**COE 301/ ICS 233, Term 172**

**Computer Architecture & Assembly Language**

**Programming Assignment# 3**

1. Names and IDs

* Yousef Khalid Majeed 201568070
* Bader Abed Almazmumi 201480600

1. Assignment Number: 3
2. Problem statement:

**Q.1.** It is required to write a MIPS assembly program that does the following:

1. Ask the user to enter the base of an input number (2-16). Your program should give an error if the entered base is outside the given range and ask the user to reenter the base.
2. Read the entered number in the specified base. Your program should report an error message if an invalid digit is entered. Write a procedure for reading a number in a given base.
3. Ask the user to enter the base he wants the number to be converted to (2-16). Your program should give an error if the entered base is outside the given range and ask the user to reenter the base.
4. Print the number in the required base. Only the significant digits have to be printed. Write a procedure for printing the number in the required base.

**Q.2.** The greatest common divisor (GCD) of two integers can be computed recursively as follows:

**int** GCD (a, b){ **if** (b == 0) **return** a **else**

**return** GCD (b, a % b)

}

1. Write a recursive MIPS function to compute the GCD of two integers. Assume that the two integers numbers will be passed in $a0 and $a1 and that the GCD values will be returned in $v0. Use MIPS programming convention in preserving registers.
2. Write a MIPS program to ask the user to enter two integer numbers, read then and then display their GCD by calling the implemented function in (i).

1. The solution along with the code:
2. Discussion:

The difficulty we faced is how to call a function from another function, but we thought for three hours and half and we solved it correctly (we think!) without any problems using the stack segment to store the values in and load it back if we want it again. It was very interesting moment after we know how to solve it. Eventually we learned a lot from this awesome homework and it was neither a hard nor an easy actually in between (we think also!).