

# Indian Institute of Technology Kharagpur

AUTUMN Semester, 2020-2021

COMPUTER SCIENCE AND ENGINEERING

CS39001: Computer Organization Laboratory

Assignment-7: MIPS-32 Assembly Language Programming

Full Marks: 30

Time allowed: 6 hours

**INSTRUCTIONS: ATTEMPT ALL PROBLEMS.** Make one submission per group of your source code on Moodle. Name your submitted source files following the format `Assgn_7_Prob_1_Grp_<Group_no>.s` (e.g. `Assgn_7_Prob_1_Grp_25.s`), etc. Inside each submitted file, there should be a clear header describing the assignment no., problem no., semester, group no., and names of group members. Liberally comment your code to improve its comprehensibility.

1. **[String Operation in MIPS-32]** Write a complete MIPS-32 program to convert a null-terminated ASCII string stored in memory, possibly containing a mixture of upper-case letters, lower-case letter, numbers and spaces, to lower-case, and store it back to memory. First store the input string in memory by creating space for it using the `.space` directive (you may assume a maximum length of the string) and collecting the string from the user, or hard-code it in your program using the `.asciiz` directive. After conversion, store back the converted string starting at an address created by the `.space` directive (remember: each character in an ASCII-encoded string occupies exactly one byte in memory). You may or may not use MIPS procedure call to solve the problem. After conversion, print the converted string on the console, accompanied by a proper message. (10 marks)
  2. **[Insertion Sort in MIPS-32]** Write a MIPS-32 program to collect an array of eight integers from the user, and then sort the array in ascending order using the *Insertion Sort* algorithm. Collect the numbers from the input console using a loop, and store in memory in an array called `array`. Do not store the numbers as scalars in eight different non-contiguous locations or in eight different registers. Write a function `InsertionSort` to implement the sorting algorithm. Pass the address of the 1-D array and the required parameters while implementing the function. After sorting, print the sorted array on the console with a proper message. (10 marks)
  3. **[Recursive Binary Search in MIPS-32]** Write a MIPS-32 program to collect an array of nine integers from the user, sort the array in ascending order using the *Insertion Sort* algorithm you have implemented in Problem-2, and then search the array for the presence of a value in the array, using *Recursive Binary Search*. The value to be searched in the array is also to be entered by the user. If the search is successful, the program will print an appropriate success message, with the array index where the value was found. If the search is unsuccessful, the program will print a failure message. **No credit will be given for an iterative (linear) search implementation.** Your program should have a recursive function `BinarySearch` to implement the sorting algorithm. After sorting the array containing the numbers, pass the address of the sorted 1-D array and the required parameters to implement the `BinarySearch` function. (10 marks)
-