Our program will be written in C while having a function written in assembly ARM and it should be able to calculate the hash of a given string like this:

A) For every capital letter that exists in the given string, first it finds the value of that letter according to the given table, and then adds this value to the hash variable.

B) For every number that exists in the given string, it subtracts that number from the hash variable.

C) The value of the hash variable is not affected by string's characters that are not capital letters or simple numbers.

| A | 10 | J | 2 | S | 26 |
|---|----|---|----|---|----|
| B | 42 | K | 36 | T | 54 |
| C | 12 | L | 3 | U | 75 |
| D | 21 | M | 19 | V | 15 |
| E | 7 | N | 1 | W | 6 |
| F | 5 | O | 14 | X | 59 |
| G | 67 | P | 51 | Y | 13 |
| H | 48 | Q | 71 | Z | 25 |
| I | 69 | R | 8 | | |

The value of the hash variable can be stored at a memory address of your choice.

Specifically, you should have:

a) A basic main function written in C language, in which you will provide the necessary string.

b) A function written in assembly ARM in which our hash value is calculated, stored in memory and it is returned back to the main function, where it is printed on the screen.