name:	Vince
course:	csci 10
assignment:	homework 8
prepared:	Mon, Mar 30, 2020 // 9:07 pm

1. What is the carry flag and how is it used on a CPU?

The carry flag is affected by the result of most arithmetic and some bit wise instructions and is also used as an input for many. The use of the carry flag in this manner enables multi-word add, subtract, shift, and rotate operations.

2. Thoroughly explain, from the reading, the example of adding 255 to 255 using 8-bit registers, and how this operation would involve the carry flag.

When adding 255 and 255 in 8-bit registers the result is 510, requiring the 9 bits 111111110. The 8 least significant bits are stored in the register as 11111110 in binary, 254 in decimal, but since there is carry out of bit 7 (the eight bit), the carry is set, indicating that the result needs 9 bits. Concatenating the carry flag and result produces the valid 9 bit value.

3. Briefly describe the N(egative), Z(ero), C(arry), and (o)V(erflow) bits of the ARM Current Program Status Register (CSPR).

Negative: is set to bit 31 of the result, so N is 1 if the signed value is negative, and cleared if the result is positive or zero.
Zero: is set if the result is zero; this is usual to denote an equal result from a comparison. If the result is non-zero, this flag is cleared.
Carry:
With the instructions ADC, ADD, and CMN, the Carry flag is set if the result would produce an unsigned overflow.
With the instructions CMP, SBC, and SUB, the Carry flag is set if the result would produce an unsigned underflow (a borrow).
For other instructions that use shifting, the Carry flag is set to the value of the last bit shifted out by the shifter.
oVerflow: for addition and subtraction, this flag is set if a signed overflow occurred. Otherwise, it is generally left alone. Note that some API conventions may specifically set oVerflow to flag an error condition.

4. Write the instruction that translates the expression r0 = r1 + r2 .

```
ADD r0, r1, r2
```

5. Write the instruction that translates the expression r0 = r1 + 1 . (1 is an immediate value)

```
ADD r0, r1, #1
```

6. Write the instruction that translates the expression r0 = r1 + r2 , setting the status bits in the CSPR.

```
ADDS r0, r1, r2
```

7. Write the instruction that translates the expression r0 = r1 - r2 .
SUB r0, r1, r2

8. Write the instruction that translates the expression r0 = r1 - 1 . (1 is an immediate value)
SUB r0, r1, #1

9. Write the instruction that translates the expression r0 = r1 - r2 , setting the status bits in the CSPR.
SUBS r0, r1, r2

10. Briefly describe RSB.
RSB is reverse substract, the difference from SUB is that operand 2 is now the value being subtracted from. RSB r0, r1, r2 stores r2 - r1 in r0.