Presentation 24

Coroutines

Announcements for This Lecture

Assignment 7

- We heard the complaints
  - Task 1 too hard for C-level
  - Task 2 just right for B/B+
- Relaxing **no code rule**
  - Can show for **Task 1 only**
  - Will guide you through it
- Due **December 21**
  - No extensions or lates
  - We have to grade it!

Lesson Videos

- **ALL** all are now posted
  - **Lesson 29** for today
  - **Lesson 30** is the last
- Will **not** cover Lesson 30
  - Not relevant for A7
  - Really if going to 2110
  - Last class = Office Hours
- Special lecture **December 10**
  - The "advising" lecture

More Announcements

Labs

- Lab today is **optional**
  - Not required at all
  - But can do for **extra credit**
  - Need 19/23 labs finished
- **No more official labs**
  - Office hours for A7 help
  - Easier than waiting in queue
  - Follows new code rules

Surveys

- Will have a **survey for A7**
- Will also have an **exit survey**
  - Follow up to Survey 0
  - Should I keep the **videos**?
  - Would you rather had a **final**?
- Also the **course evaluation**
  - Comes from engineering
  - Completely *anonymous*
  - But we know if you submit

From Last Time: Chaining Generators

```
def sumfold(input):
    """
    Generates the sums of the numbers seen so far in input

    Example: sumfold([1,2,3]) generates the numbers 1, 3, and 6

    Parameter input: The input data to sum
    Precondition: input is a iterable of numbers (int or float)
    """
    pass
```

From Last Time: Chaining Generators

```
def sumfold(input):
    """
    Generates the sums of the numbers seen so far in input

    Example: sumfold([1,2,3]) generates the numbers 1, 3, and 6

    Parameter input: The input to sum
    Precondition: input is a iterable of numbers (int or float)
    """
    pass
```

For maximum **flexibility**

From Last Time: Chaining Generators

```
def filterdiv(input,n):
    """Generates all elements of input evenly divisble by n

    The elements are generated in the order they appear in input.

    Example: filterdiv([1,2,3,4],2) generates the numbers 2 and 4

    Parameter input: The input data to filter
    Precondition: input is a iterable of int

    Parameter n: The number to divide by
    Precondition: n is an int"""
    pass
```

Activity: Call Frame Time

Parent Function

```
    def sumfold(lst):
        """Returns list of sums"""
32      sum = []
33      g = pushsum(len(lst))
34      next(g)
35      for x in lst:
36          a = g.send(x)
37          sum.append(a)
38      return sum
```

Coroutine

```
    def pushsum(n):
        """Yields sum of all sent"""
16      sum = 0
17      for x in range(n):
18          val = (yield sum)
19          sum = sum+val
20      yield sum
```

Activity: Call Frame Time

Parent Function

```
def sumfold(
    """Returns
32  sum = []
33  g = pushsum(len(lst))
34  next(g)
35  for x in lst:
36      a = g.send(x)
37      sum.append(a)
38  return sum
```

Sometimes called the **driver**

Coroutine

```
def pushsum(n):
    """Yields sum of all sent"""
16  sum = 0
17  for x in range(n):
18      val = (yield sum)
19      sum = sum+val
20  yield sum
```

