The purpose of this lab is to introduce you to branches and loops in ARM processor. You will **NEED** to study the loop, branching, shift, and logic operations instructions in the slides of chapter 2 before attempting this program.

1. **Pre-lab:**

**This must be completed before coming to the lab and submitted with the final report.**

1. Prepare the first draft of the program described below. The program should contain directives and comments. **Comments are mandatory**

**PROGRAMMING ASSIGNMENTs:**

write a fully-commented program to execute the following tasks. Comments should mostly describe WHAT you are accomplishing toward solving the problem, not HOW each instruction works alone. Every instruction may not have a comment, but small groups of instructions must have explanatory comments.

**Task (1):**

(1) Define a word memory variable called Number1 and initialize it with the values given in Table 1.

(2) Write a code to count the number of ones in Number1 and store the result in R2.

**Hint:** Make a for loop for 32 iterations– initialize R2 with zero - in each iteration make a logic shift right (LSRS) and then if the carry flag is one increment R2

**Task (2):**

(1) Initialize R0 with the values given in Table 2.

(2) Write if-else statement to store 0x00000000 in R1 if bit 5 of R0 is zero and store 0xFFFFFFFF in R1 if bit 5 is one

**Task (3):**

(1) Define an array of words called Passwords and initialize it with the values given in Table 3

(2) Define a variable called input\_password and initialize it with 290657

(3) Write a program to store 0x00000000 in R4 if input\_password is not the list of passwords and store 0xFFFFFFFF in R4 if input\_password is in the list

Note 0x00000000 is a delimiter – your program should work without any modifications even if the size of the array changes – you should use while loop the ends when the delimiter is found

**Task (4):**

(1) Define an array of 10 words called Array and initialize it with -10, 14, -1, 0, 22, -7, -100, 0, -99, 77

(2) Write a program to count the number of negative elements and store the count in R4 and count the number of zeros and store the result in R5.

Hint: It is obvious that the numbers are signed

1. **In lab:**
2. Run the tasks using single step, run to cursor, or breakpoint procedures to debug the programs. Ensure that the tasks work as it is supposed to.
3. Record the results as follows:-

Table 1: Task (1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Number1 | 0xFFF00F00 | 0x00000000 | 0xFFFFFFFF | 0xF0000000 |
| R2 |  |  |  |  |

Table 2: Task (2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| R0 | 0x000000F0 | 0x00000000 | 0xFFFFFFFF | 0xF0000A00 |
| R1 |  |  |  |  |

Table 3: Task (3)

|  |  |  |
| --- | --- | --- |
| Passwords | 345676, 87654, 902345, 290657, 0x00000000 | 1234, 34567, 345676, 87654, 902345, 56784, 0x00000000 |
| R4 |  |  |

Table 4: Task (4)

|  |  |  |
| --- | --- | --- |
| Array | -10, 14, -1, 0, 22, -7, -100, 0, -99, 77 | 200, -14, 0, 0, 22, 0, -100, 0, -80, -180 |
| R4 |  |  |
| R5 |  |  |

Approved: Lab TA \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date \_\_\_\_\_\_\_\_\_\_\_\_

1. **Post lab:**

**Things to turn in as your Lab Report, attached in this order:**

1. This assignment sheet, with your name at the top, signed by the TA where shown.
2. **[6 marks]** The pre-lab program.
3. **[20 marks]** A printout of the program.
4. **[8 marks]** Tables 1 to 4
5. **[2 marks]** Why it is wrong to use LSR in task 1?
6. **[2 marks]** Does it matter if you use right or left shifts in task 1?