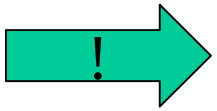
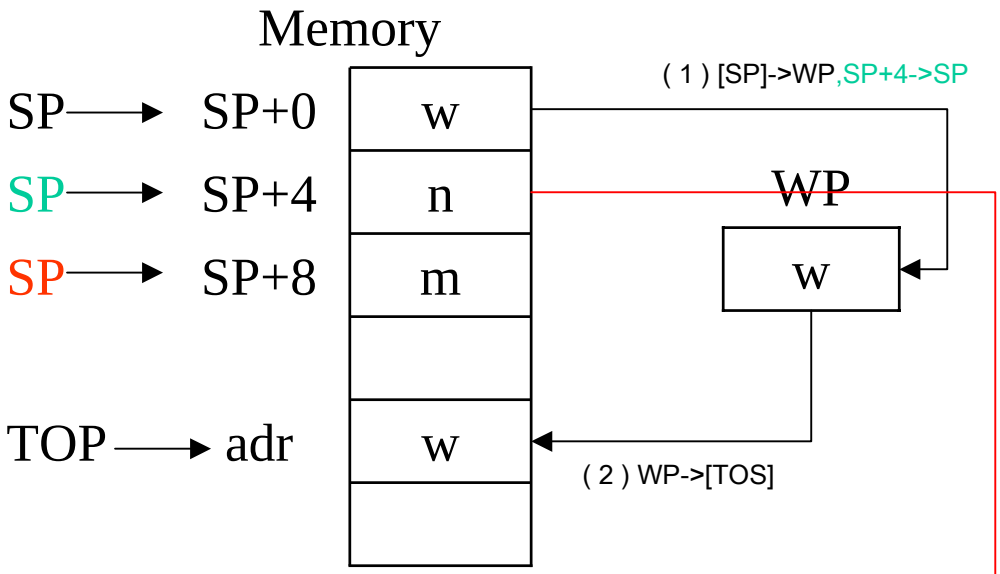
! (w adr --)

```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:





Data Stack	
	w
TOS	n
SP+0	m
SP+4	:

(3) [SP]->TOS,SP+4->SP

! (w adr --)

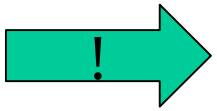
```
CODE ! ( w a -- ) \ Pop the data stack to memory.
  ( 1 ) [SP]->WP,SP+4->SP
  ( 2 )  WP->[TOS]
  ( 3 ) [SP]->TOS,SP+4->SP
   $NEXT \ NEST
   END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:

! (w adr --)

```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

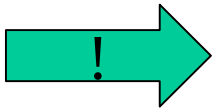
Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:

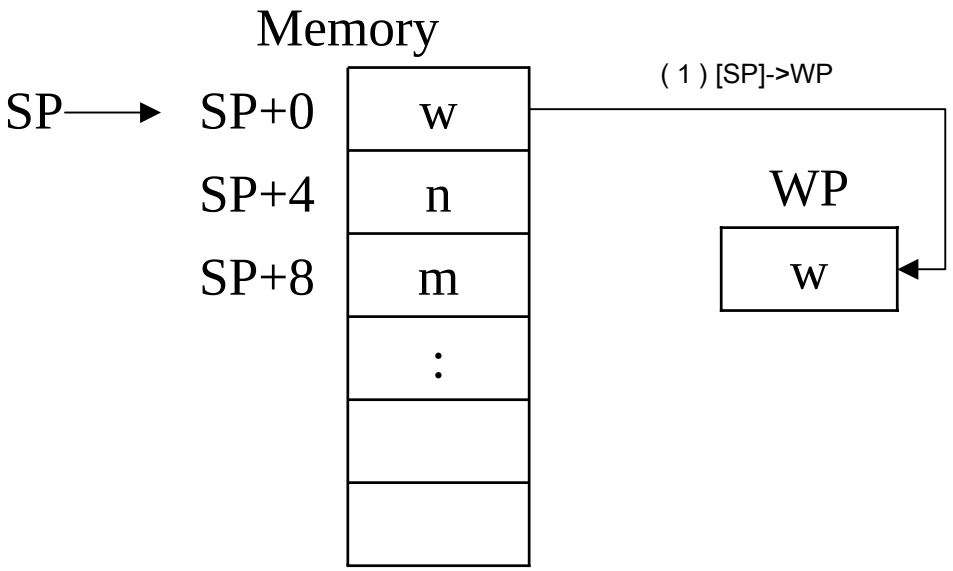


! (w adr --)

