



**THE UNIVERSITY OF KANSAS**

**SCHOOL OF ENGINEERING**

**DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE**

EECS 645 – Computer Architecture

Spring 2021

Homework 08 (MARS)

Student Name:

Student ID:

---

## Homework 08

In this homework you are required to write a routine that generates the Fibonacci sequence  $fib\_array(0)$ ,  $fib\_array(1)$ ,  $fib\_array(2)$ , ... ,  $fib\_array(n)$ , using **only basic, i.e., not pseudo, MIPS assembly instructions**. When given the arguments  $n$  in register  $\$a0$ , and the base address  $\&fib\_array$  in register  $\$a1$ , your routine should generate the Fibonacci sequence and store it into the data segment starting at address  $0x100100A0$ .

For example, when  $n = 18$  your code should save  $fib\_array(0)$  at address  $0x100100A0$ ,  $fib\_array(1)$  at address  $0x100100A4$ ,  $fib\_array(2)$  at address  $0x100100A8$ , ... ,  $fib\_array(17)$  at address  $0x100100E4$ , and  $fib\_array(18)$  at address  $0x100100E8$ .

**Implement** your routine using **two different methods** as follows:

- a) **Non-recursively** as described by the following pseudo code:

```
unsigned int fib_seq(unsigned int n, &fib_array) {
    fib_array[0] = 0;
    if (n>=1) fib_array[1] = 1;
    for i = 2 to n
        fib_array[i] = fib_array[i-1] + fib_array[i-2];
    return fib_array[n];}
```

- b) **Recursively** using either one of the following two options:

- **Option 1** – a caller that calls the recursive routine “`unsigned int fib(unsigned int n)`” from *Problem 2.31* (in chapter 2 of the second textbook) from *HW07*:

```
unsigned int fib_seq(unsigned int n, &fib_array) {
    for i = 0 to n
        fib_array[i] = fib(i);
    return fib_array[n];}
```

where

```
unsigned int fib(unsigned int n) {
    if (n==0 || n==1) return n;
    else return fib(n-1) + fib(n-2);}
```

- **Option 2** – a caller that recursively calls itself as follows:

```
unsigned int fib_seq(unsigned int n, &fib_array) {
    fib_array[0] = 0;
    if (n>0) {
        fib_array[1] = 1;
        if (n>1) fib_array[n] = fib_seq(n-1, fib_array) + fib_seq(n-2, fib_array);
    }
    return fib_array[n];}
```

**Hint:** For option 1, you could use the solution of *Problem 2.31* from *HW07* to implement the *recursive* version of your code.

**Note:** The function in *Problem 2.31* returns one Fibonacci number  $fib(n)$ , while here you are required to generate a sequence of Fibonacci numbers, i.e.,  $fib(0)$ ,  $fib(1)$ ,  $fib(2)$ , ... ,  $fib(n)$ .

### Steps:

- 1) Download the file “HW08\_MARS.zip” from blackboard and extract its contents.
- 2) Launch MARS, “Mars4\_5.jar”.
- 3) Open and edit the template source files:  
“\HW08\_MARS\fibonacci\_sequence\_non-recursive.asm”, and  
“\HW08\_MARS\fibonacci\_sequence\_recursive.asm”
- 4) Verify the correctness of your code by checking the content of the data segment starting from address  $0x100100A0$ .