

School of Computer Science
McGill University

Available On: Sunday October 8th, 2017.

Due Date: Friday October 20th, 2017, by 11:59 pm.

Hand in your solutions using *mycourses*, following the instructions below. For each problem use the provided template .asm (text) file as a starting point. **(Late policy: 25% per day late, with no submissions accepted after Sunday October 22nd.)**

1 Least Common Multiple (20 marks)

Write a MIPS procedure `lcm` that accepts two integers as arguments and returns their least common multiple. That is, given a and b , the procedure should find the smallest integer which is a multiple of both numbers. For example the least common multiple of 4 and 6 is 12. You can use the `div` instruction with suitable supporting instructions for this problem.

Write a `main` procedure which calls test your function with two hard-coded arguments, and then prints out the result.

2 String Capitalization (30 marks)

You are now required to write a MIPS program that reads a string from standard input, and then prints the same string but with each word capitalized. You can assume that the string input by the user is comprised of words consisting of sequential keyboard characters, with each word separated by a single “space” character and with each word beginning with an alphabetical character. The program should be able to handle strings up to 127 characters in length. No characters other than lowercase letters should be modified. Use properties of the ASCII code to solve this problem.

Implement the conversion operation in a subroutine named `upper`, which accepts the address of the input string as its only argument.

Create a `main` procedure which prompts for input from standard input, calls your procedure, and then prints the output.

As a working example, given the input string “the c12a:t in Th*2 hat” the printed output should read “The C12a:t In Th*2 Hat”.

3 Dijkstra’s GCD Algorithm [50 marks]

Dijkstra created an algorithm for finding the greatest common divisor (GCD) of a number based on the principle that if a number c divides a and b , then it divides $a - b$. Below is the algorithm in C:

```

int gcd(int m, int n){
    if(m==n)
        return m;
    else if(m>n)
        return gcd(m-n, n);
    else
        return gcd(m,n-m);
}

```

Create a subroutine, “gcd”, that implements this algorithm. Note that this must be a recursive subroutine.

Also create a short “main” procedure to test the subroutine. The “main” procedure should prompt the user for two integers (to be typed in using standard input) and should print out the GCD of these numbers.

IMPORTANT

1. For each of the above problems we have provided a template .asm (text) file for you to use as a starting point.
2. Comment your code in detail. If there are no comments, and the code is incorrect, you will get ZERO marks. If you make any special assumptions in your programs, or if you feel there are ideas that need explanation, describe them in a separate README file.
3. Proper use of the argument registers, return value registers, and the stack to save variables or to allocate memory when necessary is required.
4. Turn in a single zipped file containing your 3 .asm files and the README file in your assignment folder on *mycourses*.
5. Hints, suggestions and clarifications may be posted on the discussion board on *mycourses* as questions arise. Even if you don’t have any questions, it is a good idea to check the discussion board.