CSE 331L: Microprocessor Interfacing and Embedded Systems Lab

Summer 2025

Arm Assembly (Part-3)

Class # 04

Recap

- Logical Instruction
- CMP Instruction
- Shift Instruction
- Rotate Instruction
- Arrays

Branch

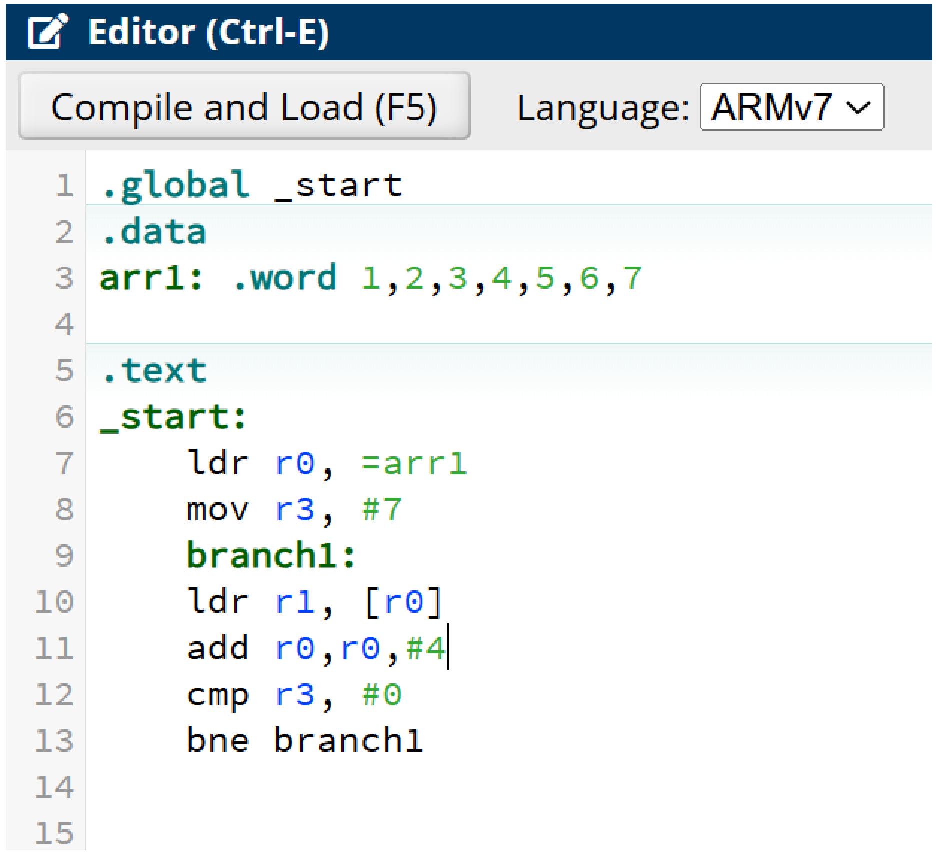
- What is branch?
- Why branch is used for?

Branch

- What is branch?
    - In ARM Assembly, a branch is an instruction used to change the flow of execution in a program — similar to a jump or goto in high-level languages.
- Why branch is used for?
    - A branch is used in ARM assembly to change the normal sequential flow of execution in a program. It allows the program to jump to another instruction, enabling important programming constructs

