

Indian Institute of Technology (IIT-Kharagpur)

AUTUMN Semester, 2021

COMPUTER SCIENCE AND ENGINEERING

Computer Organization and Architecture Laboratory

MIPS Assignment 1

September 1, 2021

AIM: To get acquainted with MIPS assembly language and the system calls. Partial marks will be awarded for incorporating interactive interface as specified, appropriate use of system calls for printing and taking inputs, suitable commenting and correct implementation of the logic.

INSTRUCTIONS: Make one submission per group in the form of a single zipped folder containing your source code(s). Name your submitted zipped folder as Assgn_2_Grp_GroupNo.zip and (e.g. Assgn_2_Grp_25.zip). Inside each submitted source files, 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.

Question 1 (Tutorial: Do not submit the solution)

Write a complete MIPS-32 program that -

1. Reads one positive integer ($n > 0$) with the prompt – “Enter a positive integer:”. (After the input number is collected from the user, there should be sanity checking to ensure that the integer is positive.)
2. Calculate the sum of all integers from 1 to n (including n).
3. Accordingly, print the calculated sum with the message – “The sum of the first $< n >$ integers is ”

Steps to follow in QtSpim:

1. Write your MIPS code in a text file with “.s” extension.
2. Open QtSpim and load the $< filename.s >$ file.
3. You can run the code by clicking on the “play” button at the top.

4. Put your inputs in the console. The output will be displayed.
5. To debug, you can execute the program in a single step fashion and reading the intermediate values in the register.
6. Every time you execute the program, you need to reload the $\langle filename.s \rangle$ file.

Question 2

Write a complete MIPS-32 program which:

1. Reads two positive integers with the prompt – “Enter the first positive integer:” and “Enter the second positive integer:”. (After the input numbers are collected from the user, there should be sanity checking to ensure that the integers are positive.)
2. Calculate their GCD by repeated subtraction as given in the below algorithm (Algorithm 1). DO NOT USE ANY INTEGER DIVISION INSTRUCTION, EVEN IF IT IS AVAILABLE IN MIPS-32.

Algorithm 1 Find GCD of two numbers

1: procedure GCD(a, b)	▷ The GCD of a and b
2: if $a == 0$ then return b ;	
3: while $b \neq 0$ do	
4: if $a > b$ then $a = a - b$;	
5: else $b = b - a$;	
6: return a	

3. Accordingly, print the calculated GCD with the message – “GCD of the two integers is: ”.

Question 3

Write a complete MIPS-32 program which:

1. Reads a positive integer $n \geq 10$ with the prompt – “Enter a positive integer greater than equals to 10:”. (After the input number is collected from the user, there should be sanity checking to ensure that the integer is ≥ 10 .)
2. Determines whether it is prime or composite. Prime numbers are considered as the numbers which retain only two factors i.e. one and itself. The integer which can be obtained by multiplying the two smallest positive numbers and includes at least one divisor other than 1 is known as composite numbers.

3. Display the finding of your program with a proper message at the end by printing “Entered number is a PRIME (or COMPOSITE) number.”.

Question 4

Write a complete MIPS-32 program which:

1. Reads an integer with the prompt – “Enter a positive integer:”. (After the input number is collected from the user, there should be sanity checking to ensure that the integer is positive.)
2. Checks whether it is a perfect number. A perfect number is equal to the sum of its proper divisors (including 1 and excluding itself). The smallest perfect number is 6, which is the sum of 1, 2, and 3. Some other perfect numbers are 28, 496, and 8128.
3. Accordingly, prints the message – “Entered number is (not) a perfect number.”.