CSCI 201 Computer Organization I

MIPS Programming Project 2

Due November 17th, 4PM ET

Description:

Assume your Howard ID as a decimal integer is X. Let N = 26 + (X % 11) where % is the modulo operation, and M = N – 10.

You will write a MIPS program that reads a string of *up to* 1000 characters from user input.

- Remove leading and trailing blank spaces and tab characters. After that
  - If the string has zero characters or more than 4 characters or has at least one illegal character (a character outside the set described below), the program prints the message of "Unrecognized input".
  - If the string has only the characters from '0' to '9' and from 'a' to β and from 'A' to Δ, the program prints out the number of digits in the input, a comma, and then the unsigned decimal integer corresponding to the base-N number represented by the string. β stands for the M-th lower case letter and Δ stands for the M-th upper case letter in the English alphabet. In a base-N number, both 'a' and 'A' correspond to the decimal integer of 10, both 'b' and 'B' to 11, and so on, and both β and Δ correspond to N – 1.
- The program must exit after processing one single user input.
- The conversion from the input string to a decimal number must be done in a subprogram, where the memory address of the input string is passed into the subprogram via a register, and the decimal number is returned via a register. The number must be printed outside the subprogram.
- Do not print any prompt messages.

Sample test cases (assuming the Howard ID is 12345678):

12345678 % 11 = 4, therefore the base is 26 + 4 = 30, β is 't' and Δ is 'T'.

- Input: 0
  Output: 1,0
- Input: A
  Output: 1,10
- Input: 012
  Output: 3,32
- Input: 100
  Output: 3,900
- Input: 1023
  Output: 4,27063
- Input: A12t

Output: 4,270989

- Input: 12345
  Output: Unrecognized input
- Input: axyz
  Output: Unrecognized input
- Input: a b
  Output: Unrecognized input
- Input: a@t
  Output: Unrecognized input

Requirements:

- The program must be able to run correctly under QtSpim.
- No more commits after the submission deadline.
- The program must be named as project2.s.
- **The program must be in a new Github repo different than that used for Project 1.**
- The output must have the exact format as specified above.
- The program must use one or more loops to process the characters in the user input, instead of producing multiple segments of similar code with each segment processing one single character.
- All development must be done with Github. Specifically, there MUST BE ONE COMMIT for EVERY (or fewer for the last) five lines of new or modified MIPS code (excluding empty lines, and lines with only comment, directives, labels). More frequent commit is fine. With each commit, the commit message must explain the purpose of the added/changed code. **The commits must be done while the program is developed**. Commits done after the development completion or aggregated towards the end of development is NOT ACCEPTABLE.
- **Submissions not fully meeting the above requirements will lose significant portion (≥ 70%) of the credits.**
- Syntax error or program terminating abnormally without output will result in **zero credits**. Therefore, make sure to test your program completely before submission.

Submission:

- Add 'csdrli' (the instructor) as collaborators to your Github repository.
- Create a plain text file named readme.txt. Such a file can be created with notepad on Windows and nano on MacOS or Linux. The file should include ONLY the link to your Github repository that can be used to clone your repo.
- Submit the readme.txt file to http://hucs.dynu.net/lij/courses/submit_hw.html under "CSCI 201 MIPS Programming Project 2". Anything else must not be submitted.