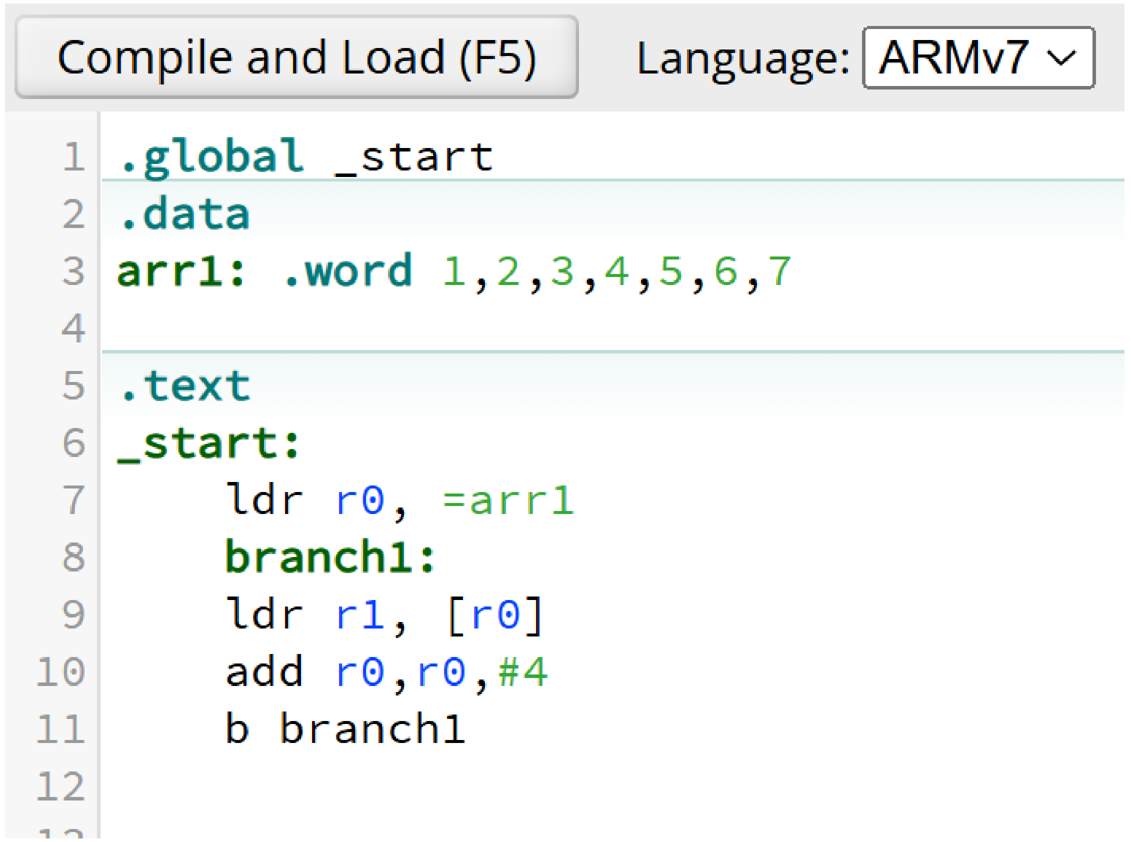
Branch types

- Conditional Branch



```
.global _start
.data
arr1: .word 1,2,3,4,5,6,7

.text
_start:
    ldr r0, =arr1
    mov r3, #7
    branch1:
    ldr r1, [r0]
    add r0,r0,#4
    cmp r3, #0
    bne branch1
```

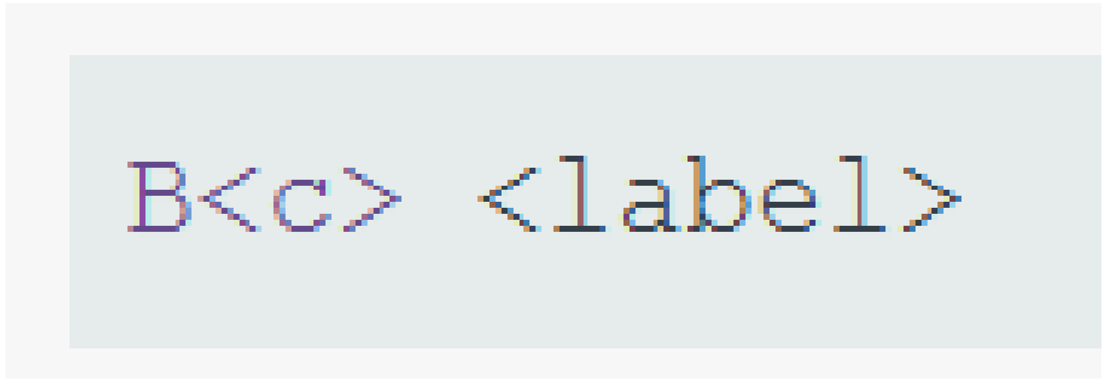
- Unconditional Branch



```
.global _start
.data
arr1: .word 1,2,3,4,5,6,7

.text
_start:
    ldr r0, =arr1
    branch1:
    ldr r1, [r0]
    add r0,r0,#4
    b branch1
```

