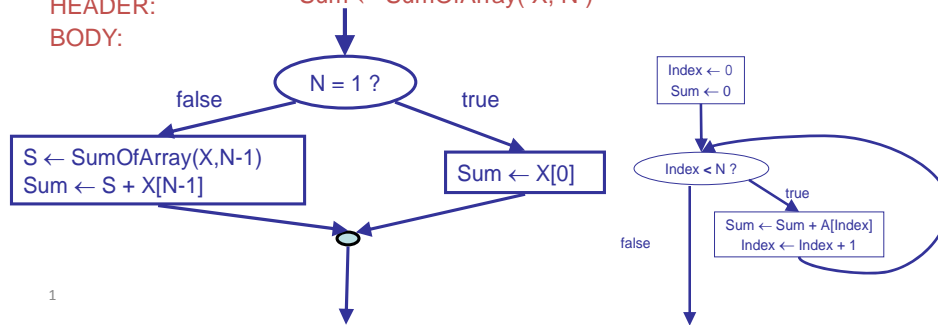


# Recursive Sum of Array

- Write a recursive algorithm to find the sum of the values in array positions 0...(N-1):

GIVENS: X (an array)  
 N (the length of the array)  
 S (sum of array of size N-1)  
 INTERMEDIATES: Sum (sum of array of size N)  
 RESULT:  $Sum \leftarrow SumOfArray(X, N)$   
 HEADER:  
 BODY:

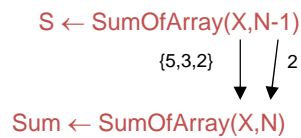


1

## Trace for this value of X: (p.1)



statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	3	?	?
1. test: N = 1? <b>false</b>				
2. S ← SumOfArray(X, N-1) see page 2				
3. Sum ← S + X[N-1]				



2

## Trace page 2

statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	2	?	?
1. test: N = 1? false				
2. S ← SumOfArray(X,N-1) see page 3				
3. Sum ← S + X[N-1]				

$S \leftarrow \text{SumOfArray}(X, N-1)$   
 $\{5, 3, 2\}$     ↓    ↓    1  
 $\text{Sum} \leftarrow \text{SumOfArray}(X, N)$

3

## Trace, page 3

statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	1	?	?
1. test: N = 1? true				
4. Sum ← X[0]				

4

## Trace, page 3

statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	1	?	?
1. test: $N = 1?$ true				
4. $\text{Sum} \leftarrow X[0]$				5

5

## Trace page 2

statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	2	?	?
1. test: $N = 1?$ false				
2. $S \leftarrow \text{SumOfArray}(X, N-1)$ see page 3				
3. $\text{Sum} \leftarrow S + X[N-1]$				

$S \leftarrow \text{SumOfArray}(X, N-1)$   
 $\uparrow$                        $\downarrow$   $\downarrow$   
 5                      {5,3,2}                      1  
 $\text{Sum} \leftarrow \text{SumOfArray}(X, N)$

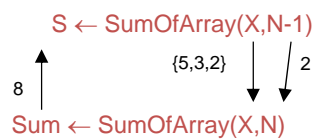
6



## Trace page 1



statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	3	?	?
1. test: $N = 1$ ? false				
2. $S \leftarrow \text{SumOfArray}(X, N-1)$ see page 2				
3. $\text{Sum} \leftarrow S + X[N-1]$				

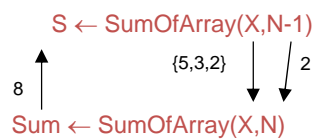


9

## Trace page 1



statement	X	N	S	Sum
initial values	{ 5, 3, 2 }	3	?	?
1. test: $N = 1$ ? false				
2. $S \leftarrow \text{SumOfArray}(X, N-1)$ see page 2			8	
3. $\text{Sum} \leftarrow S + X[N-1]$				



