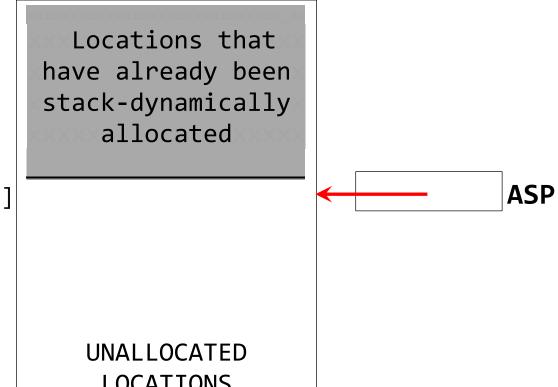
# Execution of Method Call and Return

## **S-PUSH** *y* is equivalent to: TJ.data[ASP - POINTERTAG] = y; ASP++;

#### **BEFORE** execution of S-PUSH y

Stack-Dynamically **Allocated Part** of Data Memory



TJ.data[ASP - POINTERTAG]

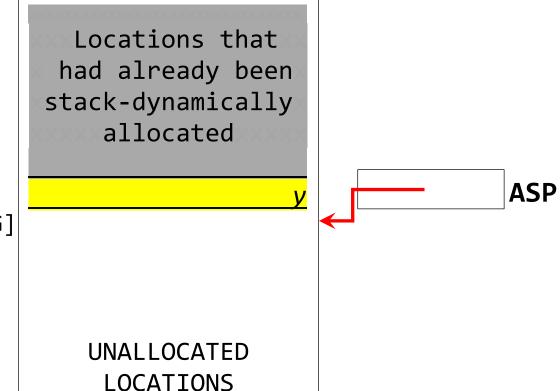
**LOCATIONS** 

```
S-PUSH y is equivalent to:
            TJ.data[ASP - POINTERTAG] = y; ASP++;
AFTER execution of TJ.data[ASP - POINTERTAG] = y;
                       Stack-Dynamically
                         Allocated Part
                         of Data Memory
                         Locations that
                       have already been
                       stack-dynamically
                           allocated
                                                       ASP
TJ.data[ASP - POINTERTAG]
                          UNALLOCATED
                           LOCATIONS
```

## 

#### AFTER execution of S-PUSH y

Stack-Dynamically
Allocated Part
of Data Memory

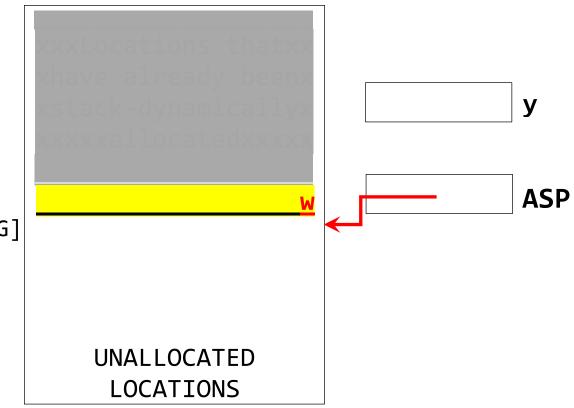


TJ.data[ASP - POINTERTAG]

# 

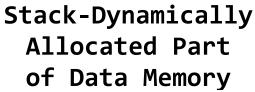
#### BEFORE execution of S-POP y

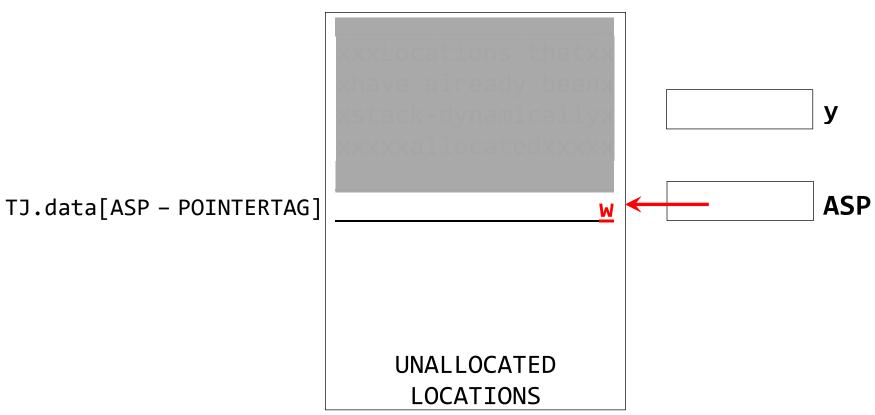




TJ.data[ASP - POINTERTAG]

#### **AFTER** execution of --ASP;

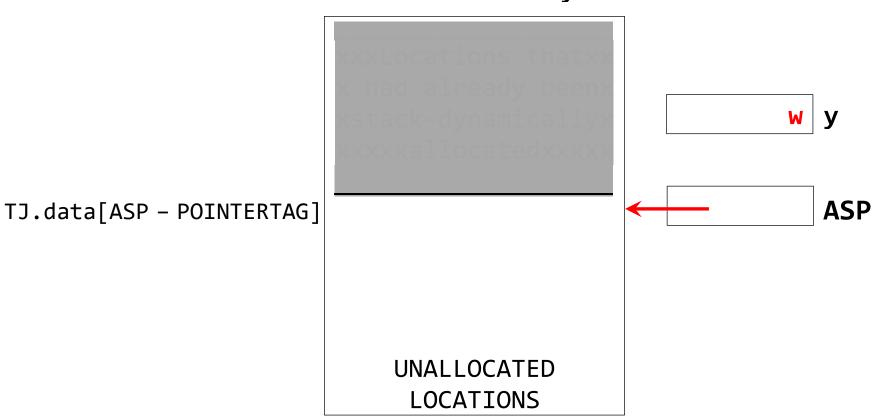




# 

#### AFTER execution of S-POP y





### TinyJ Assignment 3 Document

1. 7 stackframe locations are allocated for local variables declared in g's body.

