:

 PRINT arr[i] + " "

END FOR

 PRINT NEW LINE

END FUNCTION

MAIN:

DECLARE arr = [1, 2, 3, 4, 5, 6]

PRINT "Original array:"

CALL printArray(arr)

CALL reverseArray(arr, 0, LENGTH(arr)

- 1)

PRINT "Reversed array:"

CALL printArray(arr)

END MAIN

END

Explanation:

1. **reverseArray(arr, start, end):** This function recursively swaps the elements of the array to reverse it. The recursion continues until the `start` index is greater than or equal to the `end` index.
2. **printArray(arr):** This utility function prints all elements of the array on a single line, followed by a newline.
3. **MAIN:** The main function initializes an array, prints the original array, calls the `reverseArray` function to reverse it, and then prints the reversed array.

Implement the above pseudo-code in MIPS assembly, execute it in the MARS simulator, and submit your .asm file and video explaining the code and displaying the expected output.

Make sure you add comments next to every instruction.