10



## Find $X^N$ (version 1)



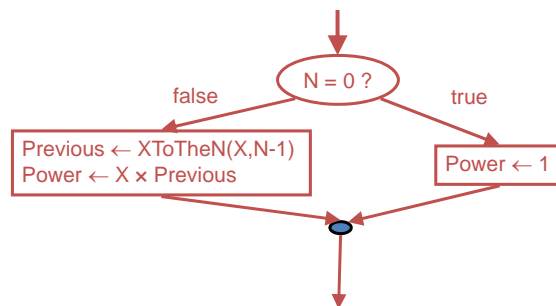
GIVENS: X (base of exponentiation)  
 N (power to which X is raised)

INTERMEDIATES: Previous (result of previous recursion)

RESULT: Power

HEADER: Power  $\leftarrow$  XToTheN (X,N)

BODY:



13

## Find $X^N$ (version 2: more efficient)



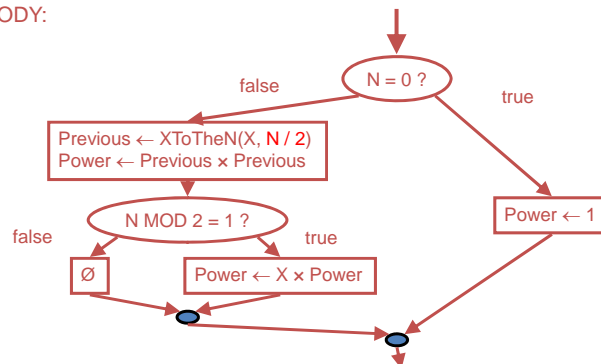
GIVENS: X (base of exponentiation)  
 N (power to which X is raised)

INTERMEDIATES: Previous (result of previous recursion)

RESULT: Power

HEADER: Power  $\leftarrow$  XToTheN (X,N)

BODY:



14

## Recursive selection sort

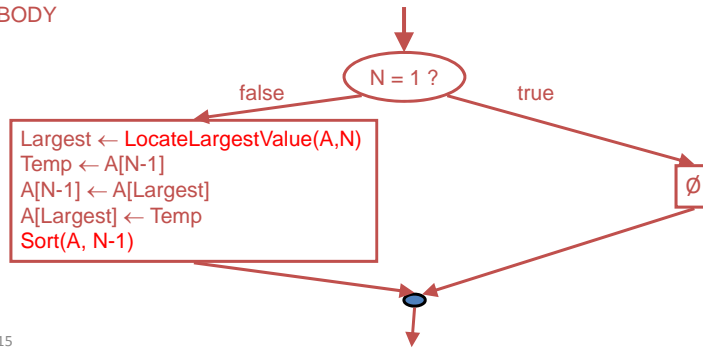
GIVENS: A (an array to sort)  
 N (number of elements in array)

INTERMEDIATES: Largest (position of largest value in array)  
 Temp (used for swapping)

RESULTS: (none)

HEADER: Sort(A, N)

BODY:



15

## Recursive location of largest value

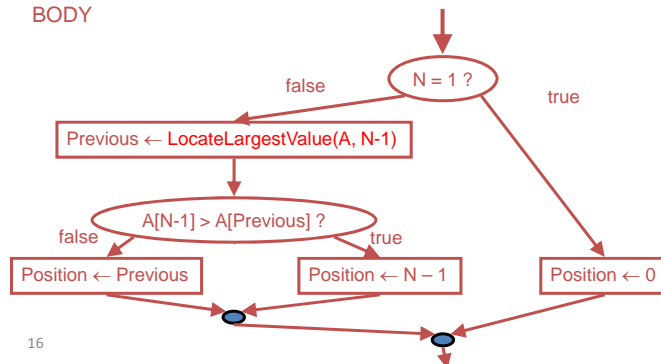
GIVENS: A (an array to sort)  
 N (number of elements in array)

INTERMEDIATES: Previous (position of largest value in smaller array)

RESULTS: Position (index of largest value in array)

HEADER: Position ← LocateLargestValue( A, N )

BODY:



16