- 1. 7 stackframe locations are allocated for local variables declared in g's body.
- 2. The code memory address of the first VM instruction generated for method g is 671.

- 1. 7 stackframe locations are allocated for local variables declared in g's body.
- 2. The code memory address of the first VM instruction generated for method g is 671.
- 3. Method g returns control to its caller by executing:
  713: RETURN 4

- 1. 7 stackframe locations are allocated for local variables declared in g's body.
- 2. The code memory address of the first VM instruction generated for method g is 671.
- 3. Method g returns control to its caller by executing:
  713: RETURN 4
- 4. The code memory address of the first VM instruction generated for g(17,23,7,3) is 44.

### TinyJ Assignment 3 Document

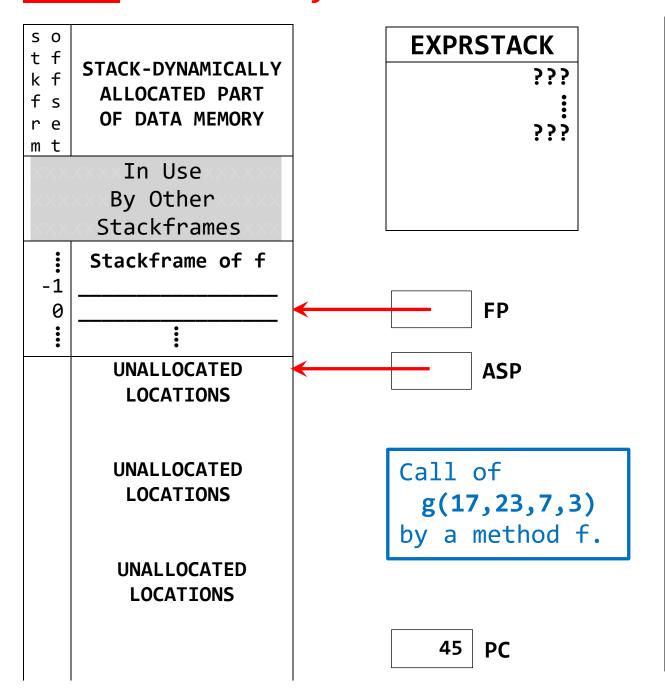
```
Suppose a method f calls a method g as follows g(17,23,7,3)
--e.g., within: System.out.print(g(17,23,7,3));
```

#### Suppose further that:

- 1. 7 stackframe locations are allocated for local variables declared in g's body.
- 2. The code memory address of the first VM instruction generated for method g is 671.
- 3. Method g returns control to its caller by executing:
  713: RETURN 4
- 4. The code memory address of the first VM instruction generated for g(17,23,7,3) is 44.

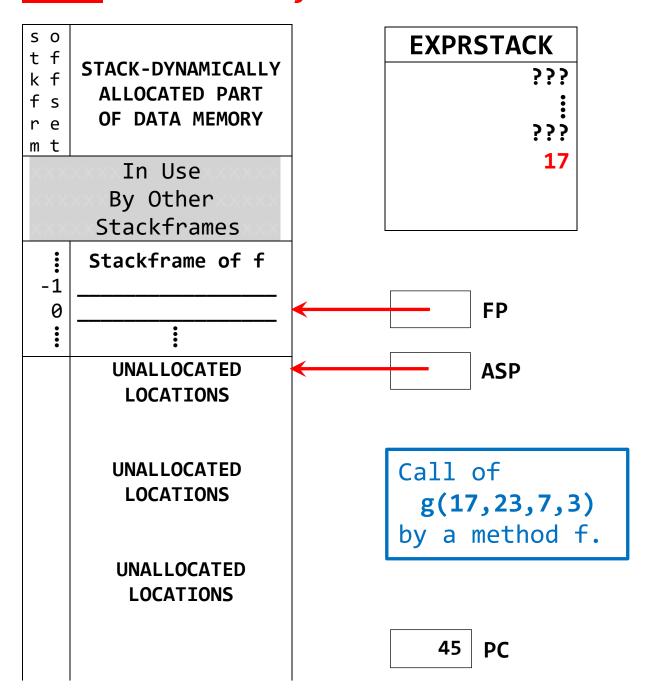
The following slides show how the call g(17,23,7,3) would be executed, and how 713: RETURN 4 would be executed.

#### **BEFORE** Execution of: 44: PUSHNUM 17



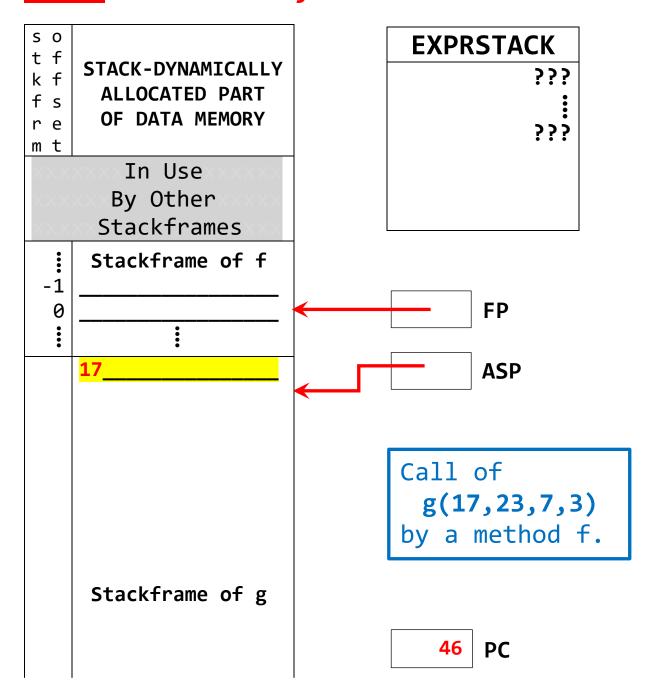
a d d r e s	CODE MEMORY
0 44 45 46 47 48 49 50 51 52 53	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM PUSHNUM 7 CODE PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
671 	INITSTKFRM 7 g's : RETURN 4 code

