

ARRAYS()

```
Create an array  $A[1 \dots n]$   
Create an array  $B[1 \dots n] \leftarrow [0 \dots 0]$   
Create an array  $C[1 \dots n] \leftarrow [1 \dots n]$   
Create an array  $D[1 \dots n] \leftarrow B[1 \dots n]$   
 $E[1 \dots n]$   
 $F[1 \dots n]$ 
```

STRUCTURES()

```
Create a dynamic array  $A \leftarrow []$   
Create a 2-D matrix  $B[1 \dots b][1 \dots d]$   
Create a SinglyLinkedList  $C$   
Create a CircularSinglyLinkedList  $D$   
Create a DoublyLinkedList  $E$   
Create a stack  $F$   
Create a queue  $G$   
Create a deque  $H$   
Create a BalancedSearchTree  $I$   
foreach child child of  $I.root$  node do  
|   print spam  
Create a hash set  $J$   
Create a hash map  $K$   
 $K.Add(\langle k, v \rangle)$   
Create a min-heap  $L$   
Create a max-heap  $M$ 
```

```

Foo(a, b, MyArr[1...n])
(1 + (2 / 3) × 4) − ⌊56 / 7⌋ % 8
(1 + 2) / (3 × 4) − ⌊56 / (7 % 8)⌋

while false do
|   x ← a; y ← b

if 1 + 1 = 0 then
|   a.bar()
else if not (MyArr[0] and (b.member or c)) then
|   BAZ()
else
|   return (a) ? ⌊a / b⌋ : null

for i ← 1 to 10 do
|   for j ← 1 to 5 do
|       continue
for i ← 0 to 10 increment 3 do
|   for j ← 10 down to 0 do
|       for k ← 10 down to 0 decrement 2 do
|           break

return BAR(A[1...(n − 1)])

```