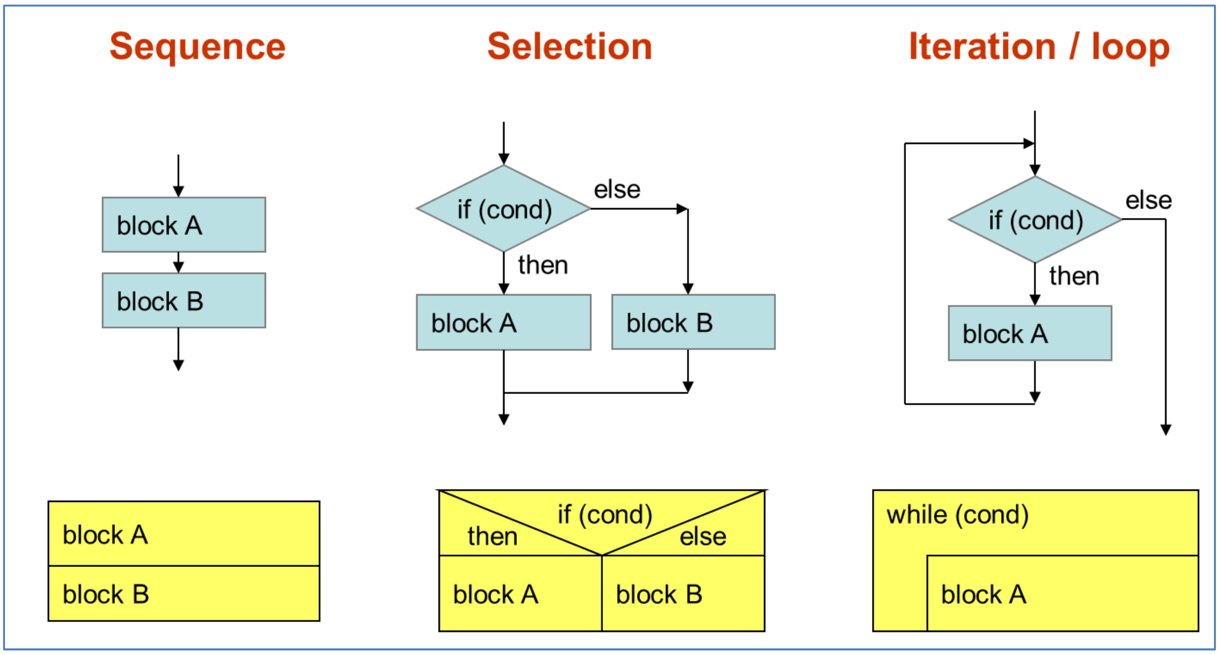
CT1 Exercises for Control Structures

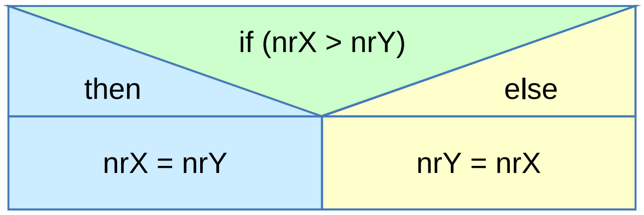
Content



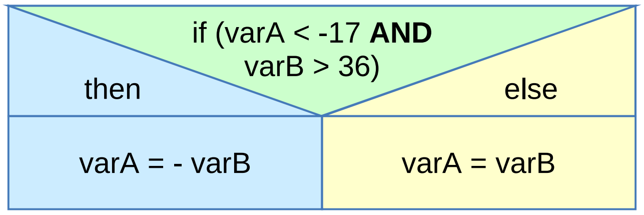
Exercise 1 – Selection/Branch

Encode the following Structograms into Flowchart, C- and ARM Assembly-language

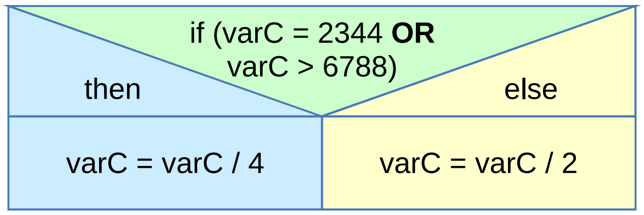
A) If-Then-Else with **unsigned** 8-bit variables



B) If-Then-Else with **signed** 8-bit variables



C) If-Then-Else with **signed** 16-bit variables



Exercise 2 – For-Loops

A) Write a for-loop in C- and ARM Assembly-language.

B) Compare your Assembly-language implementation with the compiler generated one.

Hint: In the Keil uVision5 IDE

1) create an empty C-language project (according to the respective introduction documents)
2) add the C-language for-loop to the empty main function
3) compile the project
4) set a breakpoint in at the first line of the main function
5) start debugging the program and let it run into the breakpoint
6) compare your Assembly-language implementation of the for-loop with the compiler generated one

Hint: for the purpose of this exercise, define your variables global and as “volatile” – this tells the compiler to not optimize away the access to the variables since they are not used otherwise.

Exercise 3 – From Code to Structogram

A) Analyze the following Assembly-language code and derive from this the matching structogram.

B) What result is stored in “outstr”?

```
        AREA progCode, CODE, READONLY
        THUMB

main    PROC
        EXPORT main

        LDR    R0,=srcstr
        LDR    R1,=outstr
        MOVS   R2,#0
cond    LDRB   R3,[R0,R2]
        CMP    R3,#0
        BEQ    endloop
        CMP    R3,#60
        BLO    store
        CMP    R3,#90
        BHI    store
        ADDS   R3,R3,#32
store   STRB   R3,[R1,R2]
        ADDS   R2,R2,#1
        B      cond
endloop STRB   R3,[R1,R2]

endless B      endless
        ENDP
srcstr  DCB    "This IS mY TestStriNG", 0

        AREA progData, DATA, READWRITE
outstr  SPACE 50

        END
```