#### AFTER Execution of: 44: PUSHNUM 17



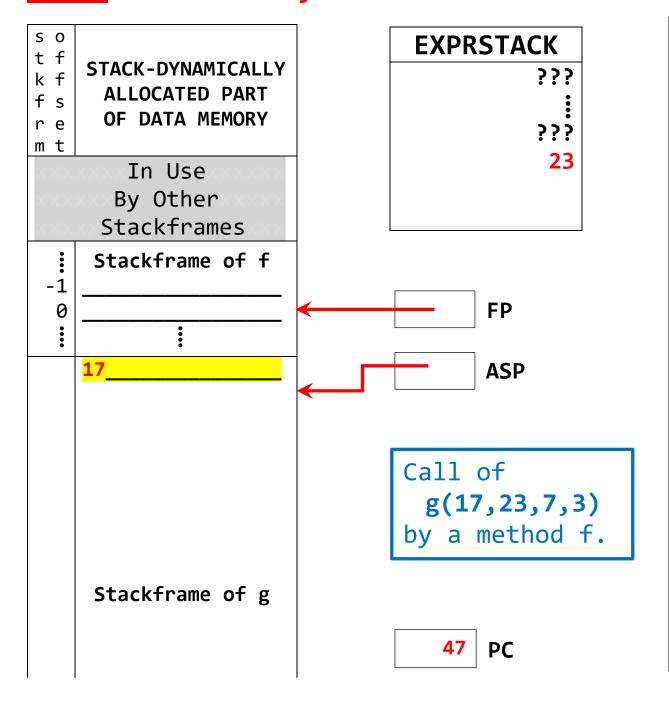
a d d r e s s	CODE MEMORY
0 44 45 46 47 48 49 50 51 52 53	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM PUSHNUM 7 CODE PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
:	
671 : 713	INITSTKFRM 7 g's .: RETURN 4 code

#### **AFTER** Execution of 45: PASSPARAM



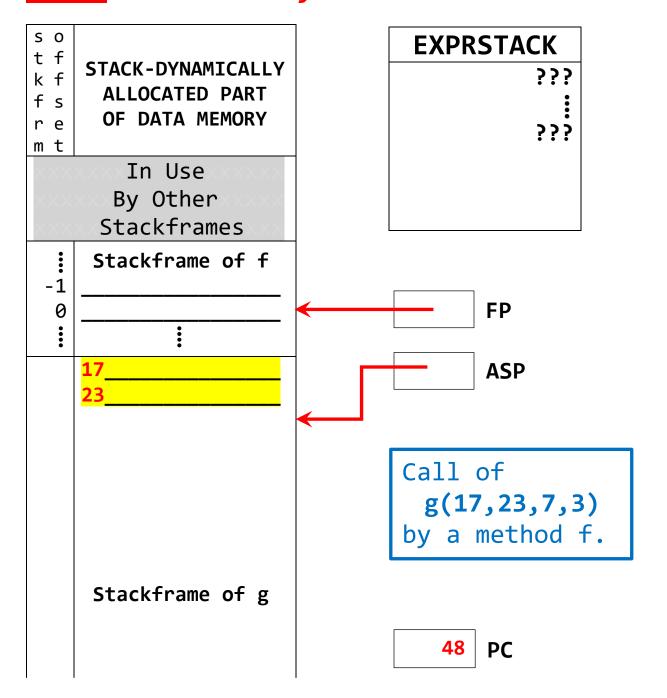
a d d r	CODE MEMORY
S S	
0 :: 44 45 46 47	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM  f's
48 49 50 51 52 53	PUSHNUM 7 CODE PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
671 	INITSTKFRM 7 g's  : RETURN 4 code

#### **AFTER** Execution of 46: PUSHNUM 23



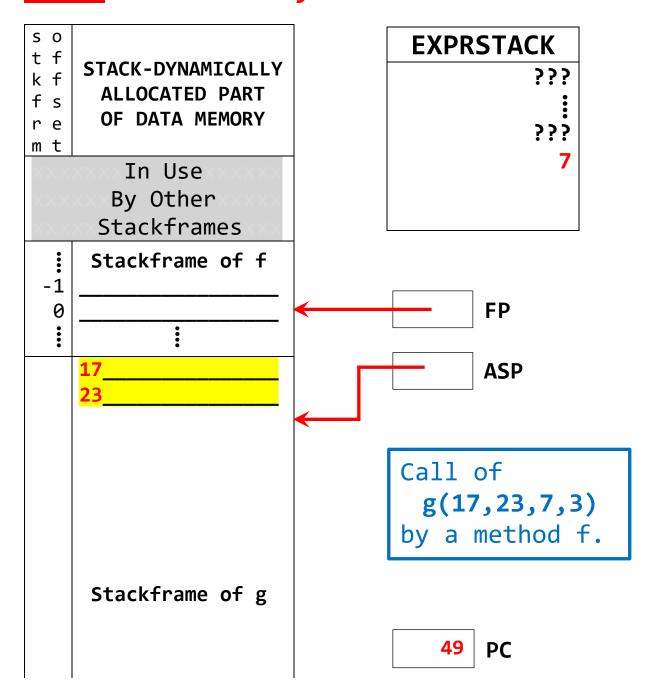
а	
d	
d	CODE MEMORY
r	CODE MEMORY
е	
S	
S	
0	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23
47	PASSPARAM f's
48	PUSHNUM 7 code
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	CALLSTATTIETTIOD 071
))	
•	
671	INITSTKFRM 7 g's
	code
713	RETURN 4