B (Unconditional Branch)

- Branch causes a branch to a target address



```
B<c> <label>
```

B_ (conditional Branch)

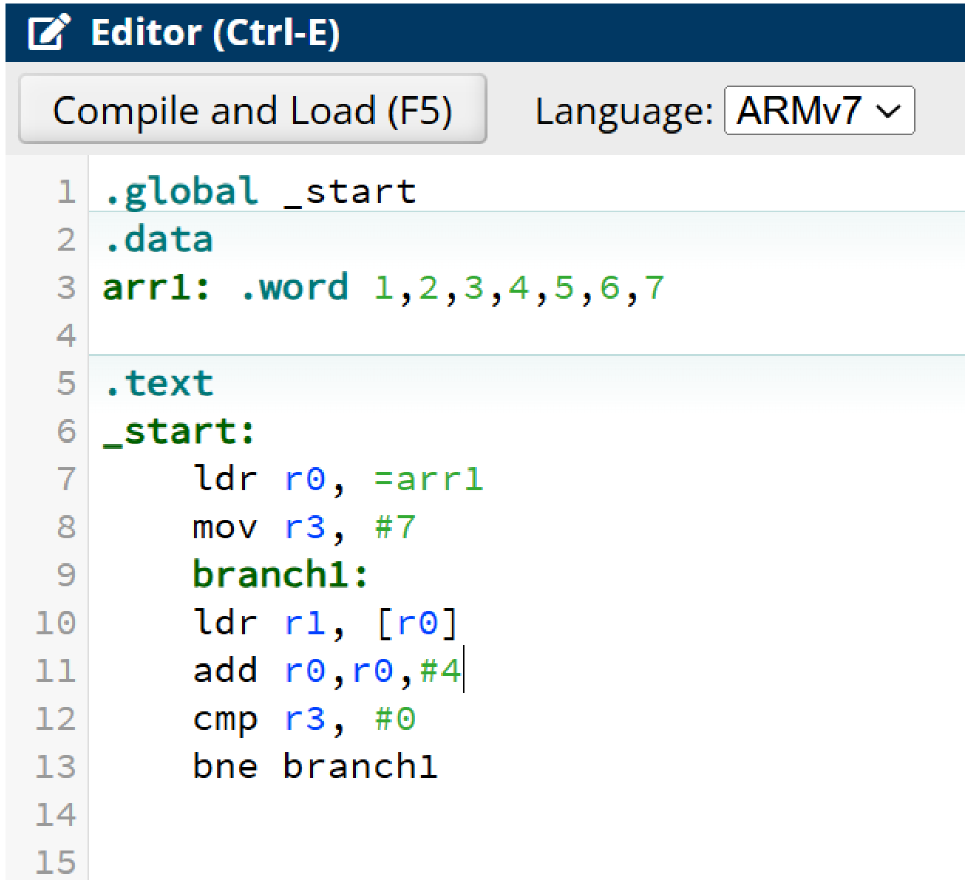
- When the condition is matched, will be executing the branch
- Several types of conditional branches:
    - **BGT**
    - **BLT**
    - **BGE**
    - **BLE**
    - **BEQ**
    - **BNE**
    - **BMI**