```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:

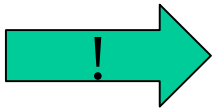


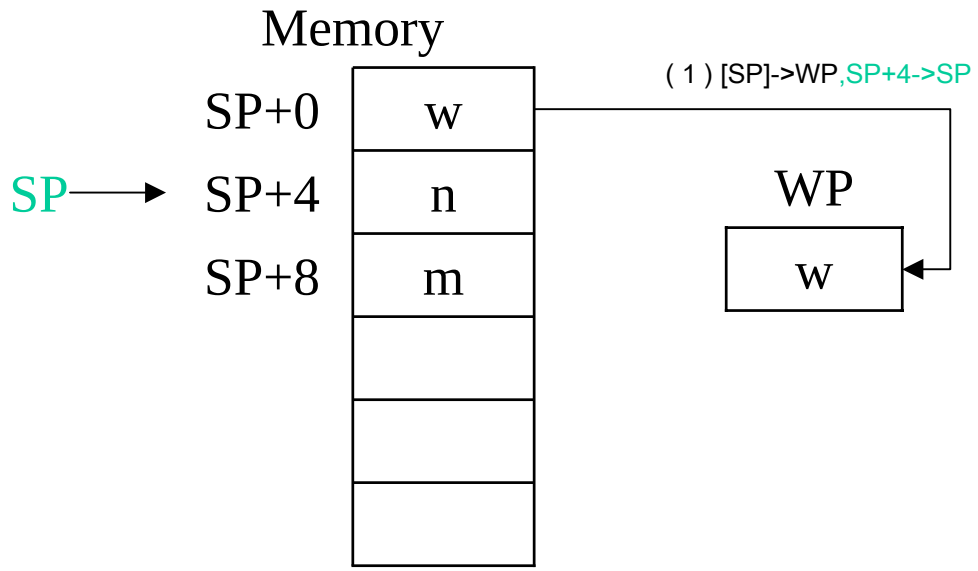


! (w adr --)

```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:

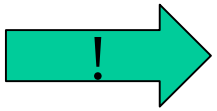


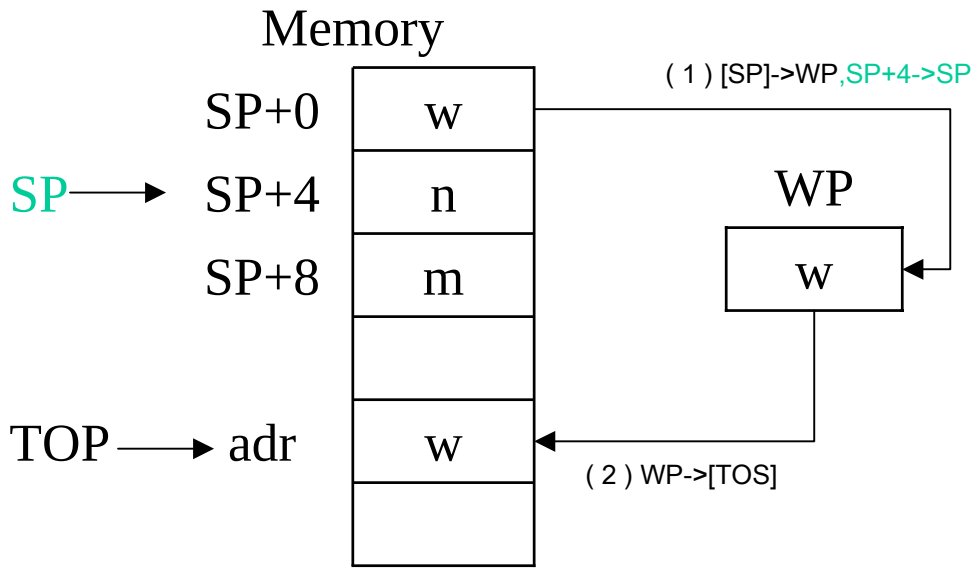


! (w adr --)

