Computer Architecture Course: IT089IU

International University – VNU HCM Date: March 2025

Dr. Le Hai Duong & Dr. Ly Tu Nga Time: 4 hours

**Laboratory Session 4**

**Procedure**

1. **Exercise 1: Int2Hex Converter. (40pts)**

**Write a program that**

* 1. Read in **ONE** unsigned integer in the range 0 to 15. Print out that number in hexadecimal. For example, given the input 13, print out 0xD. (**lab4\_1\_1.s**)
  2. Modify the previous assembly, create a procedure printHex(int num). This procedure takes in a number and print it out in hexadecimal. (**lab4\_1\_2.s**)
  3. Modify the previous assembly so that it can print out hexadecimal of A{yourdigitID}. For example, read in number A23023, print out 0xA23023. (**lab4\_1\_3.s**)

1. **Exercise 2: Fibonacci number (40 pts)**

The Fibonacci series is defined as:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Each term in the series is the sum of the preceeding two terms. So, for example, the term 13 is the sum of the terms 5 and 8.

2.1 Write the program as a counting loop that terminates when the first 100 terms of the series have been computed. Use a register for the current term and a register for the previous term. Each execution of the loop computes a new current term and then copies the old current term to the previous term register. Print out the series. (Use no function) (**lab4\_2\_1.s**)

2.2 Rewrite the program using function

void fibonacci(int numOfTerms); // this is C prototype

This function print out the Fibonacci series with one input is the number of terms numOfTerms.

2.3 Rewrite the program using recursion. The fib function should compute the nth term as such:

int fib (int N) {

return return fib(N – 1) + fib(N – 2);

}

1. **Exercise 3 (20points)**

Write a program to extract a slice from an array given a start index and length.

**Example:** Given an **array [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]**

Extract a **slice** from **index 2 to index 5** (elements [30, 40, 50, 60]).

Store the result in a new array.

**Pseudo Code**

1. **Load start index and slice length** from memory.
2. **Calculate the starting address** in the array (start\_idx \* 4 to get byte offset).
3. **Iterate over the specified range**:
   * **Load elements** from array[start\_idx].
   * **Store them** in slice[].
   * **Advance pointers** for both arrays.
   * **Decrement the counter** until all elements are copied.
4. **Exit program**.

**Reference:**

1. <https://en.wikibooks.org/wiki/MIPS_Assembly/Pseudoinstructions>
2. <https://courses.missouristate.edu/KenVollmar/MARS/Help/SyscallHelp.html>
3. <https://www.assemblylanguagetuts.com/mips-assembly-programming-tutorials/#MIPS_Data_Types>
4. <https://en.wikibooks.org/wiki/MIPS_Assembly/Arithmetic_Instructions>
5. <https://gab.wallawalla.edu/~curt.nelson/cptr280/lecture/mips%20arithmetic%20instructions.pdf>