#### **AFTER** Execution of 47: PASSPARAM



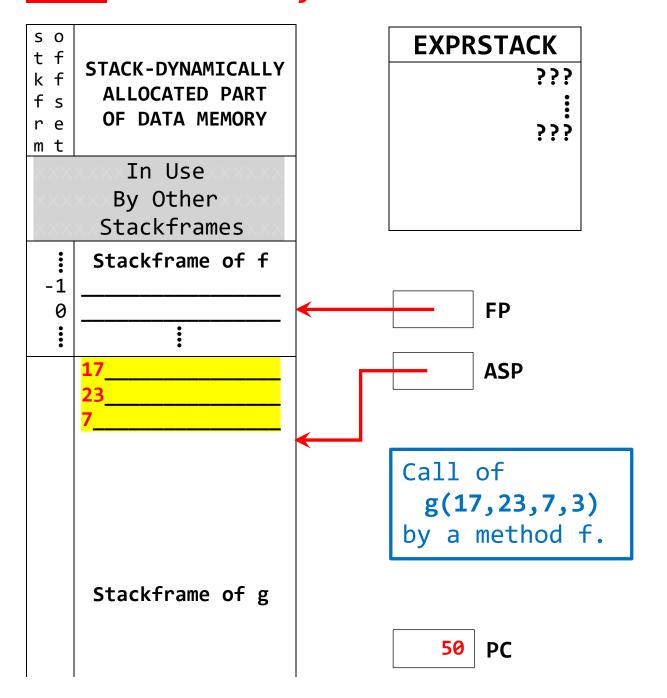
a d d r e s s	CODE MEMORY
0 44 45 46 47 48 49 50 51 52 53	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM PUSHNUM 7 code PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
:	
671 : 713	INITSTKFRM 7 g's  : RETURN 4 code

#### AFTER Execution of 48: PUSHNUM 7



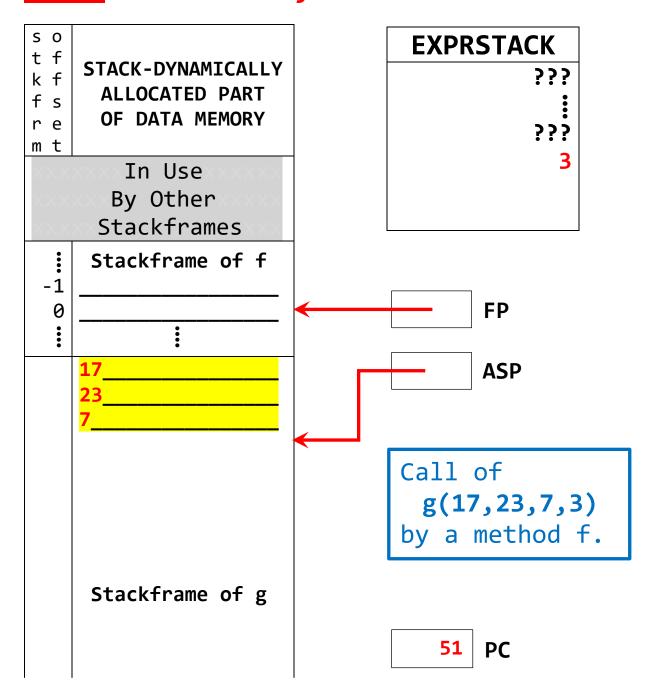
a d d r e s	CODE MEMORY
<ul> <li>0</li> <li>44</li> <li>45</li> <li>46</li> <li>47</li> <li>48</li> <li>49</li> <li>50</li> <li>51</li> <li>52</li> <li>53</li> </ul>	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM PUSHNUM 7 CODE PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
671 : 713	INITSTKFRM 7 g's  : RETURN 4 code

#### **AFTER** Execution of 49: PASSPARAM



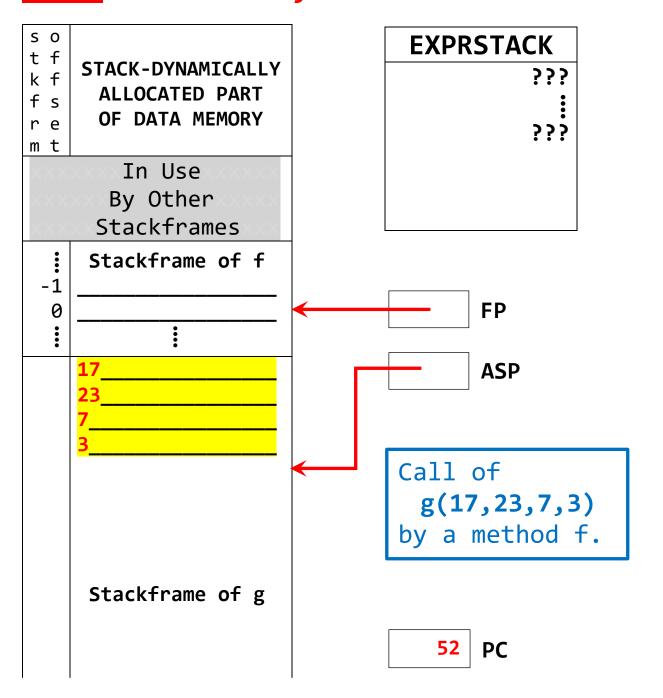
а	
d	
d	CODE MEMORY
r	CODE MEMORY
е	
S	
S	
0	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23 f's
47	PASSPAKAM
48	PUSHNUM 7 code
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
i	
:	
671	INITSTKFRM 7 \sigma' \sigma'
:	: 8
; 713	RETURN 4 code
/13	REIUKN 4

