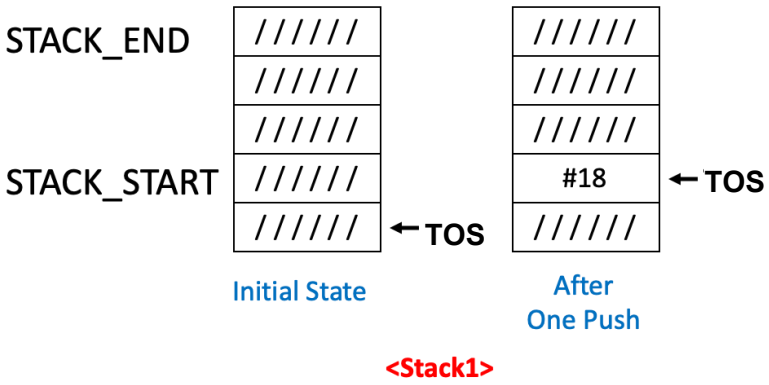
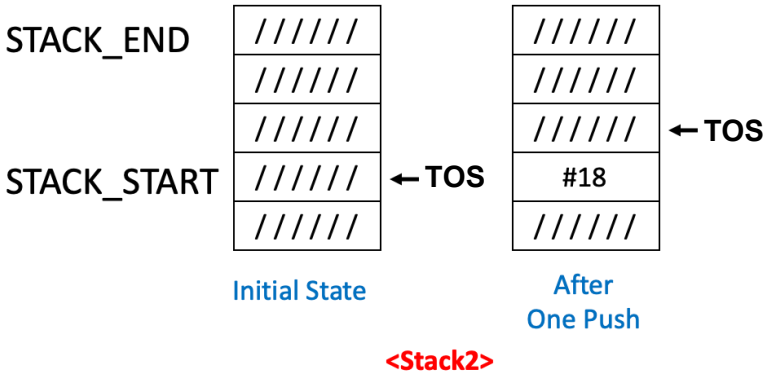
Worksheet 3 - Stack

Exercise 1:
In this course, we will introduce two different styles of stacks by how TOS pointer is defined. Their functions are the same, so the caller-function does not see the difference. However, the actual implementation would look different.

- Stack1: TOS pointer holds the memory location of the top item.
- Stack2: TOS pointer holds the memory location of the next available spot.





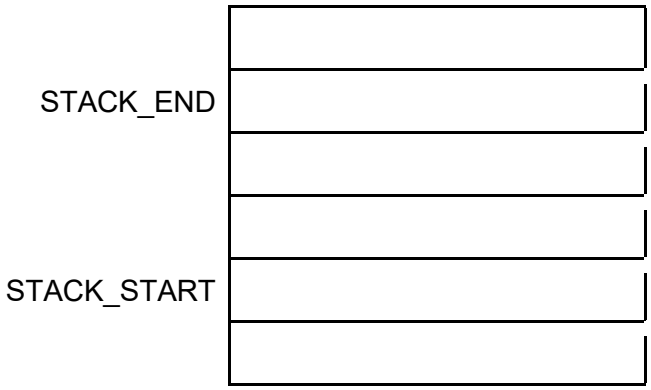
Implement the basic functionality of PUSH and POP for stack1 and stack2. The detail codes are omitted.

```
;stack1
PUSH
ADD R6, R6, #-1 ; decrement TOS pointer
STR R0, R6, #0 ; store data to TOS

POP
LDR R0, R6, #0 ; load data
ADD R6, R6, #1 ; increment TOS pointer
```

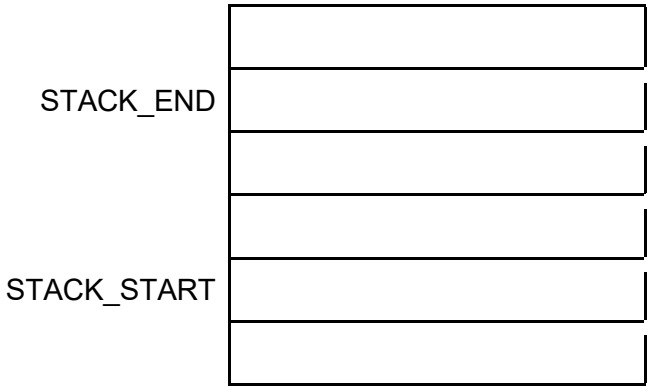
```
;stack2
PUSH
STR R0, R6, #0 ; store data to TOS
ADD R6, R6, #-1 ; decrement TOS pointer

POP
ADD R6, R6, #1 ; increment TOS pointer
LDR R0, R6, #0 ; load data
```

Locate the TOS pointer for the following conditions.

