Autumn 2022, Homework 2 Key

Q1. (4 pts) Application Program Status Register (APSR)'s flags

After the following piece of instructions is executed, what value will be maintained in each of NZCV flags in APSR?

```
MOV     R0, #0x80000000
MOV     R1, #0x1
SUBS    R2, R1, R0
```

Flag	Value	Reason
N	1	The MSB is still 1, so negative
Z	0	The result is not 0
C	0	There is no carry over
V	1	The result is out of range

1 point for each row

Q2. (6 pts) Memory Endianness and Alignment

1) Allocate #76543210 to memory address 0x20000000. (2pts)

#76543210 = #0x048F F4EA

Big endian

Address	Data Contents (in hex)
0x20000000	04
0x20000001	8F
0x20000002	F4
0x20000003	EA

If the order is correct but numbers are wrong, -0.5
If the order is completely wrong, -1.

Little endian

Address	Data Contents (in hex)
0x20000000	EA
0x20000001	F4
0x20000002	8F
0x20000003	04

If the order is correct but numbers are wrong, -0.5
If the order is completely wrong, -1.

2) As you see the following example with exampleData, allocate myData to the memory and fill out the spaces to indicate how each data element is mapped. Assume that the memory is based on a 32-bit addressing system. (2pts)

An example:

```
struct exampleData {
  char a;
  short b;
};
```

	+ 0th	+ 1st	+ 2nd	+ 3rd
0th byte	a		b	b
4th byte				
8th byte				

A question you should solve:

```
struct myData {
  char a;
  int b;
  short d;
  double c;
  char *e;
  float f;
};
```

	+ 0th	+ 1st	+ 2nd	+ 3rd
0th byte	a	padding	padding	padding
4th byte	b	b	b	b
8th byte	d	d	padding	padding
12th byte	c	c	c	c
16th byte	c	c	c	c
20th byte	e	e	e	e
24th byte	f	f	f	f
28th byte				
32nd byte				

If any value is placed at the empty box, -1.
If any value is wrong, -1.

Q3. (10pts) Introduction to VisUAL

Complete the following assembly program that 1) read characters (i.e., ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ‘I’, ‘J’, ‘K’, ‘L’) given at address “src”, 2) change them to the corresponding lower-cased characters, and 3) store them at address “dst”.

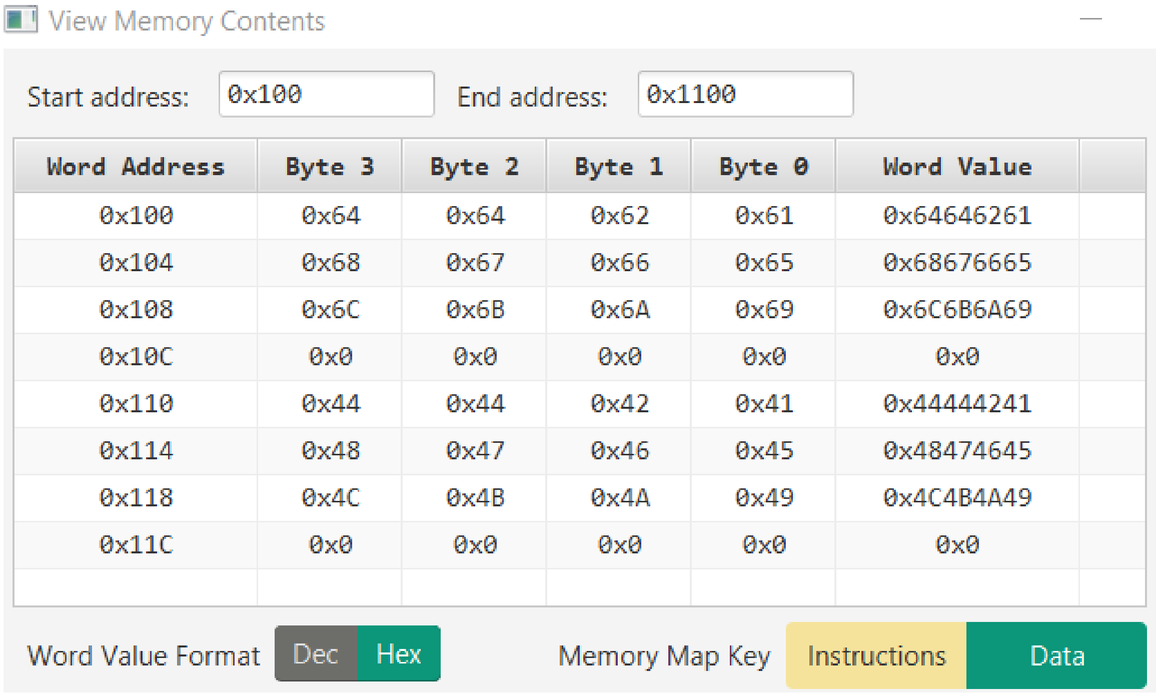
Submission: You need to submit **two files**.
File 1 (5pts): Your HW2-Q3.s.
File 2 (5pts): A **pdf** file with:

1. (1pts) A screenshot of your code.
2. (4pts) A screenshot of VisUAL’s “View Memory Contents” window.

```
src             DCB             'a', 'b', 'd', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 0
dst             DCB             0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  0

begin
                LDR             R0, =src
                LDR             R1, =dst
loop
                LDRB            R3, [R0], #1
                CMP             R3, #0
                BEQ             stop
                SUB             R3, R3, #32
                STRB            R3, [R1], #1
                B               loop
stop            END
```

The code here is just for reference.
If the code is not uploaded as an individual file, -5.
If the code is not copied to this submission, -2.
If the memory contents view is not included, -5.
If the contents look very different, -3.



Start address: 0x100

End address: 0x1100

Word Address	Byte 3	Byte 2	Byte 1	Byte 0	Word Value
0x100	0x64	0x64	0x62	0x61	0x64646261
0x104	0x68	0x67	0x66	0x65	0x68676665
0x108	0x6C	0x6B	0x6A	0x69	0x6C6B6A69
0x10C	0x0	0x0	0x0	0x0	0x0
0x110	0x44	0x44	0x42	0x41	0x44444241
0x114	0x48	0x47	0x46	0x45	0x48474645
0x118	0x4C	0x4B	0x4A	0x49	0x4C4B4A49
0x11C	0x0	0x0	0x0	0x0	0x0

Word Value Format Dec [✓ Hex]

Memory Map Key