From last time

Explain the similarities & differences between the ARM instructions `SUB R0, R1, R2` & `RSB R0, R1, R2` (2 marks)
Both take the contents of R1 & R2, subtract one from the other, and put the result in R0. SUB does R1–R2, RSB does R2–R1

Explain the similarities & differences between the ARM instructions `CMP R0, R1` & `CMN R0, R1` (2 marks)
Both use the contents of R0 & R1 to set the condition codes. CMP compares 0 & R0–R1, CMN compares 0 & R0+R1.

Explain why it can be useful to have an “RSB” operation as well as an “SUB” operation in the ARM instruction set. Give an example to illustrate your answer. (2 marks)
“SUB R0,R1,R2” is the same as “RSB R0,R2,R1” but only the last operand can be replaced by a literal e.g. “R4=R3–1” is “SUB R4,R3,#1” but “R4=1–R3” is “RSB R4,R3,#1” not “SUB R4,#1,R3”
Question

Give ARM code for:

```c
if (a == b * c)  
a = b;
else
    b = b - 1;
```

Assume a, b, c are int variables.
Answer

LDR  R0,  a
LDR  R1,  b
LDR  R2,  c
MUL  R1, R1, R2
CMP  R0, R1
BNE  skip
LDR  R0,  b
STR  R0,  a
B    both
skip LDR  R0,  b
SUB  R0, R0, #1
STR  R0,  b
both  .  .  .

or:

LDR  R0,  a
LDR  R1,  b
LDR  R2,  c
MUL  R2, R1, R2
CMP  R0, R2
BNE  skip
; (don't reload b)
STR  R1,  a
B    both
skip ; (don't reload b)
SUB  R1, R1, #1
STR  R1,  b
both  .  .  .

Question

Give ARM code for:

```c
if (people > 300)
    watts = 500;
else if (people > 100)
    watts = 150;
else
    watts = normal;
```

Assume watts, people, normal are int variables.

Note: 300, 500, 100, 150 will all fit into ARM short literals.
Answer

LDR R0, people
CMP R0, #300
BLE skip1
MOV R0, #500
STR R0, watts
B all

skip1 CMP R0, #100
BLE skip2
MOV R0, #150
STR R0, watts
B all

skip2 LDR R0, normal
STR R0, watts
all . . .

LDR R0, people
CMP R0, #300
BLE skip1
MOV R0, #500
B all

skip1 CMP R0, #100
BLE skip2
MOV R0, #150
B all

skip2 LDR R0, normal
all STR R0, watts
Question

Give ARM code for:

```c
while (side * side * side < vol)  
    side = side + 1;
```

Assume side, vol are int variables.

```
LDR R0, side
LDR R1, vol
start MUL R2, R0, R0
    MUL R2, R2, R0
    CMP R2, R1
    BGE skip
    ADD R0, R0, #1
    B start
skip STR R0, side
```
Question

Give the **best** ARM code that you can, using conditional arithmetic etc, for:

```c
while (side * side * side < vol)
    side = side + 1;
```

Assume side, vol are int variables.

```
LDR   R0, side
LDR   R1, vol
start MUL   R2, R0, R0
         MUL   R2, R2, R0
         CMP   R2, R1
         ADDLT R0, R0, #1
         BLT   start
STR    R0, side
```
Question

Give the **best** ARM code that you can, using conditional arithmetic etc., for: (x, y & z are positive int variables)

```plaintext
x = 0;
while (y - z >= 0) {
    y = y - z;
    x = x + 1;
}

MOV    R0, #0; R0 = x
LDR    R1, y
LDR    R2, z
start SUBS   R3, R1, R2
MOVGE R1, R3
ADDEGE R0, R0, #1
BGE   start
STR    R0, x
STR    R1, y
```