#### **AFTER** Execution of 50: PUSHNUM 3



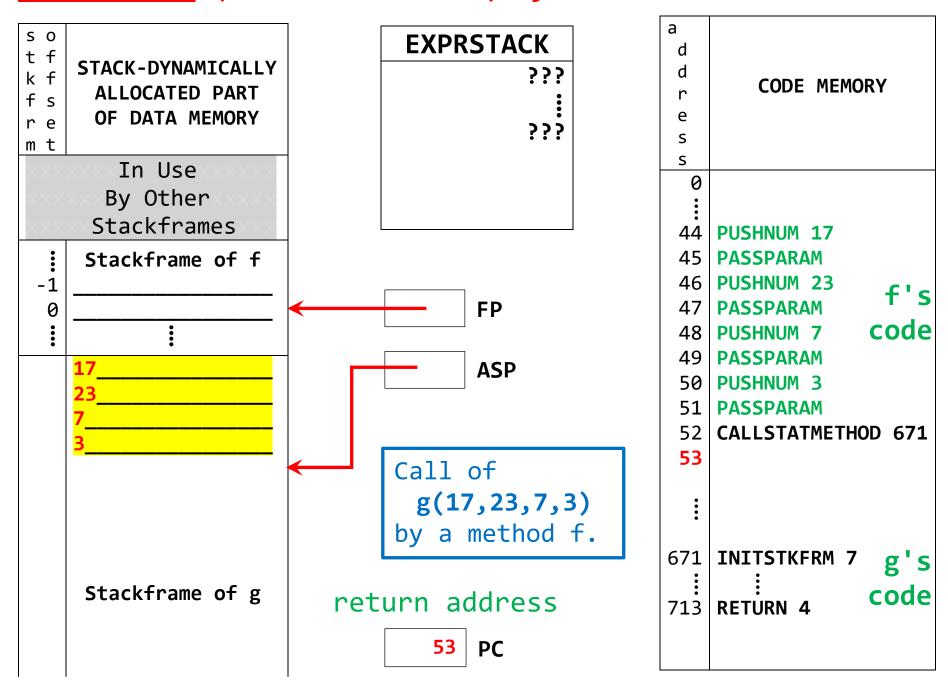
a	
d	
d	CODE MEMORY
r	
e s	
s S	
0	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23
47	PASSPARAM f's
48	PUSHNUM 7 code
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	CALLSTATMETHOD 071
))	
•	
:	
671	INITSTKFRM 7 g's
:	•
713	RETURN 4 code
3	

#### **AFTER** Execution of 51: PASSPARAM



a d d r e s	CODE MEMORY
0 44 45 46 47 48 49 50 51 52 53	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM PUSHNUM 7 CODE PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
671 : 713	INITSTKFRM 7 g's  : RETURN 4 code

#### After FETCH (BEFORE Execution) of 52: CALLSTATMETHOD 671

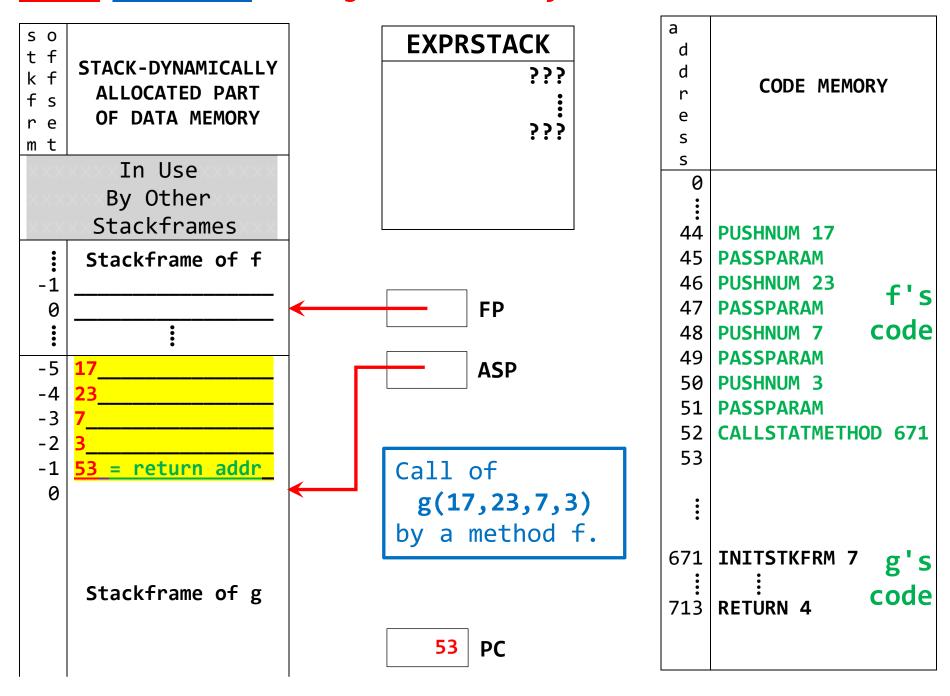


#### **52: CALLSTATMETHOD 671**

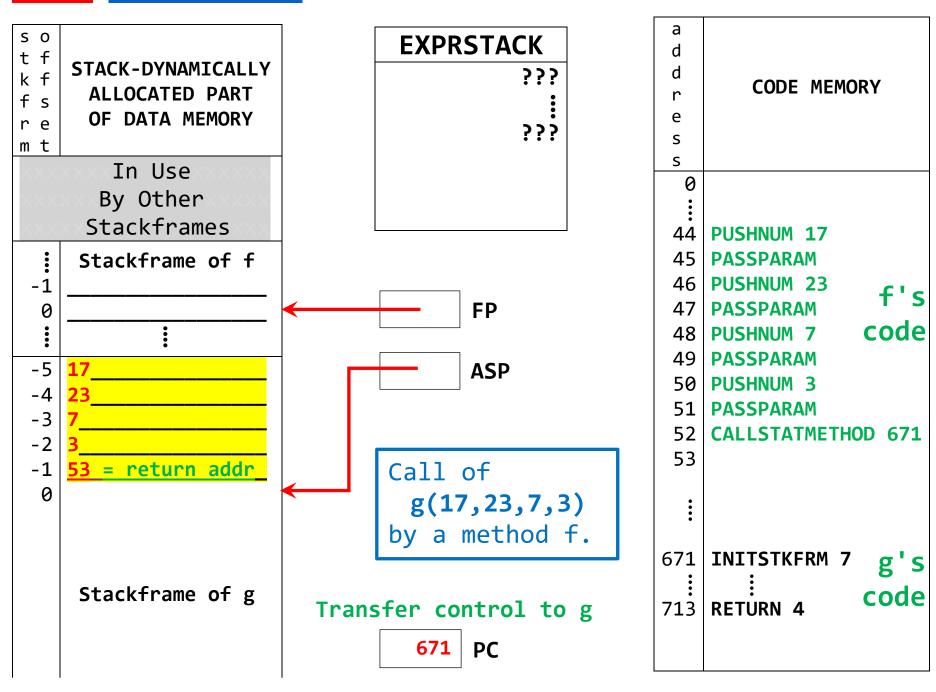
should S-PUSH PC [saves return addr (here, 53) into new frame\*] and then set PC to 671 [transfers control to g's code]