Loop

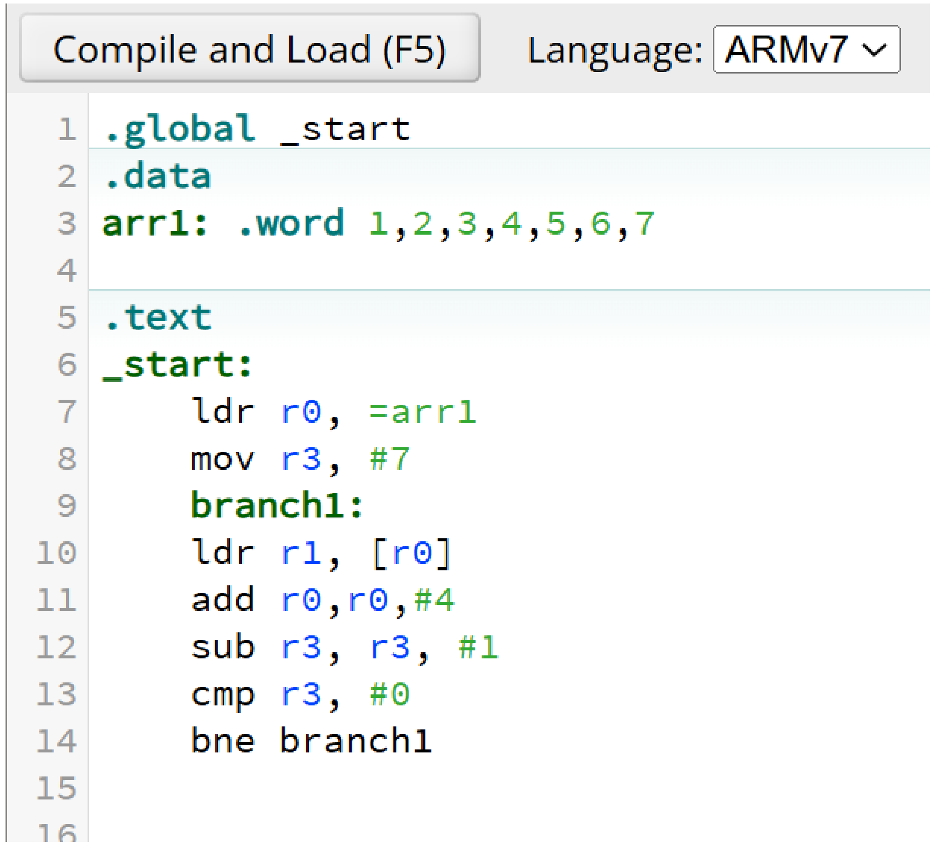
- The branch is actually doing the task of the loop.
- But one thing missing, can you guess?

Counter

- How to implement a counter inside a branch?



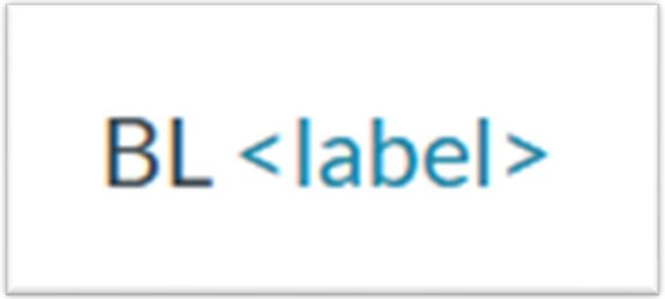
```
.global _start
.data
arr1: .word 1,2,3,4,5,6,7

.text
_start:
    ldr r0, =arr1
    mov r3, #7
    branch1:
    ldr r1, [r0]
    add r0,r0,#4
    cmp r3, #0
    bne branch1
```



```
.global _start
.data
arr1: .word 1,2,3,4,5,6,7

.text
_start:
    ldr r0, =arr1
    mov r3, #7
    branch1:
    ldr r1, [r0]
    add r0,r0,#4
    sub r3, r3, #1
    cmp r3, #0
    bne branch1
```

BL

- Branch with Link branches to a PC-relative offset, setting the register X30 to PC+4. It provides a hint that this is a subroutine call.

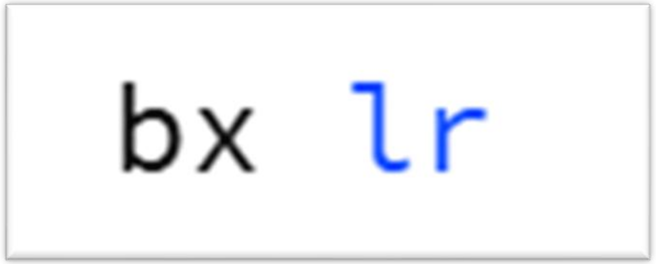


```
BL <label>
```

Referenc
e

BX

- Branch and Exchange causes a branch to an address and instruction set specified by a register.



