

What Do the execute() Methods of These Instructions Do?

ADD

JUMP

JUMPONFALSE

LE

LOADFROMADDR

PUSHLOCADDR

PUSHNUM

PUSHSTATADDR

SAVETOADDR

WRITESTRING

Execution of ADD by: `--ESP;`
`EXPRSTACK[ESP-1] += EXPRSTACK[ESP];`

s t k o f f r f a s m e t	CURRENTLY EXECUTING METHOD ACTIVATION'S STACKFRAME (Part of Data memory)
⋮	_____
-1	_____
0	_____
+1	_____
+2	_____
+3	_____
⋮	_____



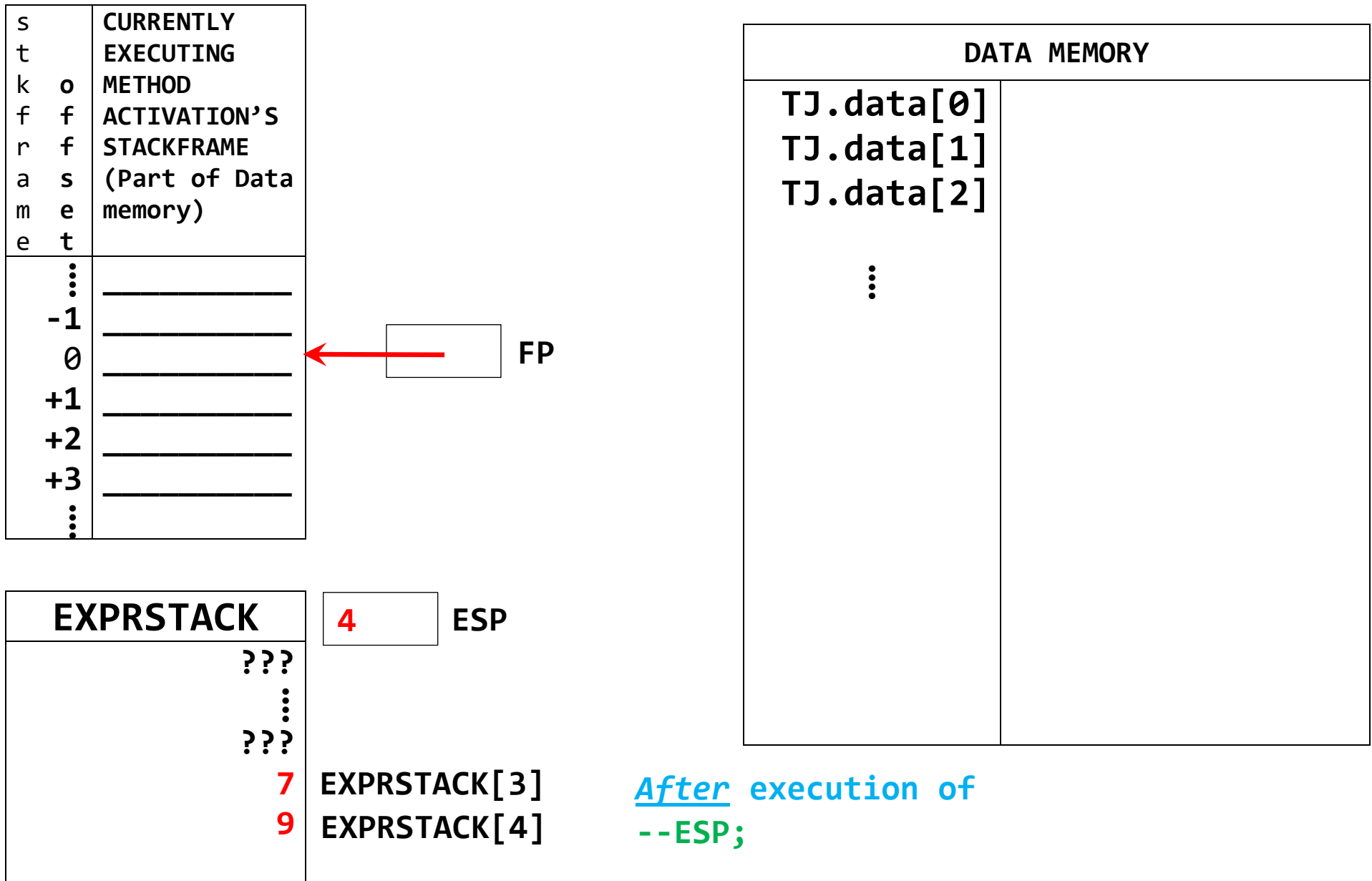
DATA MEMORY	
TJ.data[0]	
TJ.data[1]	
TJ.data[2]	
⋮	

EXPRSTACK	
???	
⋮	
???	
7	EXPRSTACK[3]
9	EXPRSTACK[4]

5 ESP

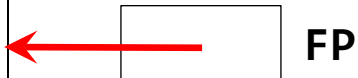
Before execution of
`--ESP;`

Execution of ADD by: `--ESP;`
`EXPRSTACK[ESP-1] += EXPRSTACK[ESP];`



Execution of ADD by: $--ESP;$
 $EXPRSTACK[ESP-1] += EXPRSTACK[ESP];$

s t k o f f r f a s m e t	CURRENTLY EXECUTING METHOD ACTIVATION'S STACKFRAME (Part of Data memory)
⋮	_____
-1	_____
0	_____
+1	_____
+2	_____
+3	_____
⋮	_____



DATA MEMORY	
TJ.data[0]	
TJ.data[1]	
TJ.data[2]	
⋮	

EXPRSTACK	
???	
⋮	
???	
16	

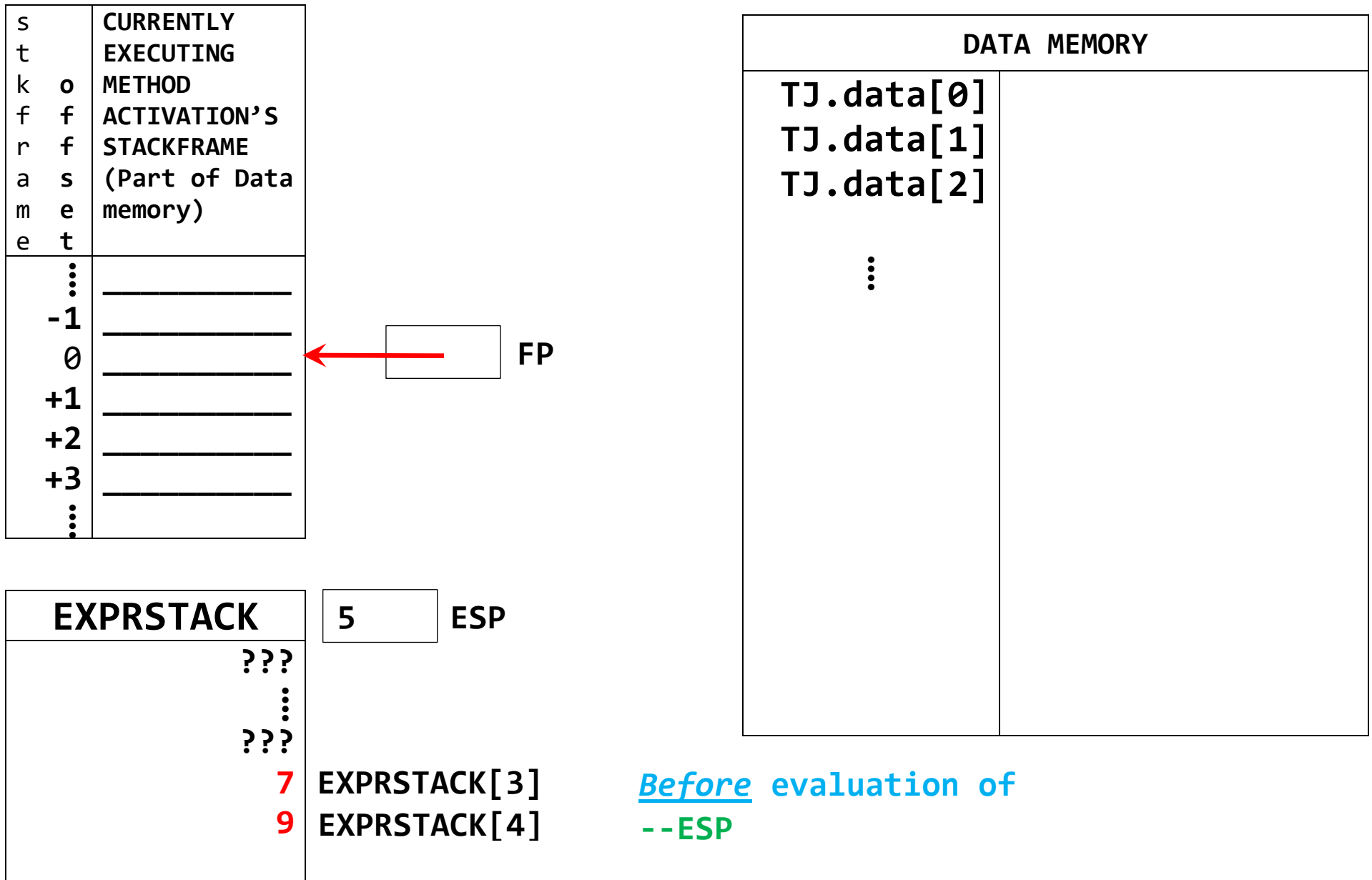
4 ESP

EXPRSTACK[3]

After execution of
 $EXPRSTACK[ESP-1] += EXPRSTACK[ESP];$

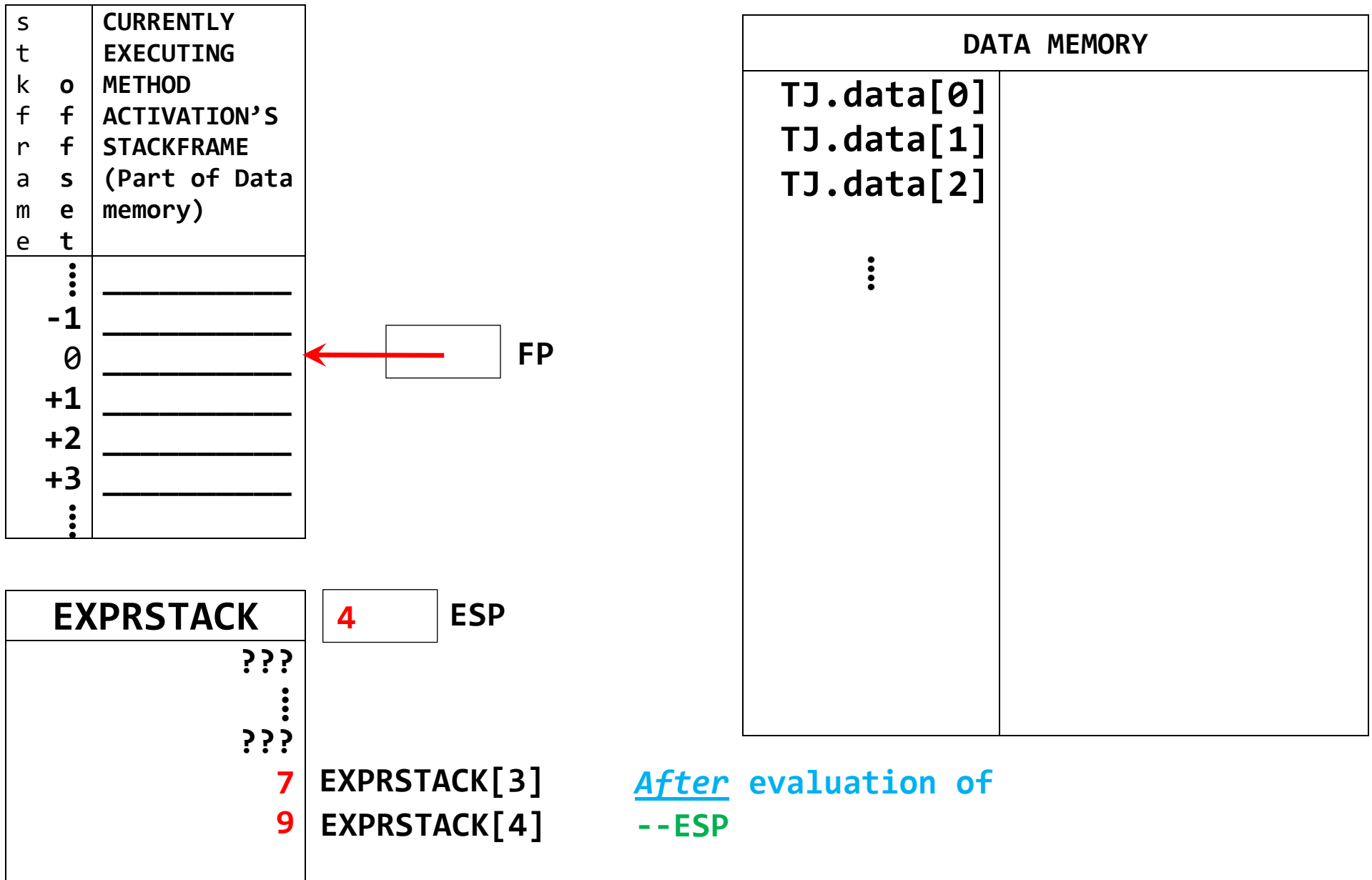
Execution of LE by:

$\text{EXPRSTACK}[\text{--ESP}-1] = (\text{EXPRSTACK}[\text{ESP}-1] \leq \text{EXPRSTACK}[\text{ESP}]) ? 1 : 0;$



Execution of LE by:

$\text{EXPRSTACK}[\text{--ESP}-1] = (\text{EXPRSTACK}[\text{ESP}-1] \leq \text{EXPRSTACK}[\text{ESP}]) ? 1 : 0;$



Execution of LE by:

$\text{EXPRSTACK}[\text{--ESP}-1] = (\text{EXPRSTACK}[\text{ESP}-1] \leq \text{EXPRSTACK}[\text{ESP}]) ? 1 : 0;$

s t k o f f r a m e s	CURRENTLY EXECUTING METHOD ACTIVATION'S STACKFRAME (Part of Data memory)
⋮	_____
-1	_____
0	_____
+1	_____
+2	_____
+3	_____
⋮	_____