\*into the loc. at offset -1 of the new frame - see p. 4 of:
euclid.cs.qc.cuny.edu/316/Memory-allocation-VM-instruction-set-and-hints-for-asn-2.pdf

### AFTER S-PUSH PC during Execution of 52: CALLSTATMETHOD 671



#### AFTER Execution of 52: CALLSTATMETHOD 671



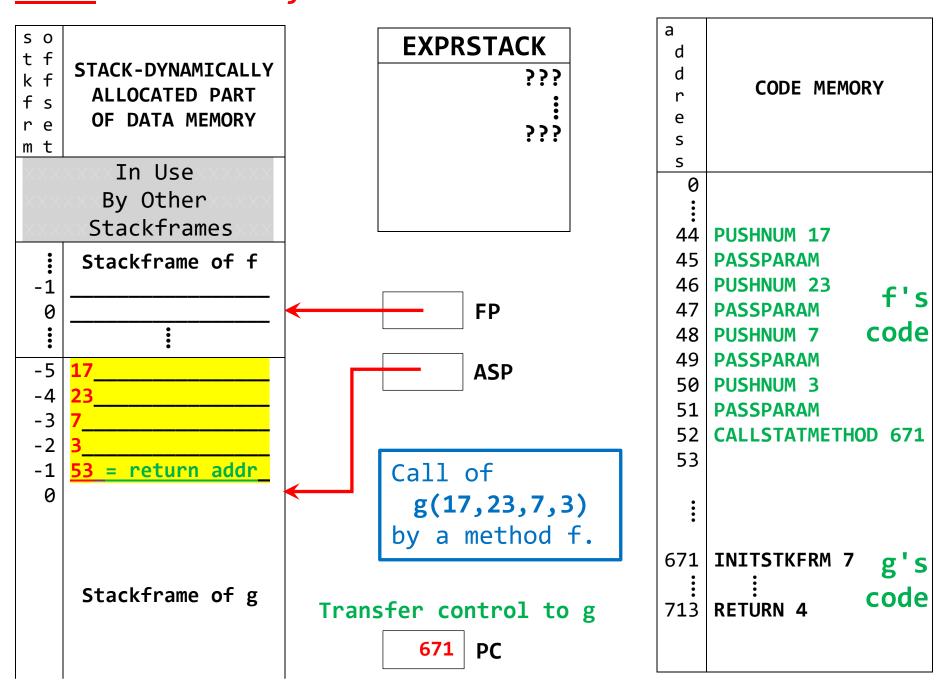
#### 671: INITSTKFRM 7

should **S-PUSH** FP [saves caller's FP at offset 0\* in the new frame] and then set FP to ASP – 1 [makes FP point to offset 0\* in the new frame] and then increase ASP by 7 [allocates space for callee's local variables]

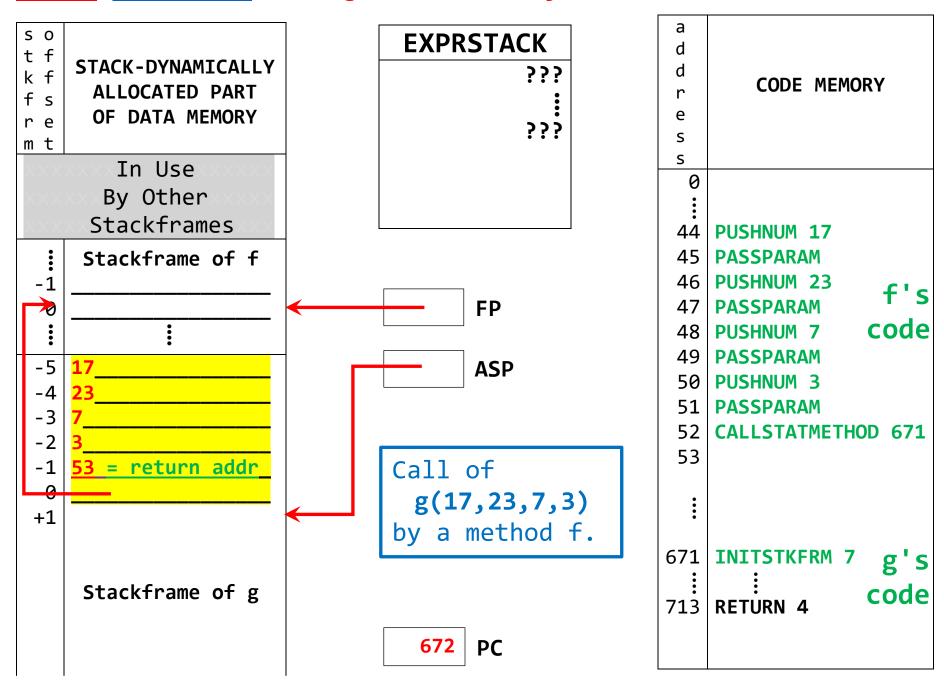
euclid.cs.qc.cuny.edu/316/Memory-allocation-VM-instruction-set-and-hints-for-asn-2.pdf

<sup>\*</sup>After the caller's FP is stored at offset 0 in the new frame, the stored pointer is called the <u>dynamic link</u>. See p. 4 of:

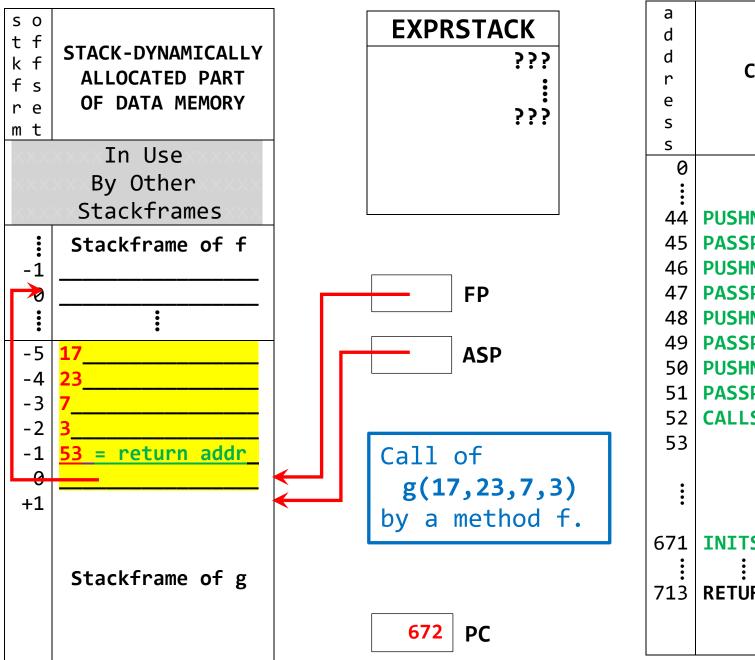
#### AFTER Execution of 52: CALLSTATMETHOD 671



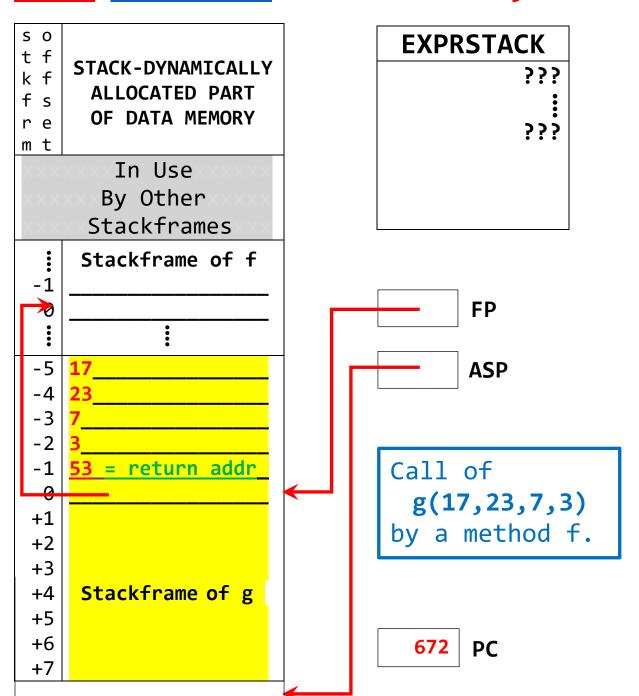
### After S-PUSH FP during Execution of 671: INITSTKFRM 7



## After Step 2 of Execution of 671: INITSTKFRM 7



## After Final Step of Execution of 671: INITSTKFRM 7



a	
d	
d	CODE MEMORY
r	CODE PIERIORI
e	
S	
S	
0	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23  F'S
47	PASSPARAM
48	PUSHNUM 7 code
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
:	
•	
671	INITSTKFRM 7 g's
:	•
713	RETURN 4 code
, 10	ILLIONIA T

#### 713: RETURN 4

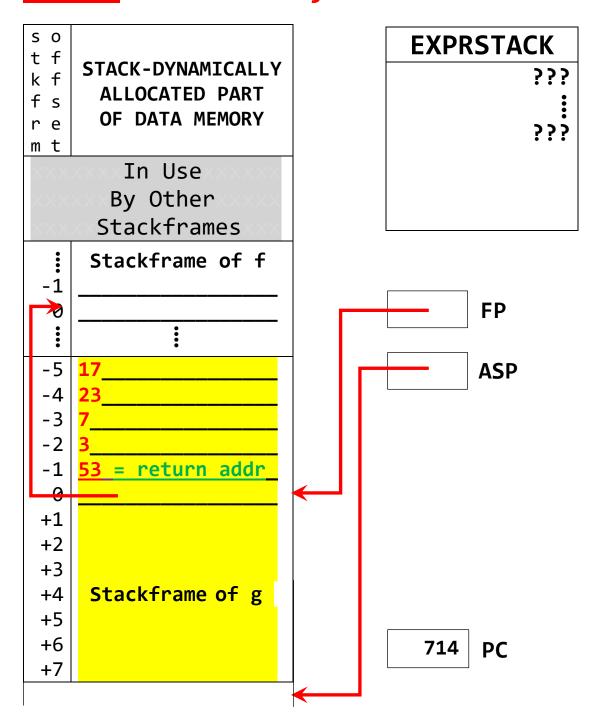
should set ASP to FP+1 [deallocates space used by callee's variables]

and then **S-POP** FP [restores caller's FP]

and then **S-POP** PC [puts the saved return address into PC]

