From last time

Explain how a table of code addresses ("jump table") can be used in the implementation of a switch statement. (4 marks)

For each "case n", the address of the start of its code is the n-th entry in the table.
(address of start of "default" code in any undefined entries)

Code for "switch(i)" loads i-th address from the table into PC, so the next instruction obeyed is the start of the code for the i-th case.
(First, bound-check i to make sure there is an i’th element in the table.)
Question

\[ b = (x == 0); \]

LDR    R0,  x
CMP    R0,  #0
MOVEQ  R1,  #1 ; true
MOVNE  R1,  #0 ; false
STRB   R1,  b
Question

\[ b4 = (b1 \& b2) \mid \neg b3; \]

```
LDRB R0, b1
LDRB R1, b2
AND R0, R0, R1
LDRB R1, b3
EOR R1, R1, #1
ORR R0, R0, R1
STRB R0, b4
```
Question

Optimise:

MOVGT  R4,  #1
MOVLE  R4,  #0
ORR   R3, R3, R4

→

ORRGT  R3, R3, #1
ORRLE  R3, R3, #0

→

MOVGT  R3,  #1
Question

\[ b = (x > 7) \mid (x < 4); \]

LDR R0, x
CMP R0, #7 ; if \((x>7)\) \(b\) is true
MOVGT R3, #1 ; else false so far
MOVLE R3, #0
CMP R0, #4
MOVLT R3, #1 ; if \((x<4)\) \(b\) is true
STRB R3, b
Question

\[ b = (x == 0) || oops(x); \]

boolean oops(int x) ...

1st parameter & result in R0

LDR R0, x
CMP R0, #0
MOVEQ R0, #1 ; if x==0 then R0 is true
BLNE oops ; if x!=0 then R0 is method result
STRB R0, b