Assignment 2 (3%)
ASCII to Integer (atoi)

Deadline: Friday 22 October 2021 at 23:59 on Submitty

Working individually, complete the assignment below. Submit your solution to Submitty (`https://submit.scss.tcd.ie`). By submitting your solution, you are confirming that you have familiarised yourself with College's policy on plagiarism (`https://libguides.tcd.ie/plagiarism`).

Your mark will be the auto-graded mark assigned by Submitty and will be marked out of 10.

You are allowed to submit five attempts for the assignment without penalty. Subsequent attempts will attract a 1 mark penalty each, up to a maximum penalty of 5 marks.

Submitty will allow you eight "late days" over the full semester. This means, for example, you can submit one assignment late by eight days or eight assignments late by one day each, without penalty. Once your "late days" are used up, you will receive zero marks for any late submissions.

Instructions

Part 1 (7 marks)

A sequence of four ASCII characters, each in the range '0' . . . '9', can represent an unsigned decimal value in the range 0 . . . 9999 in text form. Write an ARM assembly language program that will convert such a sequence to the value represented by the four digit characters. Assume that the four byte-size ASCII characters will be stored in `R1` to `R4`, with the most significant digit stored in `R4`. Store the result of the conversion in `R0`.

For example, given the following sequence of ASCII characters ...

'2', '0', '3', '4'

. . . your program should store the value 2034_{10} (`0x000007F2`) in `R0`.

Submit your solution to Submitty before moving on to Part 2.

Part 2 (3 marks)

In Part 1 above, you wrote a program that assumed that the value represented by the ASCII characters in `R1` to `R4` was a decimal value.

Extend your program to support the binary, octal (base 8) and hexadecimal numeral systems, in addition to decimal. The number base will be stored as a value in register `R5`.

Submitty will award one mark for each of the above numeral systems that your program supports, up to a total of 3 marks.

Example 1: Given the value 2 in R5 (indicating binary) and the following sequence of ASCII characters ...

‘1’, ‘0’, ‘1’, ‘1’

. . . your program should store the value 11_{10} (`0x0000000B`) in `R0`.

Example 2: Given the value 16 in R5 (indicating hexadecimal) and the following sequence of ASCII characters ...

‘0’, ‘1’, ‘A’, ‘4’

. . . your program should store the value 420_{10} (`0x000001A4`) in `R0`.