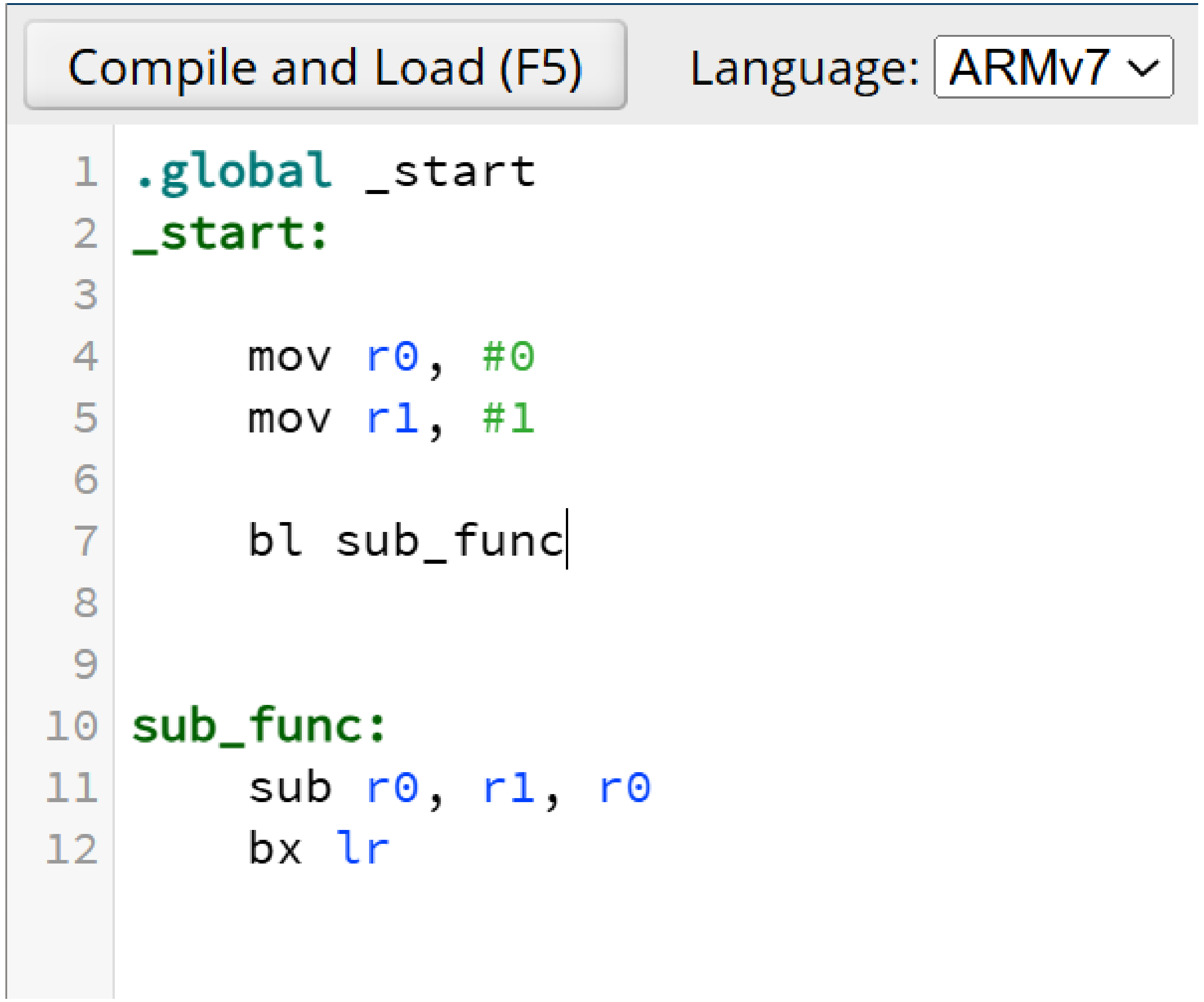
```
bx lr
```

Referenc
e

Why BL and BX

- For making a function call in assembly language
  - Function call uses r0 to r4 GP registers