```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:

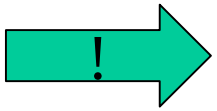


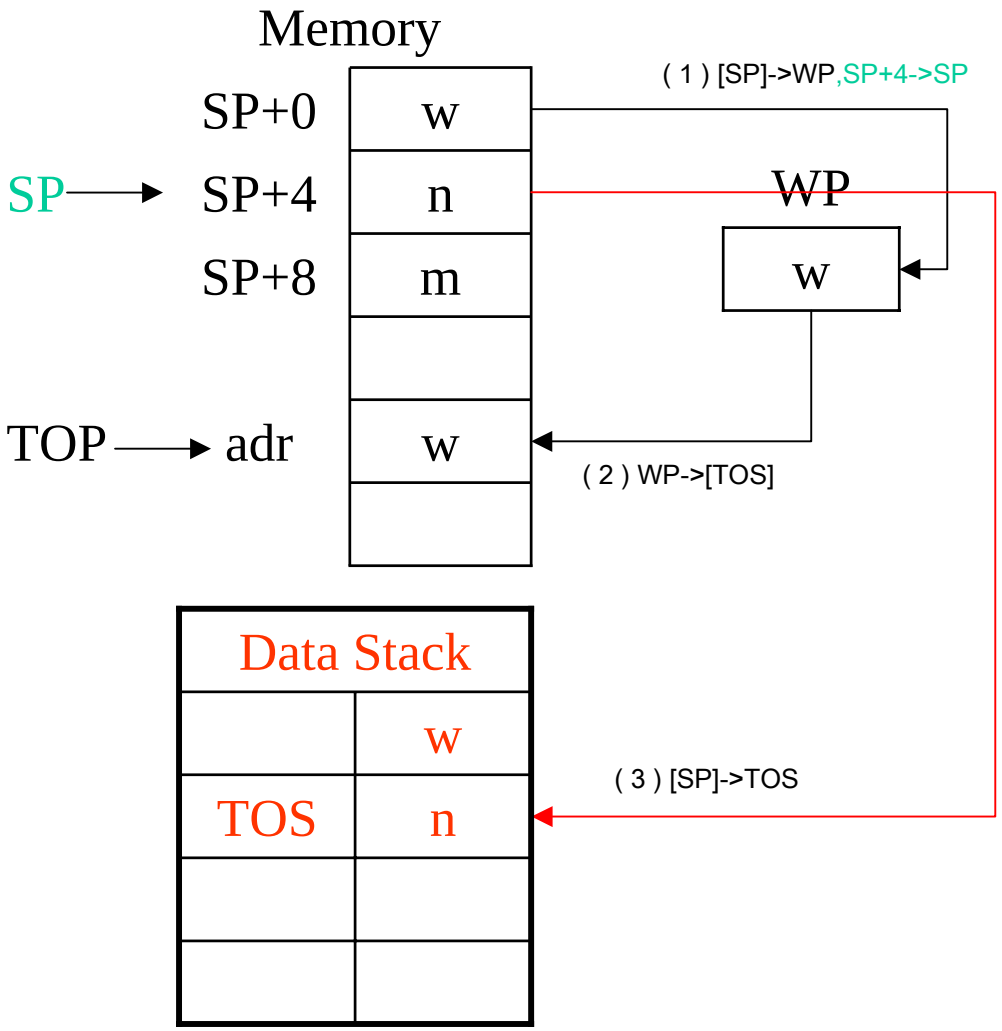


! (w adr --)

