System Software – Assignment 3

In this final assignment you will have to draw on much of what we have learned in the course to date, especially conditional jumps, loops, the VDU and procedures.

Write assembly code that performs each of these four tasks. Create a single assembly file (.asm) per task.

You can call each file "Assignment3-part-n" where n is the task number.

For this assignment, you must create a Design Summary Document. For each part of the assignment, you must explain in English how your code works. A paragraph or set of bullet points for each part of the assignment will suffice. Submit this as a single document. Number each section corresponding to each part of the assignment.

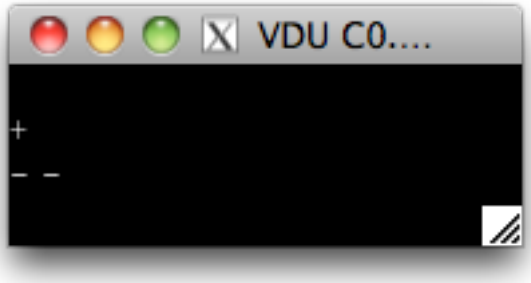
Read the instructions carefully before attempting each question.

1. Create a program that takes two single-digit numbers from simple keyboard input and adds them together. Print the answer to the VDU.

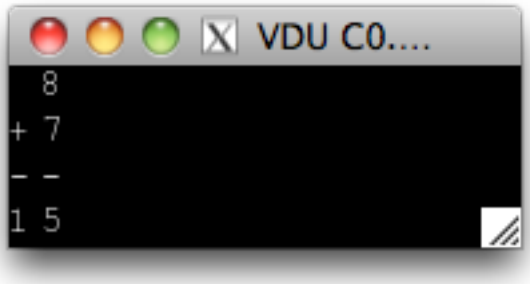
   Requirements:

   - **Your program must be capable of printing a two-digit answer.** For example, if the input is 8 and 9, the answer printed must be 17 (a 1 in one VDU memory slot, and a 7 in another beside it).
   - Your program must be capable of handling non-numeric character input. If an input character is not a number, re-prompt for input. Keep doing so until a valid character is entered (i.e. 0-9).

2. Write a similar program, this time show the sum on the VDU as in the screenshot below.



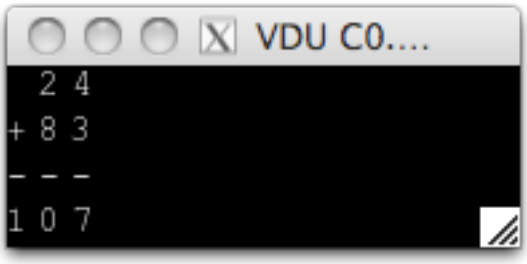
   Print the input numbers to the VDU as they are input, then print out the answer below the line.



Requirements:

- Numbers must be displayed as they are input.
- Once all numbers are input, only then can you add them together and display the result.
- Again, your program must check if the input characters are valid.
- Use the DB and ORG directives to print the "-" and "+" characters to the screen.

3. Create a similar program that adds two 2-digit numbers together.



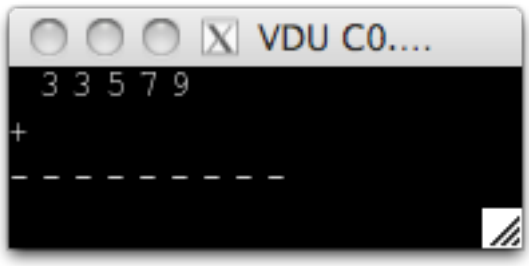
Requirements:

- The user must input the first number, right-to-left, then the second number to be printed below the first as in the screenshot. After each digit is entered, print it to the screen.

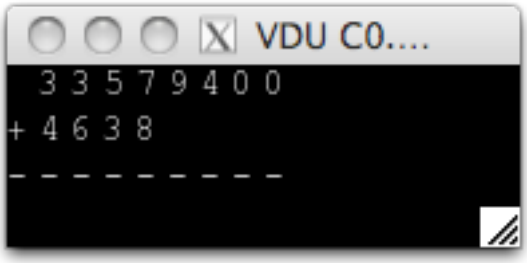
Hints:

- In order to write a solution for this, think about how you add numbers on paper ("carry the one").
- You are performing a repetitive task on each column of numbers. As such you should write a procedure that takes two numbers as input and prints the answer to the VDU.
- You should also write a procedure that gets input for each 2-digit number and prints it to the screen (digit by digit).
- For answers that are greater than 10 (0A in Hex), you may want to use a "carryOne" flag. (a memory slot that contains either a 1 or a 0).

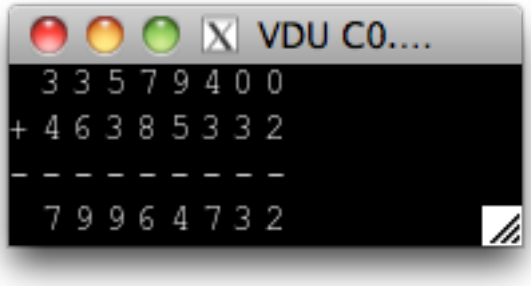
4. Extend your program from part 3 to create an adder for two 8-digit numbers.



User has input first 5 digits of the first number



User has input first 4 digits of second number



The program has completed

Requirements:

- Because of memory constraints, you now must create two procedures to carry out repetitive tasks.
  - One procedure takes input from the keyboard and prints an 8-digit number to the VDU. This will be called twice – one for each number.
  - One procedure that takes two numbers, adds them, and prints the answer to the VDU (dealing with the "carry one" issue).

Deliverables

- Comment each piece of code sensibly. Marks will be awarded for this.
- Zip your four asm files, as well as your design document (using winzip or winrar - http://www.rarlab.com/).
- Call the compressed file Assignment3_xxxxxxx – substituting your student number for the x's. **Make sure you name your file correctly otherwise I will not know who owns the submission!**
- Submit your .zip file to the submission page on the moodle.

Deadline – Tuesday 30th April at 9am.