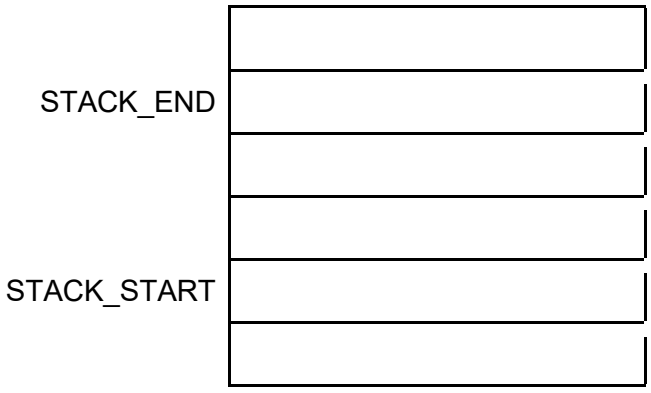
1. Stack1 is full



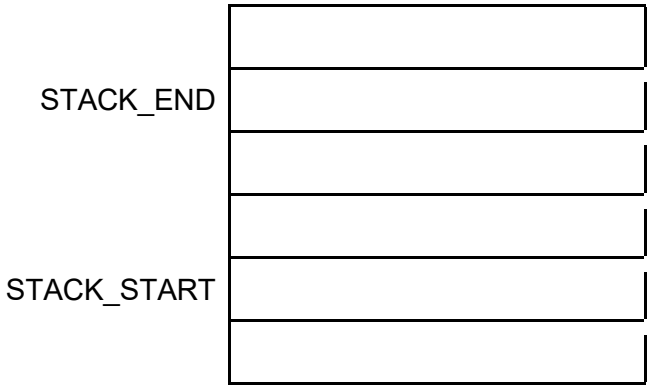
2. Stack1 is empty



3. Stack2 is full



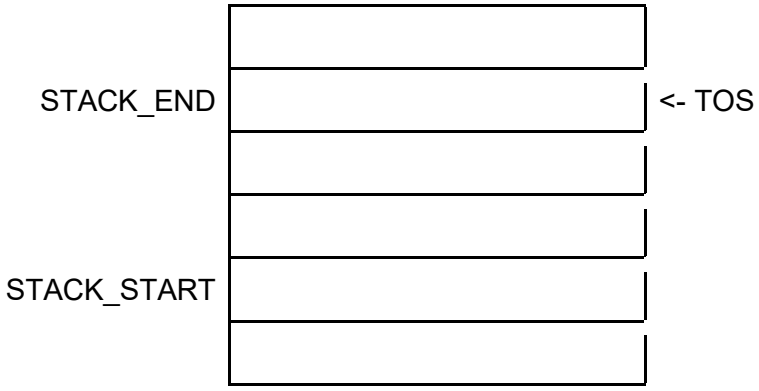
4. Stack2 is empty



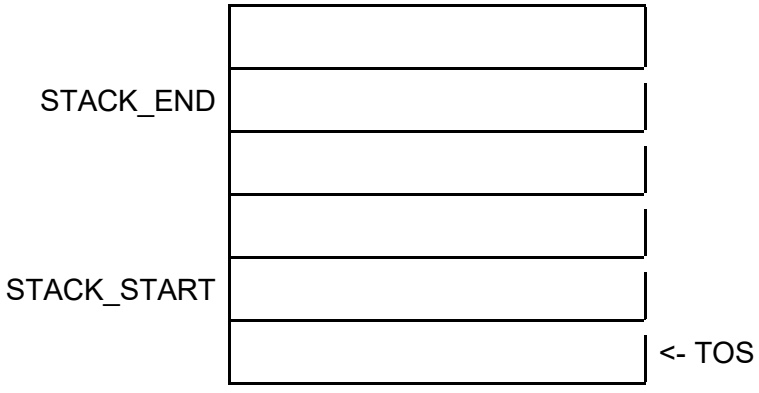
SOL:

Locate the TOS pointer for the following conditions.

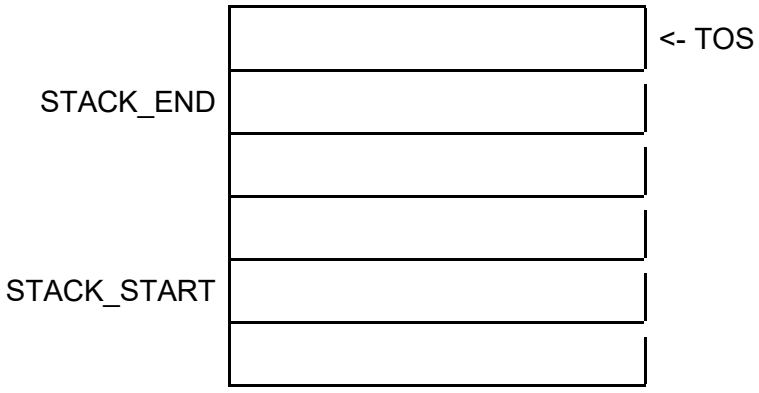
1. Stack1 is full



2. Stack1 is empty



3. Stack2 is full



4. Stack2 is empty