Code

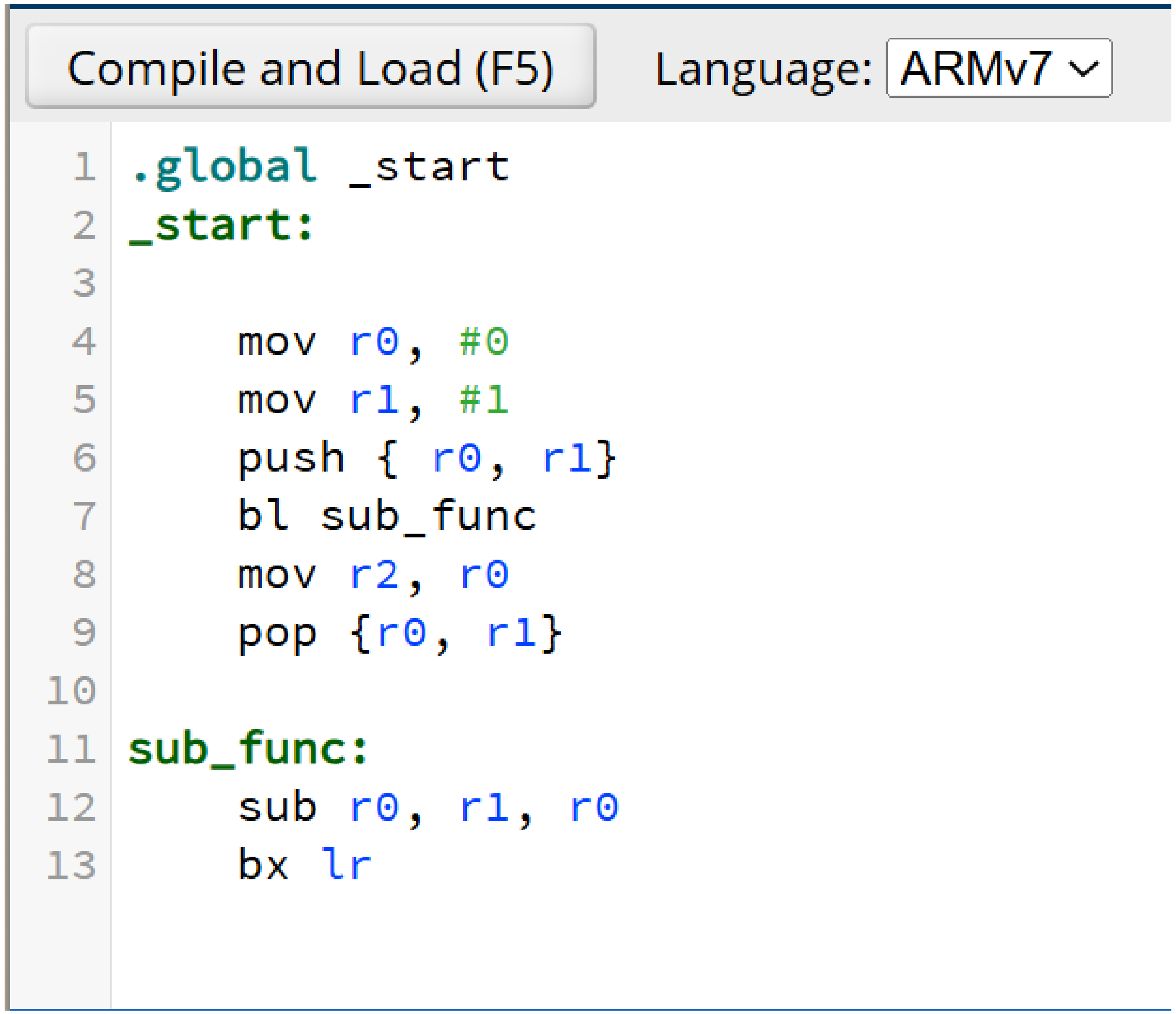


```
.global _start
_start:

    mov r0, #0
    mov r1, #1

    bl sub_func


sub_func:
    sub r0, r1, r0
    bx lr
```

Overwriting values?

- While using function in Assembly, the values of the GP registers can be changed.
- To solve this issue, we need to use **stack**

Solution →



```
.global _start
_start:

    mov r0, #0
    mov r1, #1
    push { r0, r1}
    bl sub_func
    mov r2, r0
    pop {r0, r1}

sub_func:
    sub r0, r1, r0
    bx lr
```