DATA MEMORY	
TJ.data[0]	
TJ.data[1]	
TJ.data[2]	
⋮	

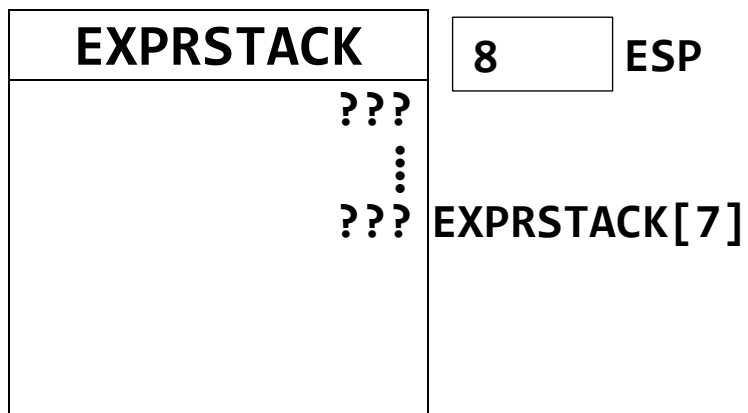
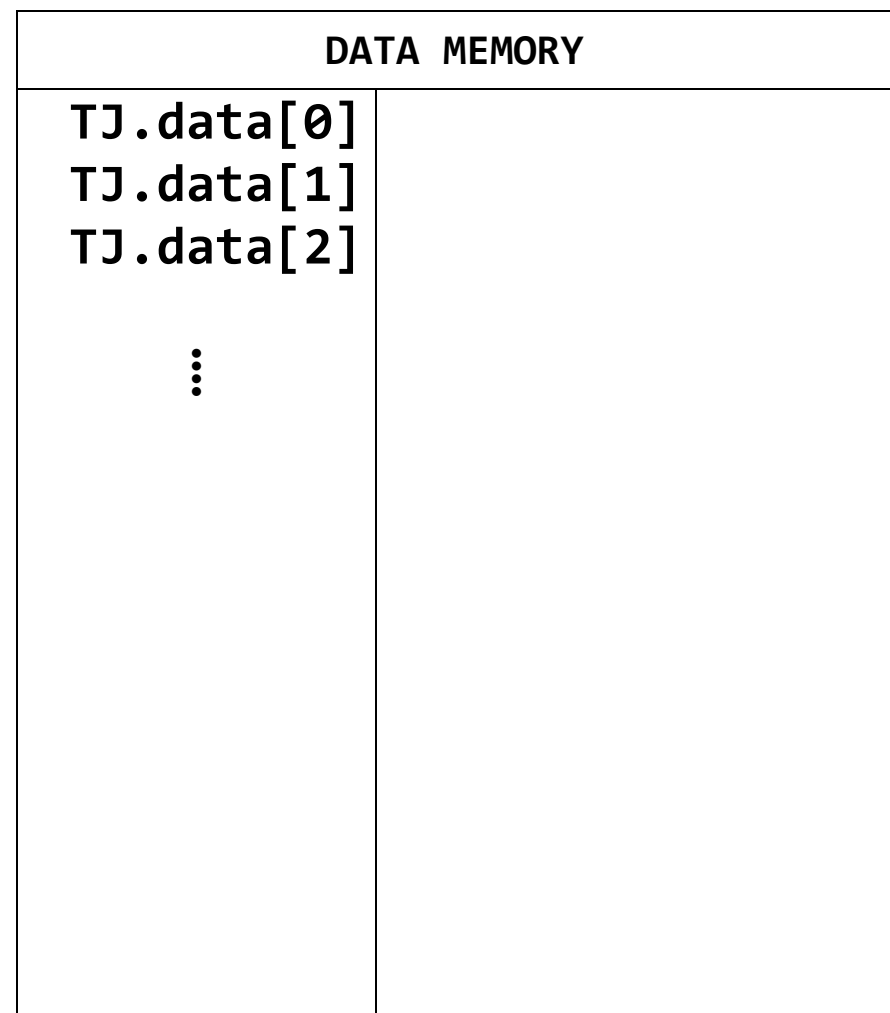
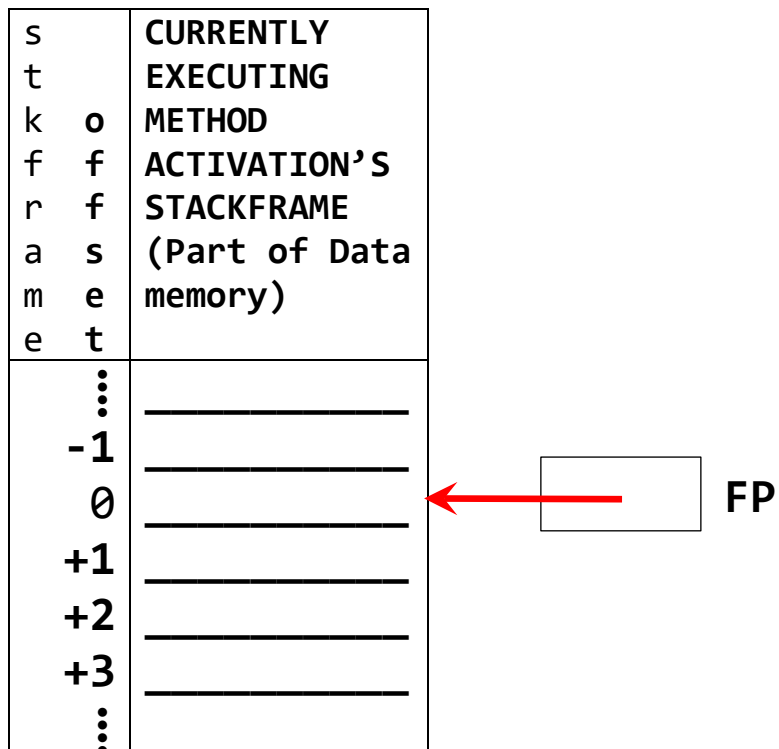
EXPRSTACK	
???	
⋮	
???	
1	



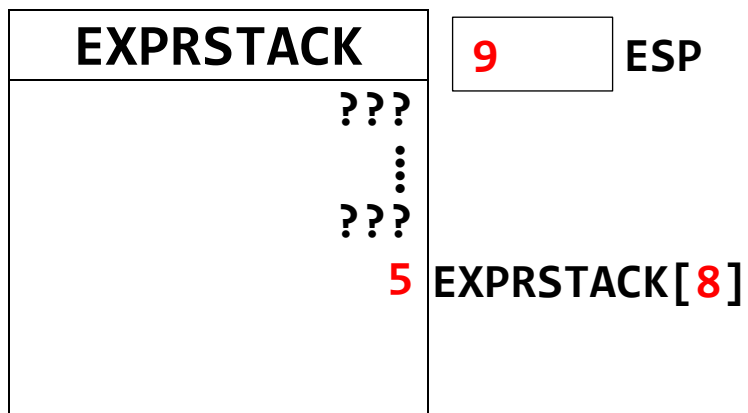
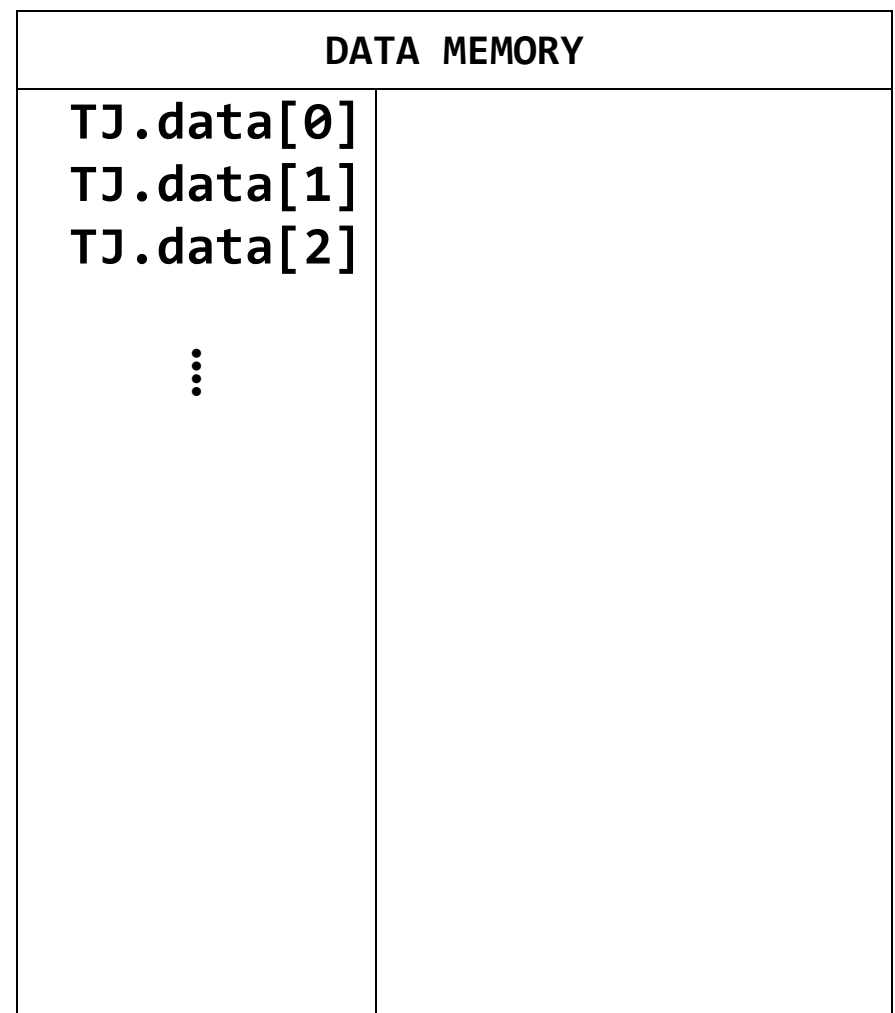
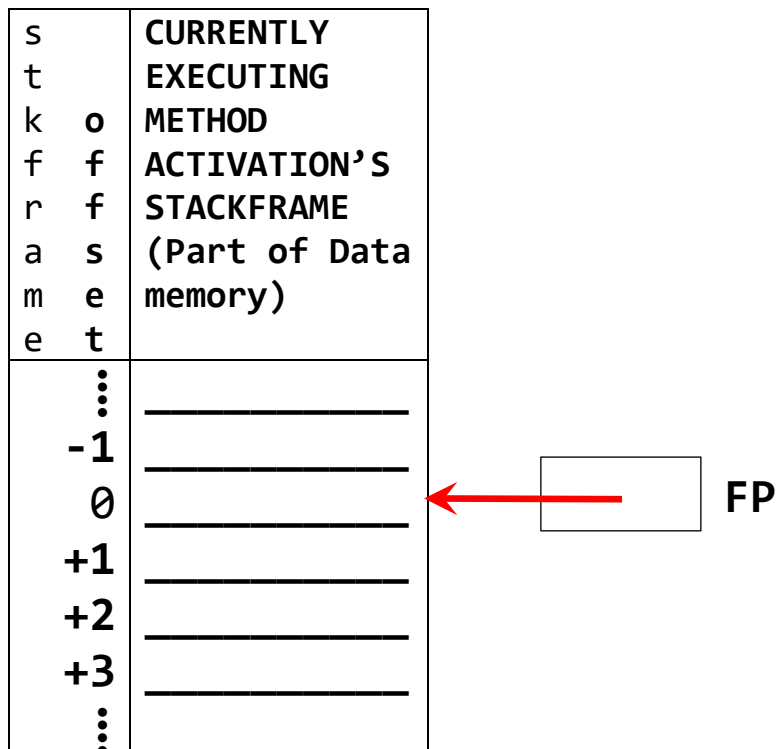
EXPRSTACK[3]

After execution of
 $\text{EXPRSTACK}[\text{--ESP}-1] =$
 $(\text{EXPRSTACK}[\text{ESP}-1] \leq \text{EXPRSTACK}[\text{ESP}])$
 $? 1 : 0;$

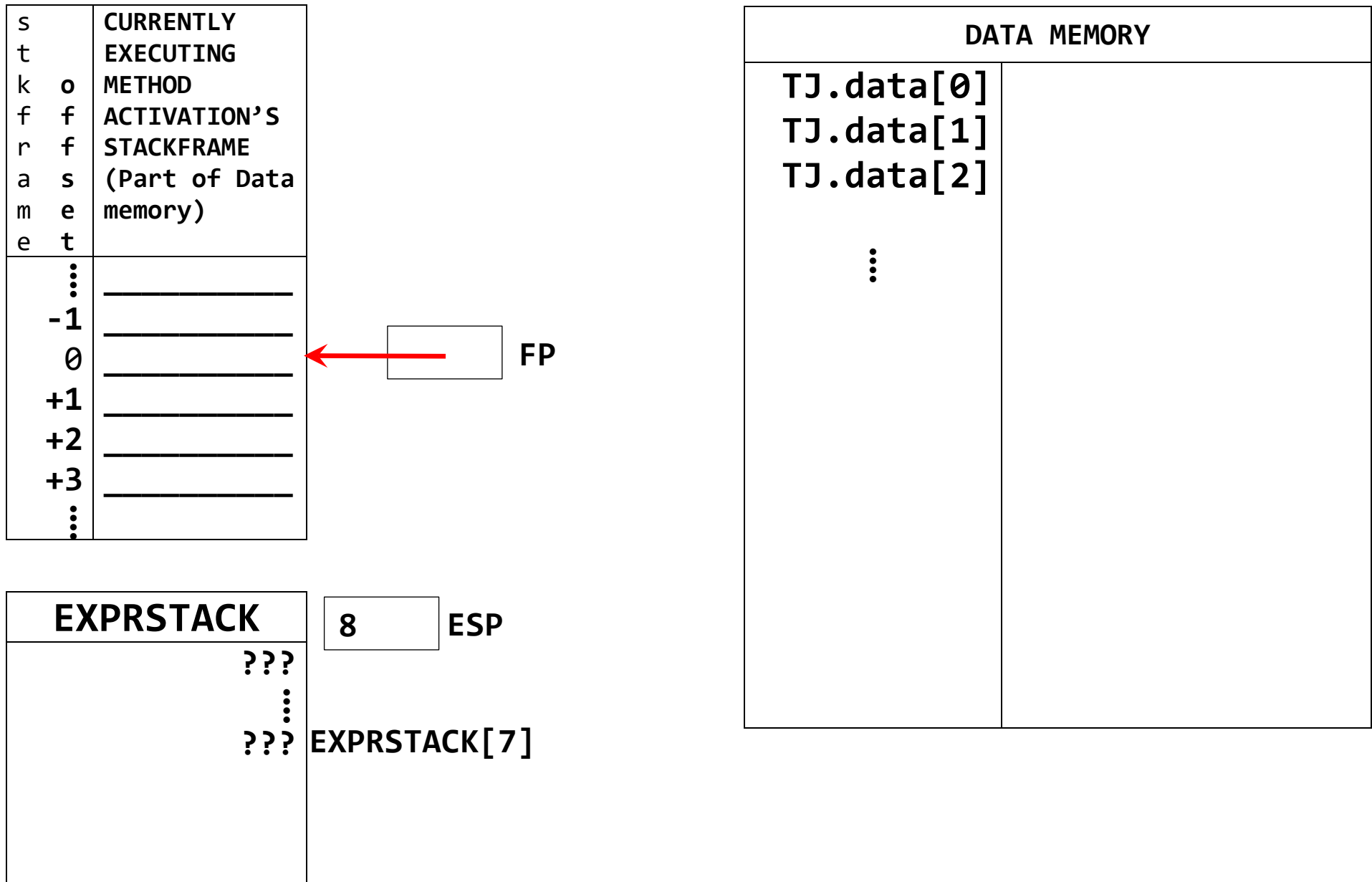
Before Execution of PUSHNUM 5:



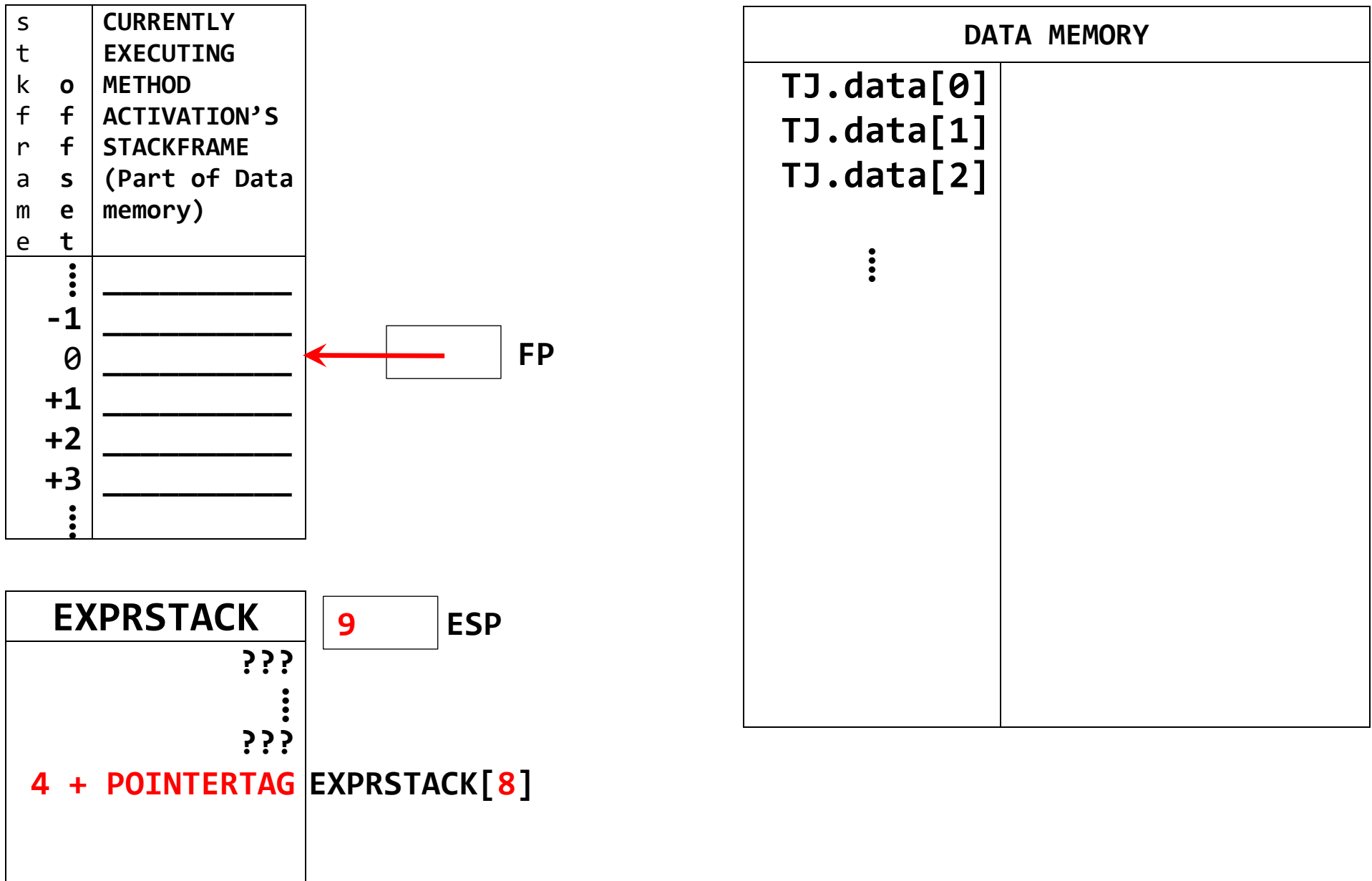
After Execution of PUSHNUM 5:



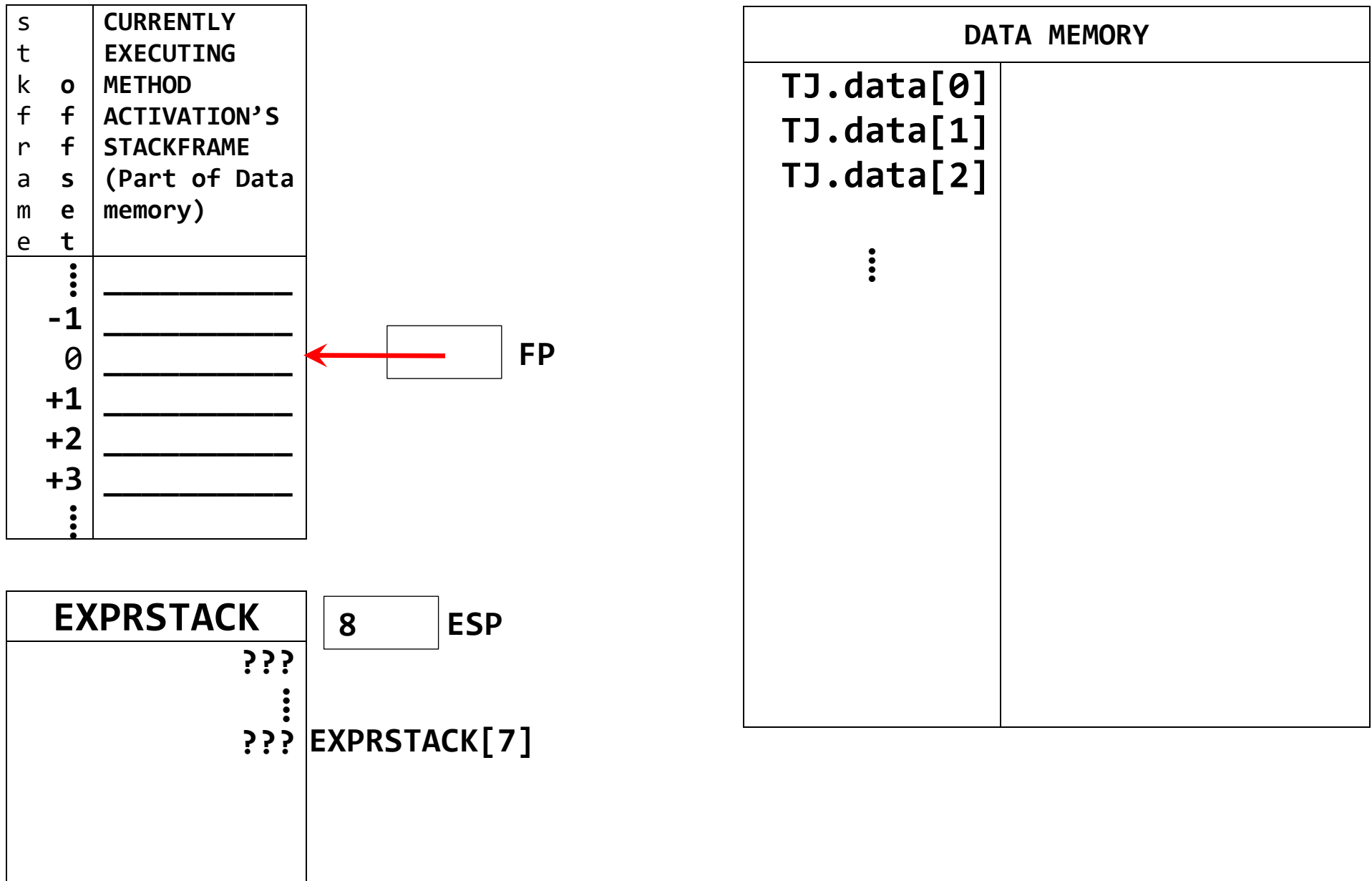
Before Execution of PUSHSTATADDR 4:



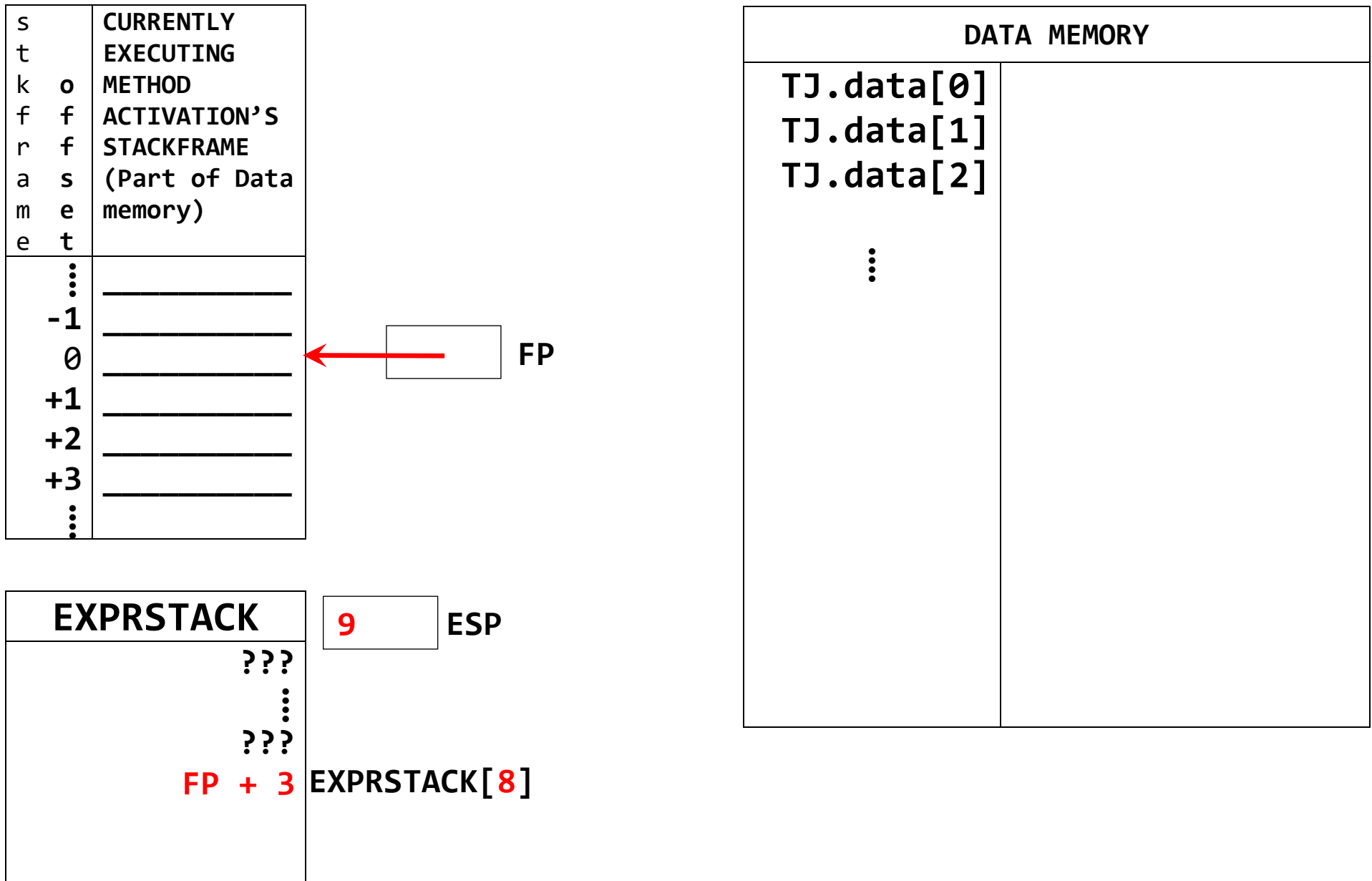
After Execution of PUSHSTATADDR 4:



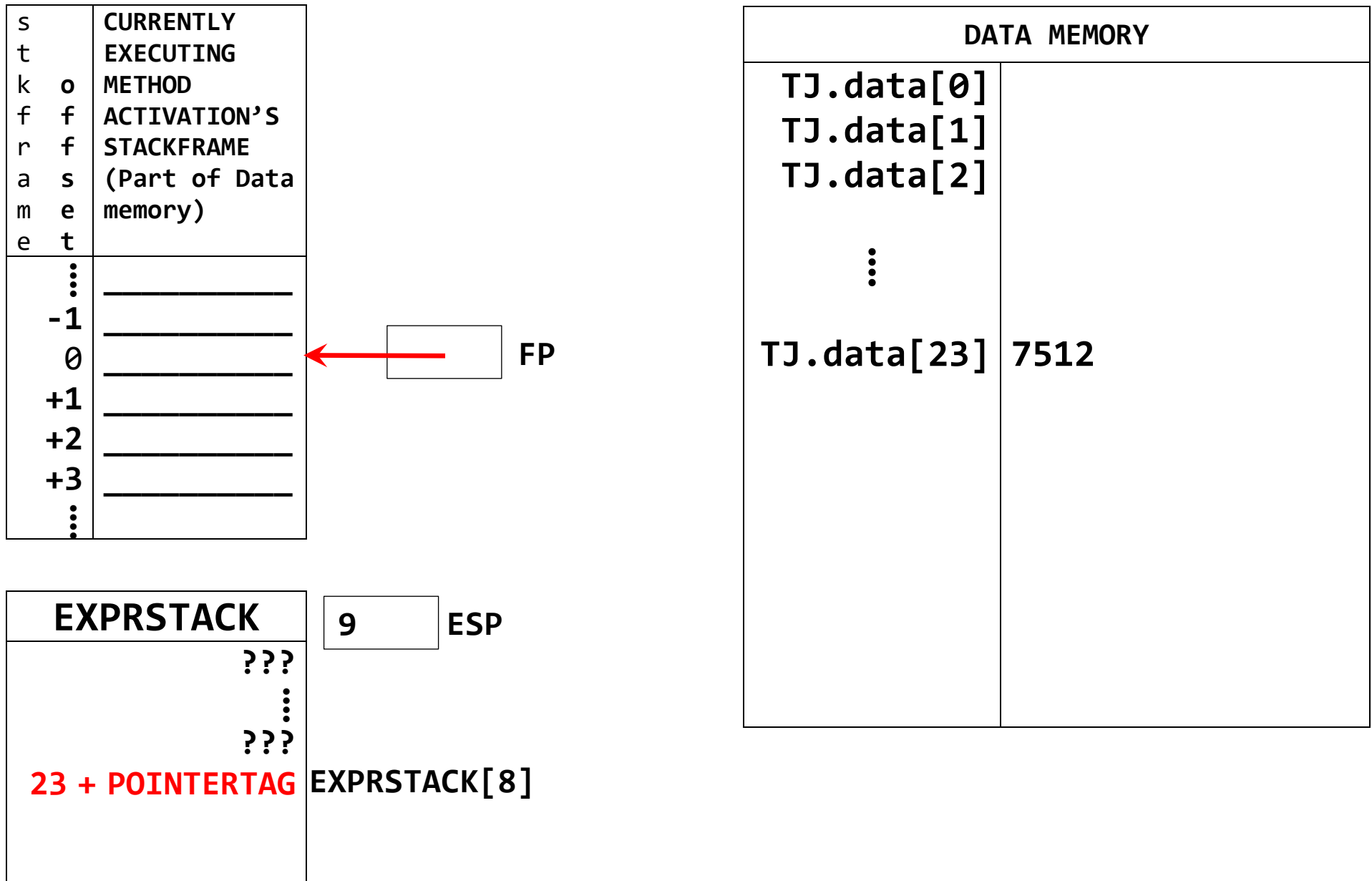
Before Execution of PUSHLOCADDR 3:



After Execution of PUSHLOCADDR 3:



Before Execution of LOADFROMADDR:



After Execution of LOADFROMADDR:

s t k o f f r f a s m e t	CURRENTLY EXECUTING METHOD ACTIVATION'S STACKFRAME (Part of Data memory)
⋮	_____
-1	_____
0	_____
+1	_____
+2	_____
+3	_____
⋮	_____



FP



DATA MEMORY	
TJ.data[0]	
TJ.data[1]	
TJ.data[2]	
⋮	
TJ.data[23]	

7512

EXPRSTACK	
???	⋮
???	
???	
7512	

9 ESP

EXPRSTACK[8]

Before Execution of SAVETOADDR:

s t k o f f r f a s m e t	CURRENTLY EXECUTING METHOD ACTIVATION'S STACKFRAME (Part of Data memory)
⋮	_____
-1	_____
0	_____
+1	_____
+2	_____
+3	_____
⋮	_____



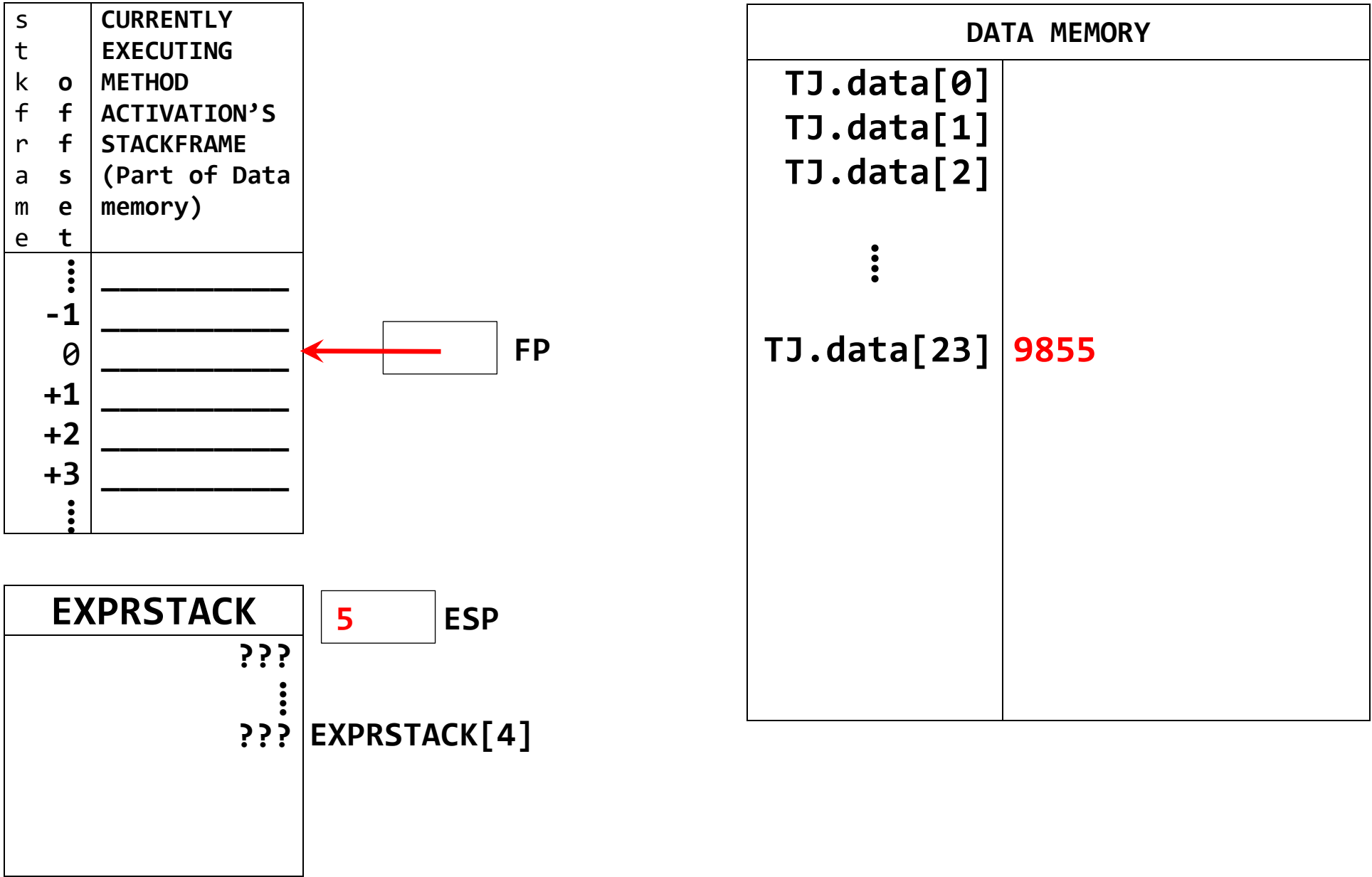
EXPRSTACK
???
⋮
???
23 + POINTERTAG
9855

7 ESP

EXPRSTACK[5]
EXPRSTACK[6]

DATA MEMORY	
TJ.data[0]	
TJ.data[1]	
TJ.data[2]	
⋮	
TJ.data[23]	???