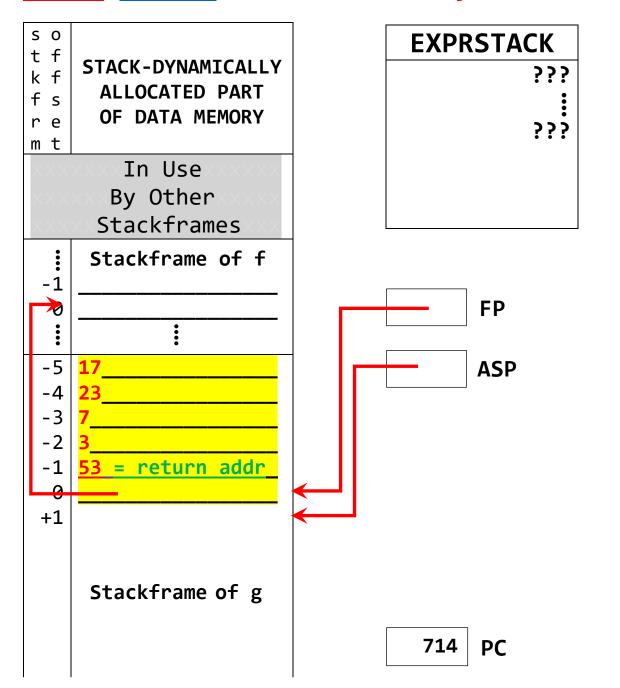
and then decrease ASP by 4 [deallocates space used by formal parameters]

#### **BEFORE** Execution of 713: RETURN 4



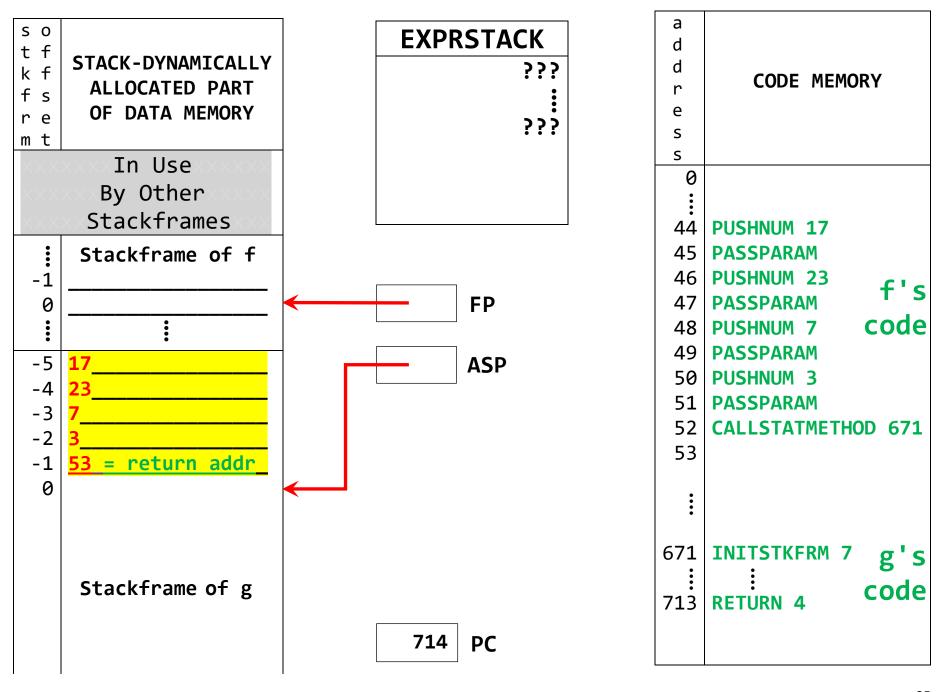
a d d r e s s	CODE MEMORY
0 44 45 46 47 48 49 50 51 52 53	PUSHNUM 17 PASSPARAM PUSHNUM 23 PASSPARAM PUSHNUM 7 code PASSPARAM PUSHNUM 3 PASSPARAM CALLSTATMETHOD 671
671 : 713	INITSTKFRM 7 g's : RETURN 4 code

# After Step 1 of Execution of 713: RETURN 4

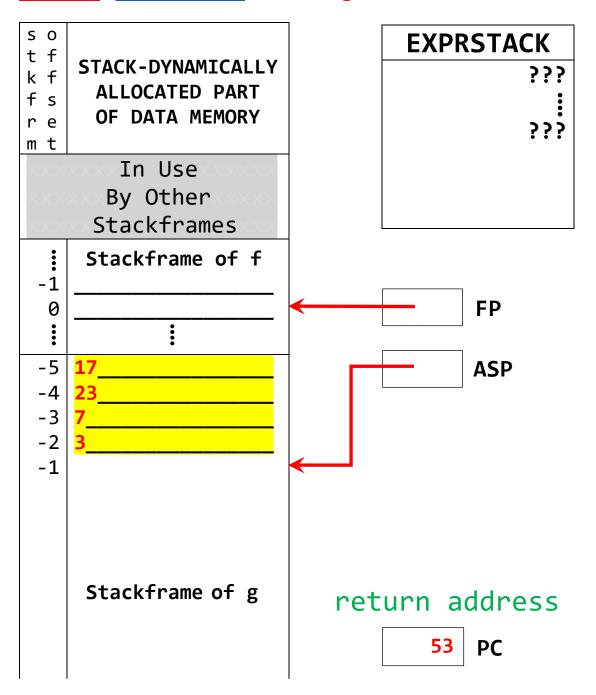


а	
d	
d	CODE MEMORY
r	CODE MEMORI
е	
S	
S	
0	
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23  F'S
47	PASSPARAM T S
48	PUSHNUM 7 code
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	CALLSTATTICTION 071
در	
•	
671	INITSTKFRM 7 g's
	code
713	RETURN 4

# After S-POP FP during Execution of 713: RETURN 4



### After S-POP PC during of Execution of 713: RETURN 4



a	
d	
d	CODE MEMORY
r	CODE TIERIORI
e	
S S	
0	
_	
11	DUCUMUM 17
44	PUSHNUM 17
45	PASSPARAM
46	PUSHNUM 23 f's
47	PASSPARAM
48	PUSHNUM 7 code
49	PASSPARAM
50	PUSHNUM 3
51	PASSPARAM
52	CALLSTATMETHOD 671
53	
•	
671	INITSTKFRM 7 g's
:	•
713	RETURN 4 code
, 1	

### After Final Step of Execution of 713: RETURN 4