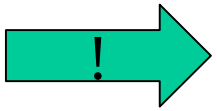
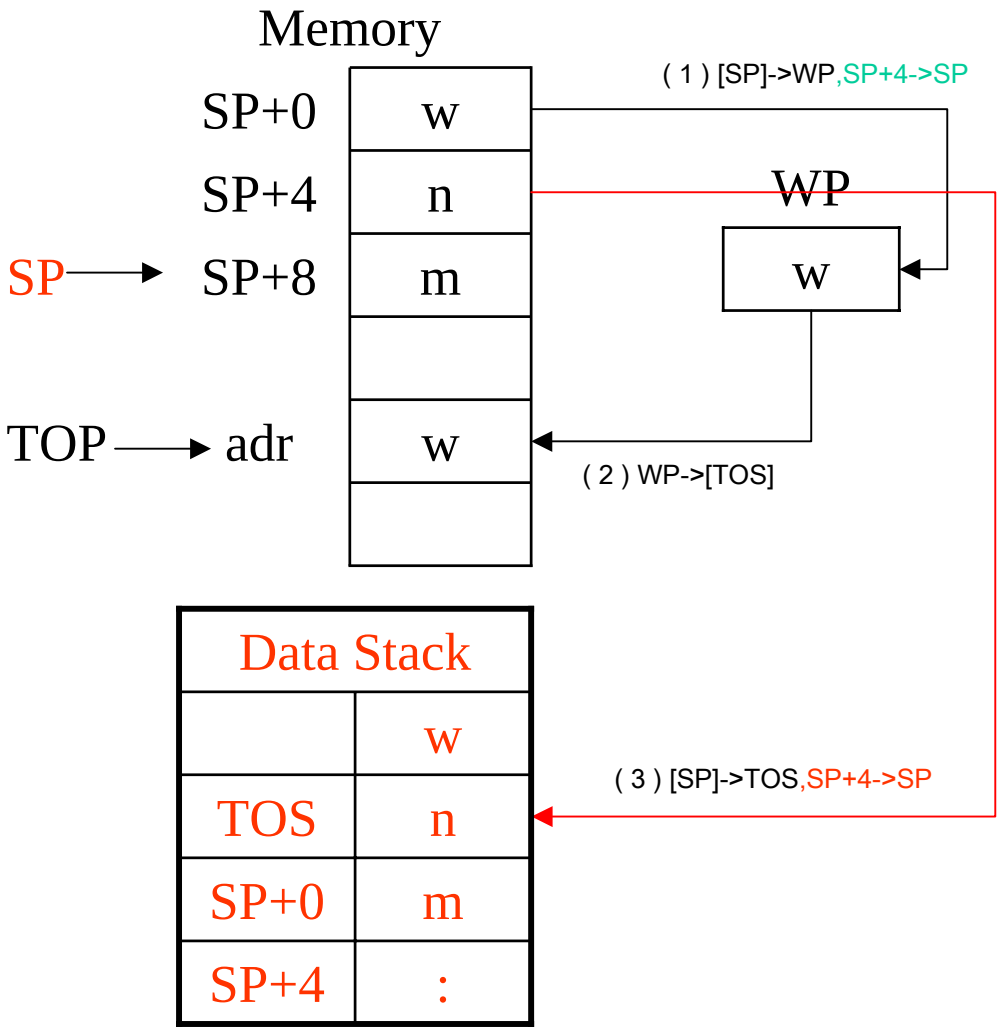
```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	:





! (w adr --)

```
CODE ! ( w a -- ) \ Pop the data stack to memory.
 ( 1 ) [SP]->WP,SP+4->SP
 ( 2 )  WP->[TOS]
 ( 3 ) [SP]->TOS,SP+4->SP
  $NEXT \ NEST
  END-CODE \ ;CODE
```

Data Stack	
TOS	adr
SP+0	w
SP+4	n
SP+8	m
SP+12	: