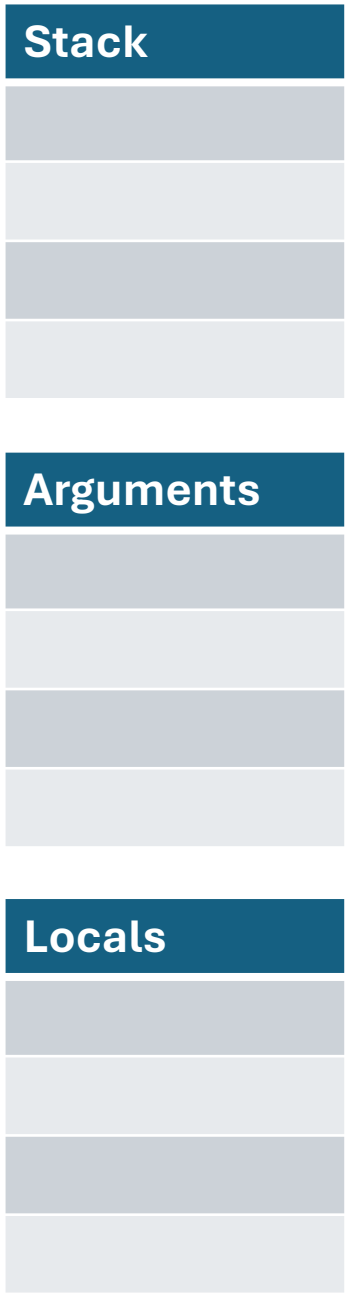
Virtual Machine Function Stack Example

-2: SP = 256
-1: goto Sys.init
0: function Add.add 0
1: push argument 0
2: push argument 1
3: add
4: return
5: function Main.main 2
6: push constant 7
7: pop local 0
8: push constant 8
9: pop local 1
10: push local 0
11: push local 1
12: call Add.add 2
13: pop local 0
14: push local 0
15: return
16: function Sys.init 0
17: call Main.main 0
18: label Sys.init$WHILE
19: goto Sys.init$WHILE

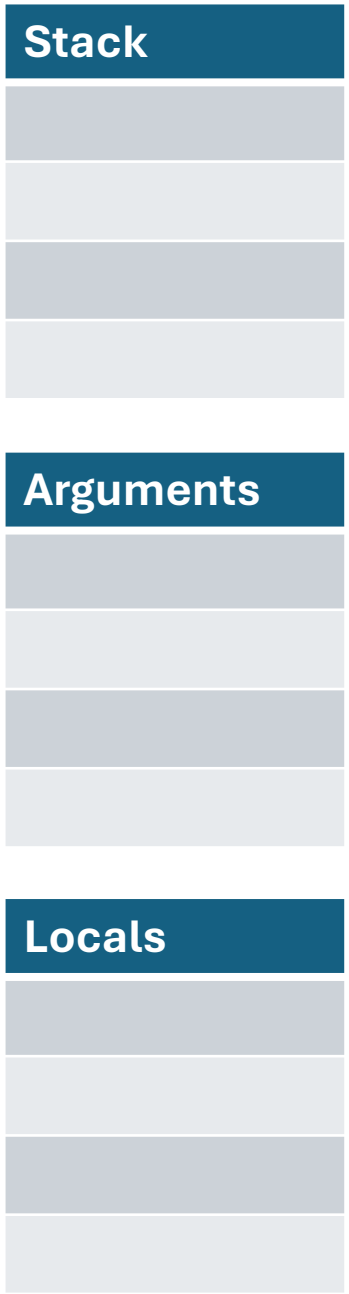


RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

```
push return-address
push LCL
push ARG
push THIS
push THAT
ARG = SP - numArgs - 5
LCL = SP
goto f
```

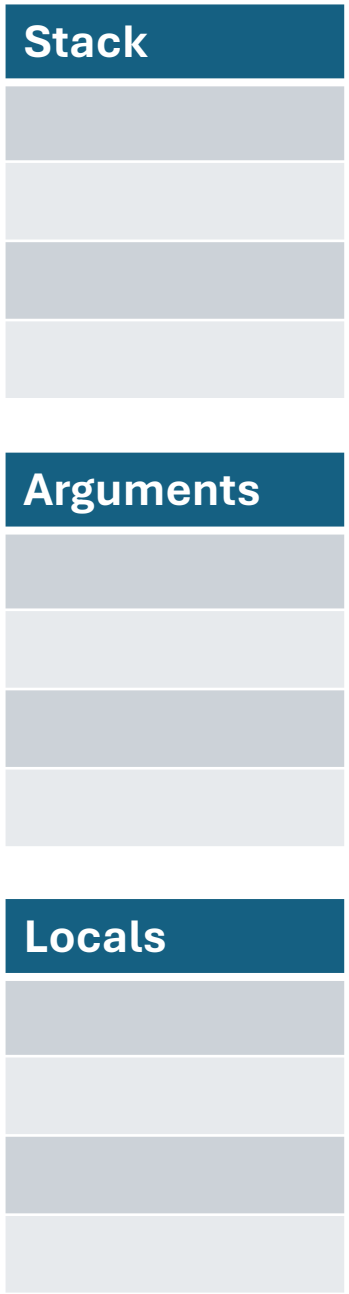


RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

```
repeat numLocals times:
  push 0
```



RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

Stack

Arguments

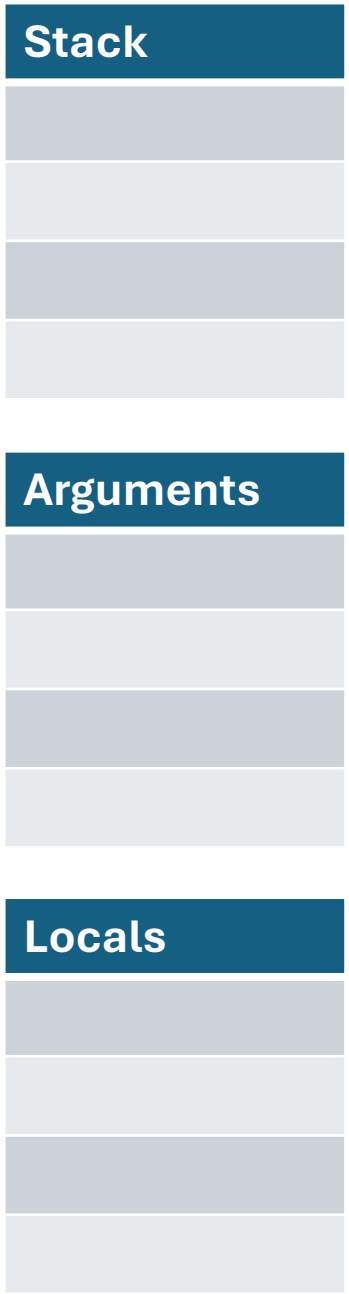
Locals

RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

```
push return-address
push LCL
push ARG
push THIS
push THAT
ARG = SP - numArgs - 5
LCL = SP
goto f
```

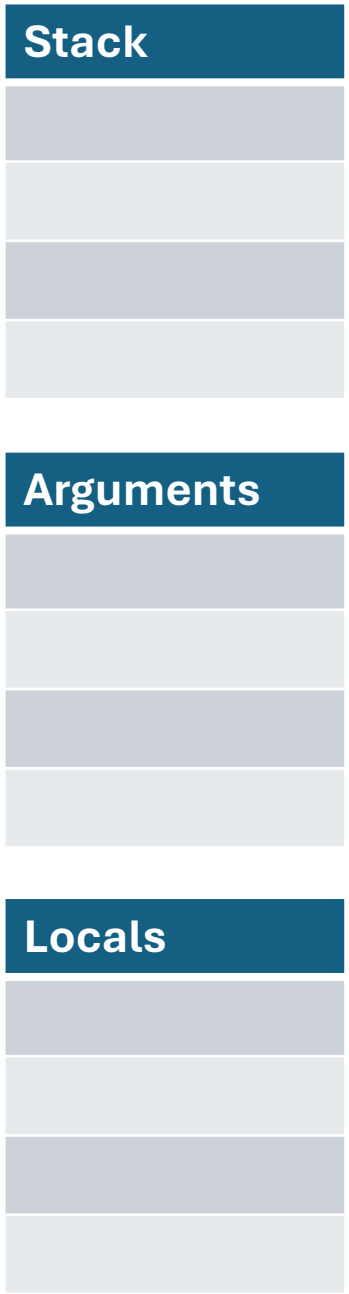


RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

-2: SP = 256
-1: goto Sys.init
0: function Add.add 0
1: push argument 0
2: push argument 1
3: add
4: return
5: function Main.main 2
6: push constant 7
7: pop local 0
8: push constant 8
9: pop local 1
10: push local 0
11: push local 1
12: call Add.add 2
13: pop local 0
14: push local 0
15: return
16: function Sys.init 0
17: call Main.main 0
18: label Sys.init$WHILE
19: goto Sys.init$WHILE

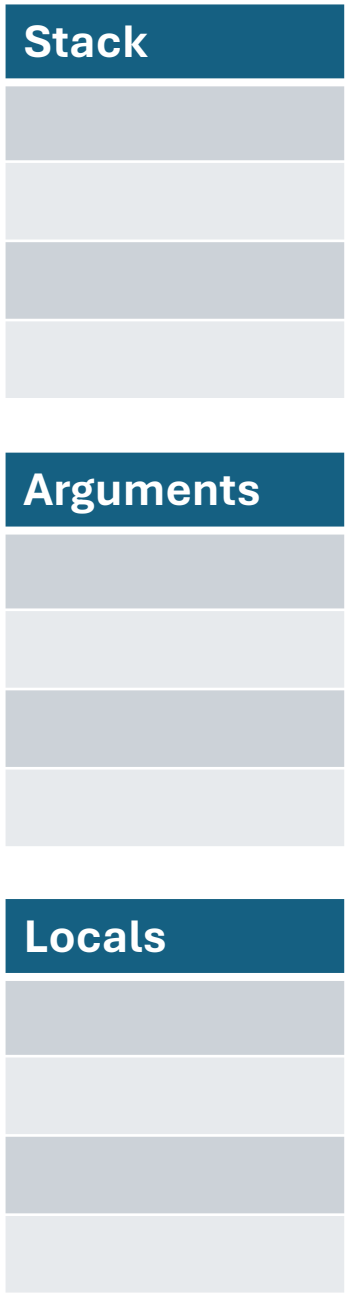
repeat numLocals times:
push 0



RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

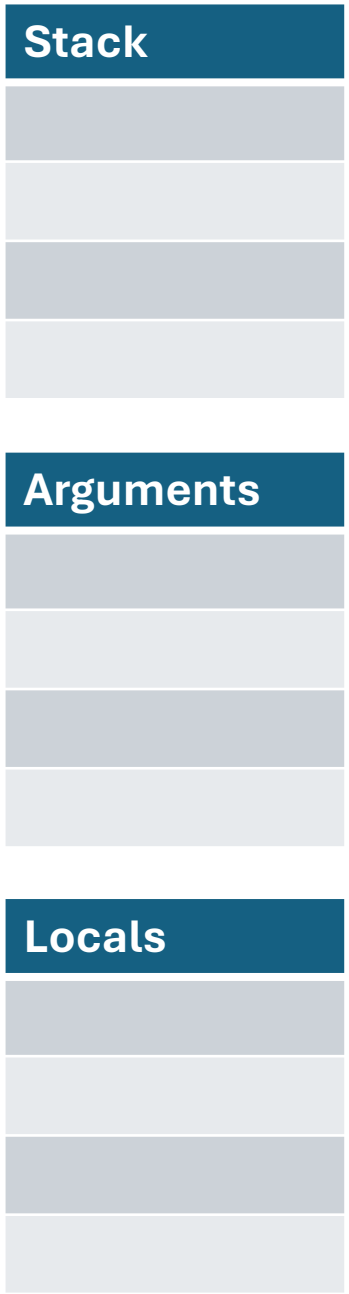


RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

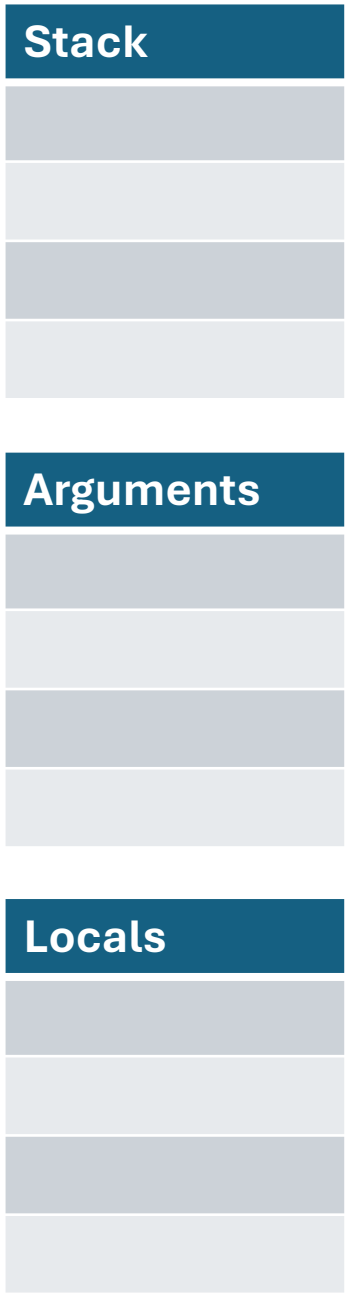
```
FRAME = LCL
RET = *(FRAME-5)
*ARG = pop()
SP = ARG+1
THAT = *(FRAME-1)
THIS = *(FRAME-2)
ARG = *(FRAME-3)
LCL = *(FRAME-4)
goto RET
```



RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

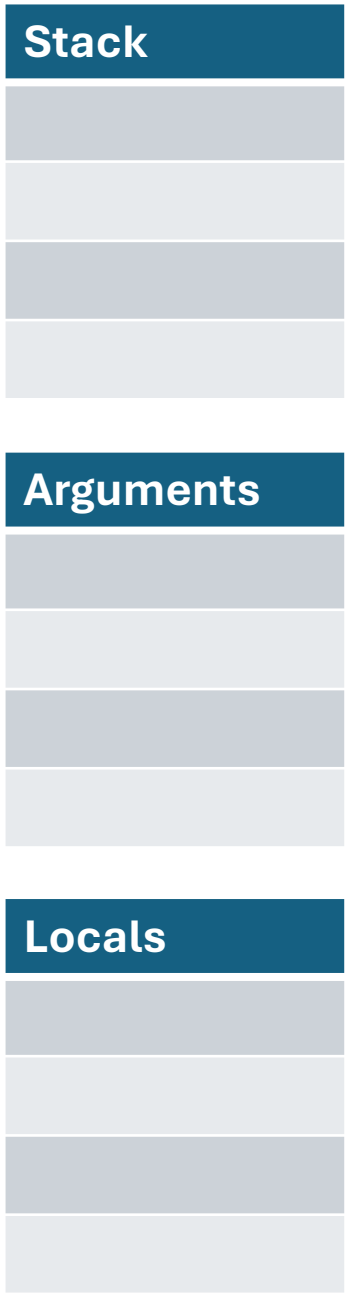


RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```

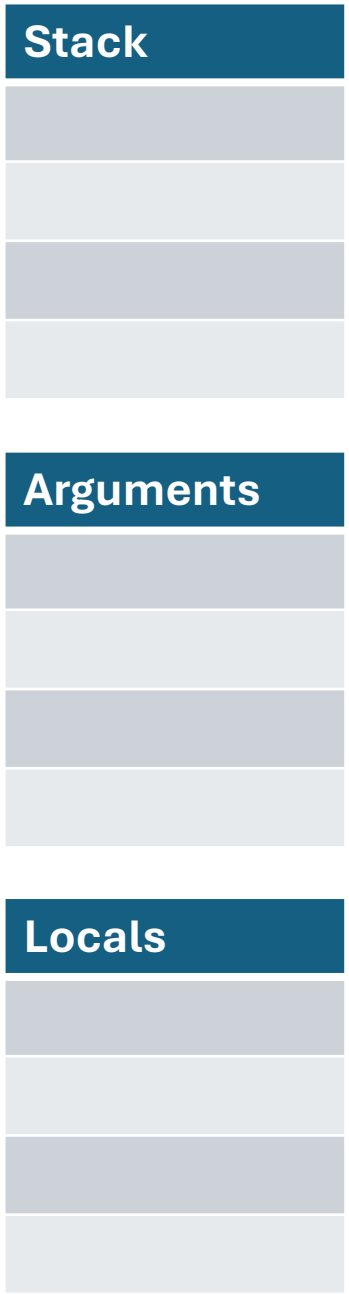
```
FRAME = LCL
RET = *(FRAME-5)
*ARG = pop()
SP = ARG+1
THAT = *(FRAME-1)
THIS = *(FRAME-2)
ARG = *(FRAME-3)
LCL = *(FRAME-4)
goto RET
```



RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	

```
-2: SP = 256
-1: goto Sys.init
0:  function Add.add 0
1:    push argument 0
2:    push argument 1
3:    add
4:    return
5:  function Main.main 2
6:    push constant 7
7:    pop local 0
8:    push constant 8
9:    pop local 1
10:   push local 0
11:   push local 1
12:   call Add.add 2
13:   pop local 0
14:   push local 0
15:   return
16: function Sys.init 0
17:   call Main.main 0
18:   label Sys.init$WHILE
19:   goto Sys.init$WHILE
```



RAM

Address	Value
0	
1	
2	
3	
4	
..	
13	
14	
...	
256	
257	
258	
259	
260	
261	
262	
263	
264	
265	
266	
267	
268	
269	
270	
271	
271	
273	