After Execution of SAVETOADDR:



BEFORE execution of: **WRITESTRING 3 9**

s t k o f f r a m e t	CURRENTLY EXECUTING METHOD ACTIVATION'S STACKFRAME (Part of Data Memory)

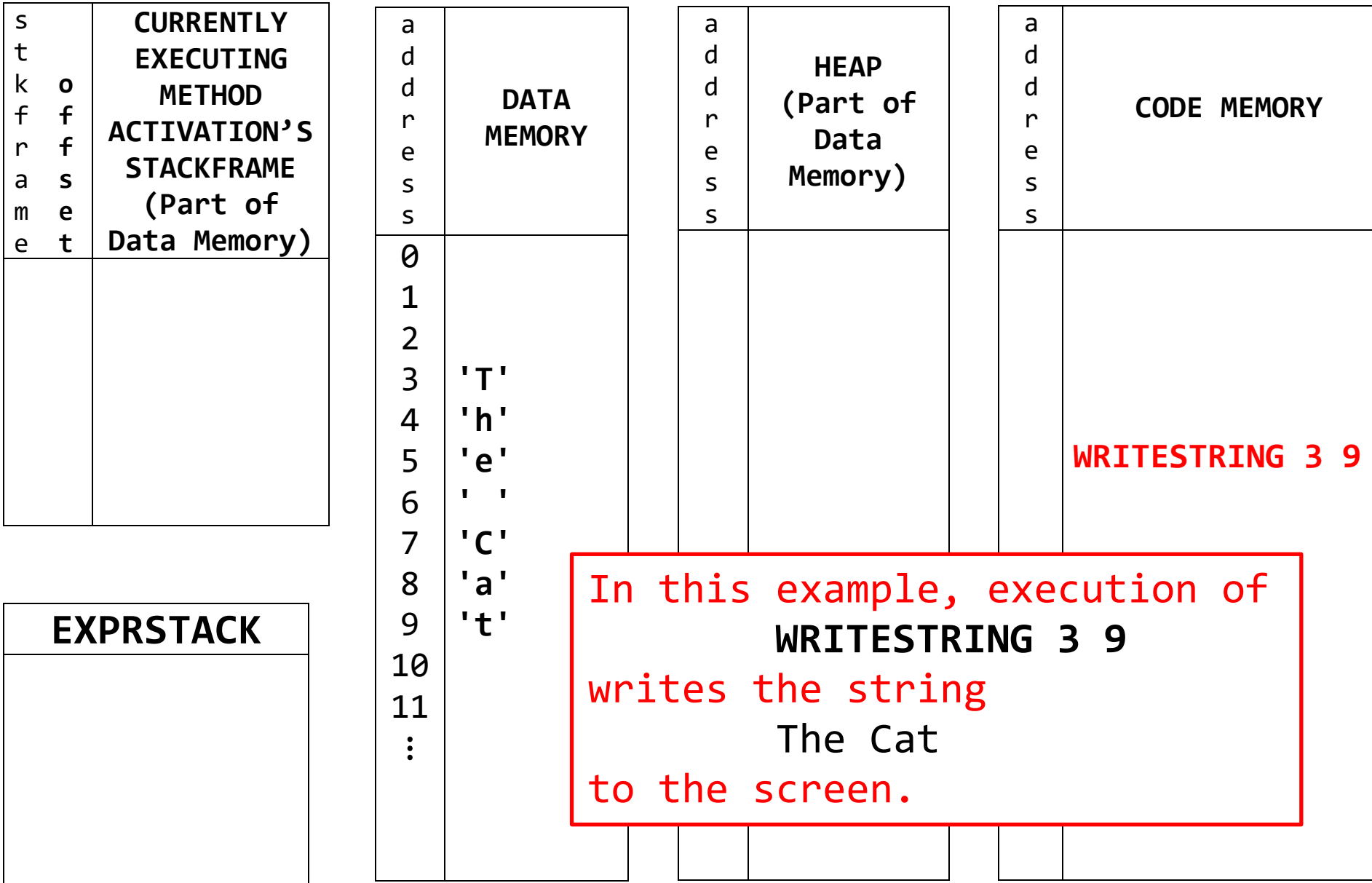
EXPRSTACK

a d d r e s s	DATA MEMORY
0	
1	
2	
3	'T'
4	'h'
5	'e'
6	' '
7	'C'
8	'a'
9	't'
10	
11	
⋮	

a d d r e s s	HEAP (Part of Data Memory)

a d d r e s s	CODE MEMORY
	WRITESTRING 3 9

AFTER execution of: **WRITESTRING 3 9**



BEFORE execution of: **JUMP 87**

PC 34

NOTE: The PC register is the program counter; it is used to store the code memory address of the next instruction to be executed.

a d d r e s s	CODE MEMORY
0	
1	
⋮	
33	JUMP 87
⋮	
87	

AFTER execution of: **JUMP 87**

PC

87

NOTE: The PC register is the program counter; it is used to store the code memory address of the next instruction to be executed.

a d d r e s s	CODE MEMORY
0	
1	
⋮	
33	JUMP 87
⋮	
87	

BEFORE execution of: **JUMPONFALSE 77** (Example 1)

PC 52

NOTE: The PC register is the program counter; it is used to store the code memory address of the next instruction to be executed.

EXPRSTACK

???

⋮

???

1

9

ESP

EXPRSTACK[8]

a d d r e s s	CODE MEMORY
0	JUMPONFALSE 77
1	
⋮	
51	
⋮	
77	

AFTER execution of: **JUMPONFALSE 77** (Example 1)

PC 52

NOTE: The PC register is the program counter; it is used to store the code memory address of the next instruction to be executed.

EXPRSTACK
???
⋮
???

8

ESP

a d d r e s s	CODE MEMORY
0	JUMPONFALSE 77
1	
⋮	
51	
⋮	
77	

BEFORE execution of: **JUMPONFALSE 77** (Example 2)

PC 52

NOTE: The PC register is the program counter; it is used to store the code memory address of the next instruction to be executed.

EXPRSTACK

???

⋮

???

0

9

ESP

EXPRSTACK[8]

a d d r e s s	CODE MEMORY
0	JUMPONFALSE 77
1	
⋮	
51	
⋮	
77	

AFTER execution of: **JUMPONFALSE 77** (Example 2)

PC

77

NOTE: The PC register is the program counter; it is used to store the code memory address of the next instruction to be executed.

EXPRSTACK

???
⋮
???

8

ESP

a
d
d
r
e
s
s

CODE MEMORY

0
1
⋮
51
⋮
77

JUMPONFALSE 77