Activity: Call Frame Time

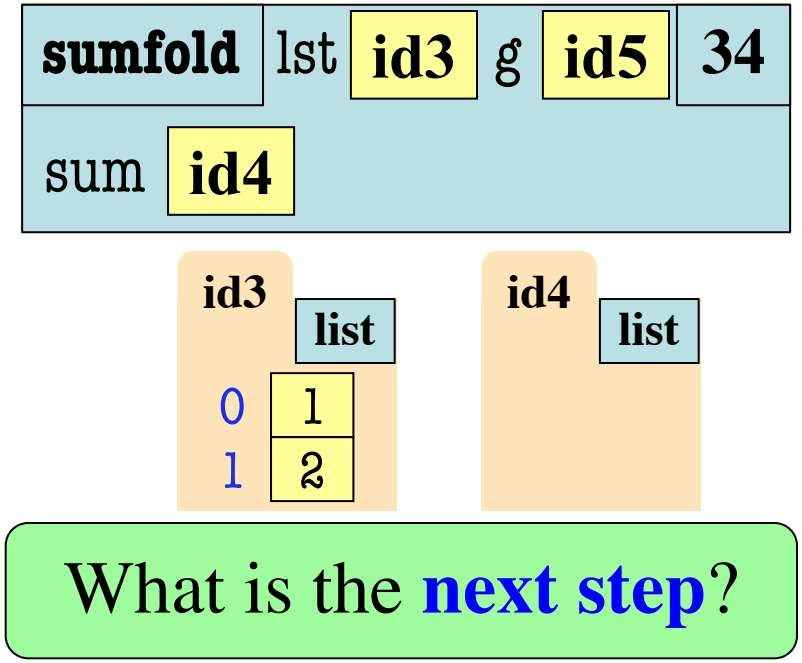
Parent Function

```
def sumfold(lst):
    """Returns list of sums"""
32  sum = []
33  g = pushsum(len(lst))
34  next(g)
35  for x in lst:
36      a = g.send(x)
37      sum.append(a)
38  return sum
```

Function Call

```
>>> x = sumfold([1,2])
```

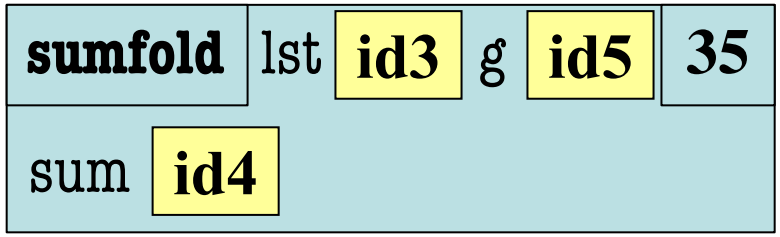
Assume we are here:



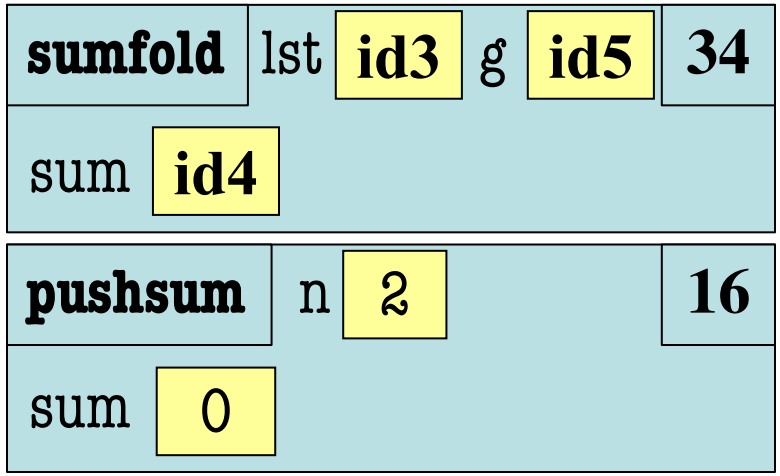
What is the **next step**?

Which One is Closest to Your Answer?

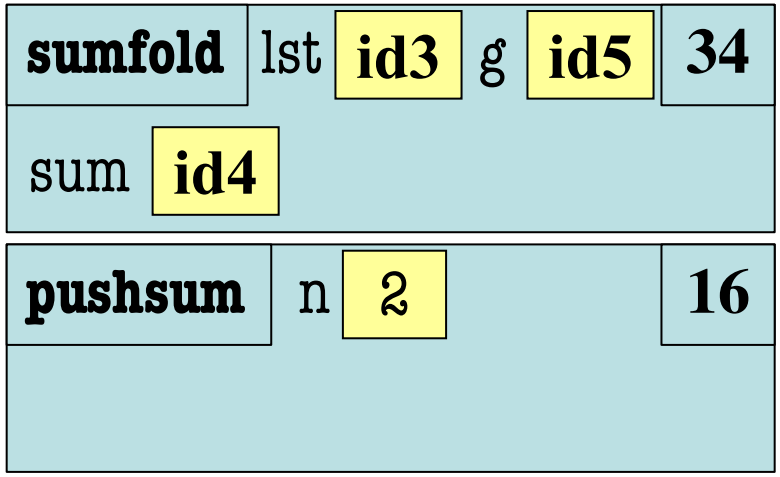
A:



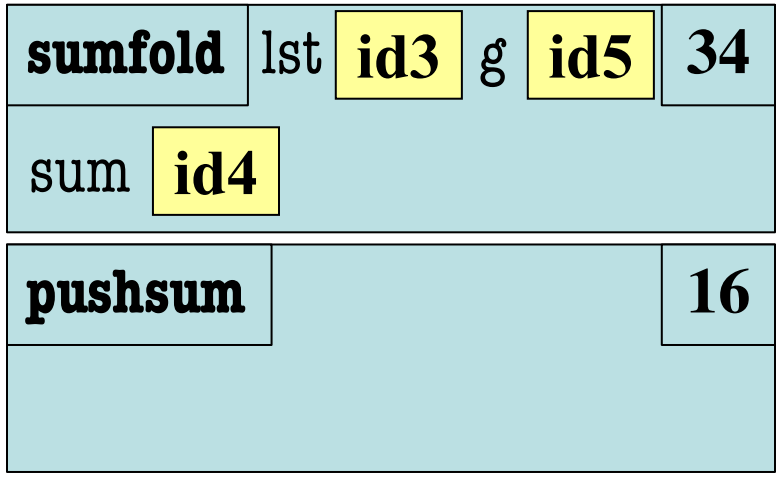
B:



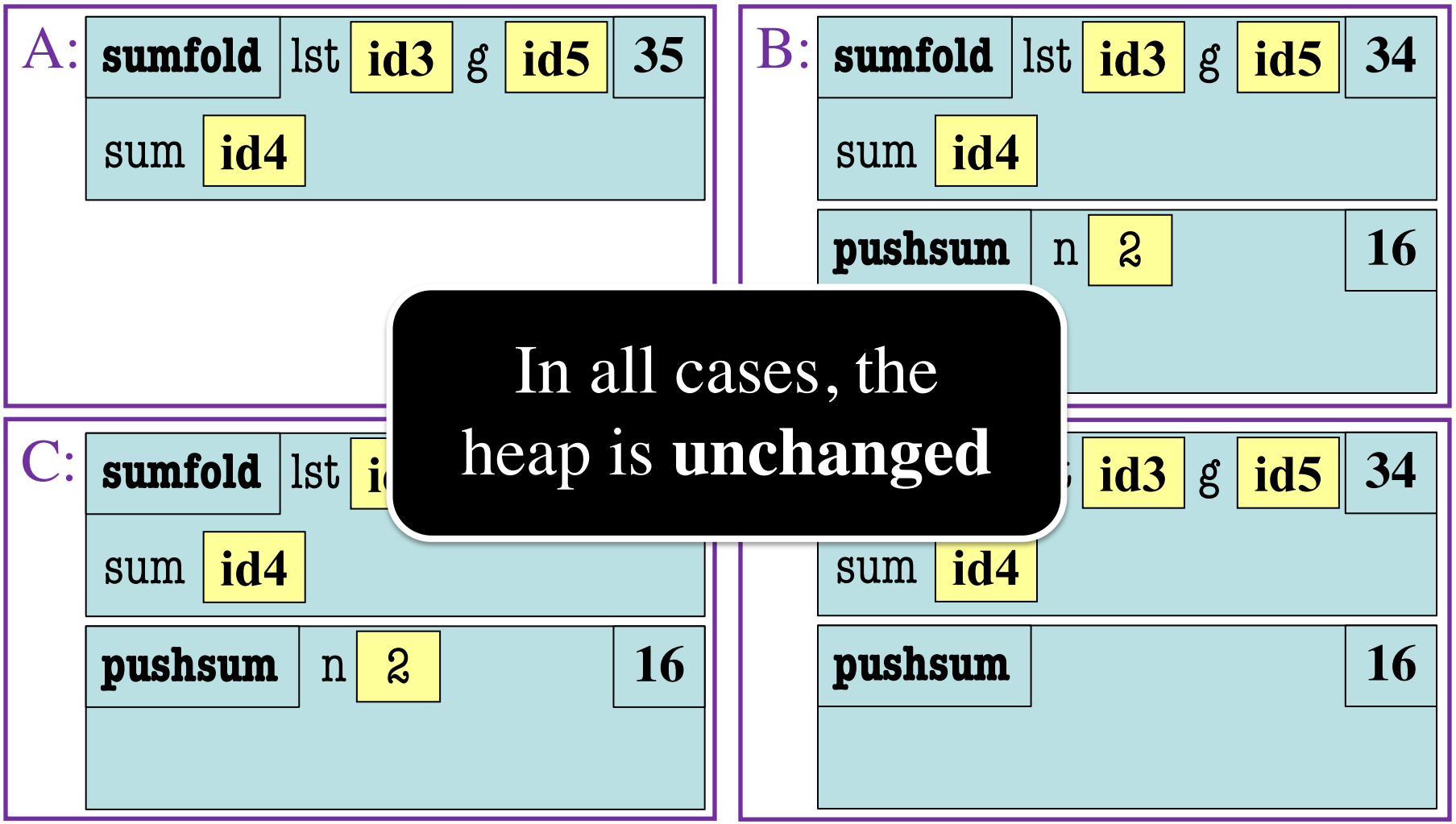
C:



D:



Which One is Closest to Your Answer?



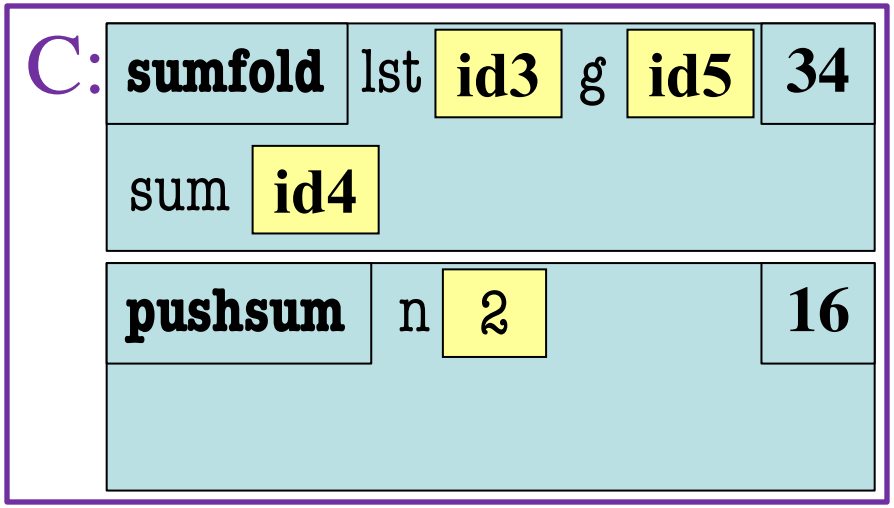
Activity: Call Frame Time

Coroutine

```
def pushsum(n):
    """Yields sum of all sent"""
16  sum = 0
17  for x in range(n):
18      val = (yield sum)
19      sum = sum+val
20  yield sum
```

Function Call

```
>>> x = sumfold([1,2])
```



What is the **next step**?

Which One is Closest to Your Answer?

A:

sumfold	lst	**id3**	g	**id5**	35
sum	**id4**				
pushsum	n	2			17
sum	0				

B:

sumfold	lst	**id3**	g	**id5**	34
sum	**id4**				
pushsum	n	2	x	1	17
sum	0				

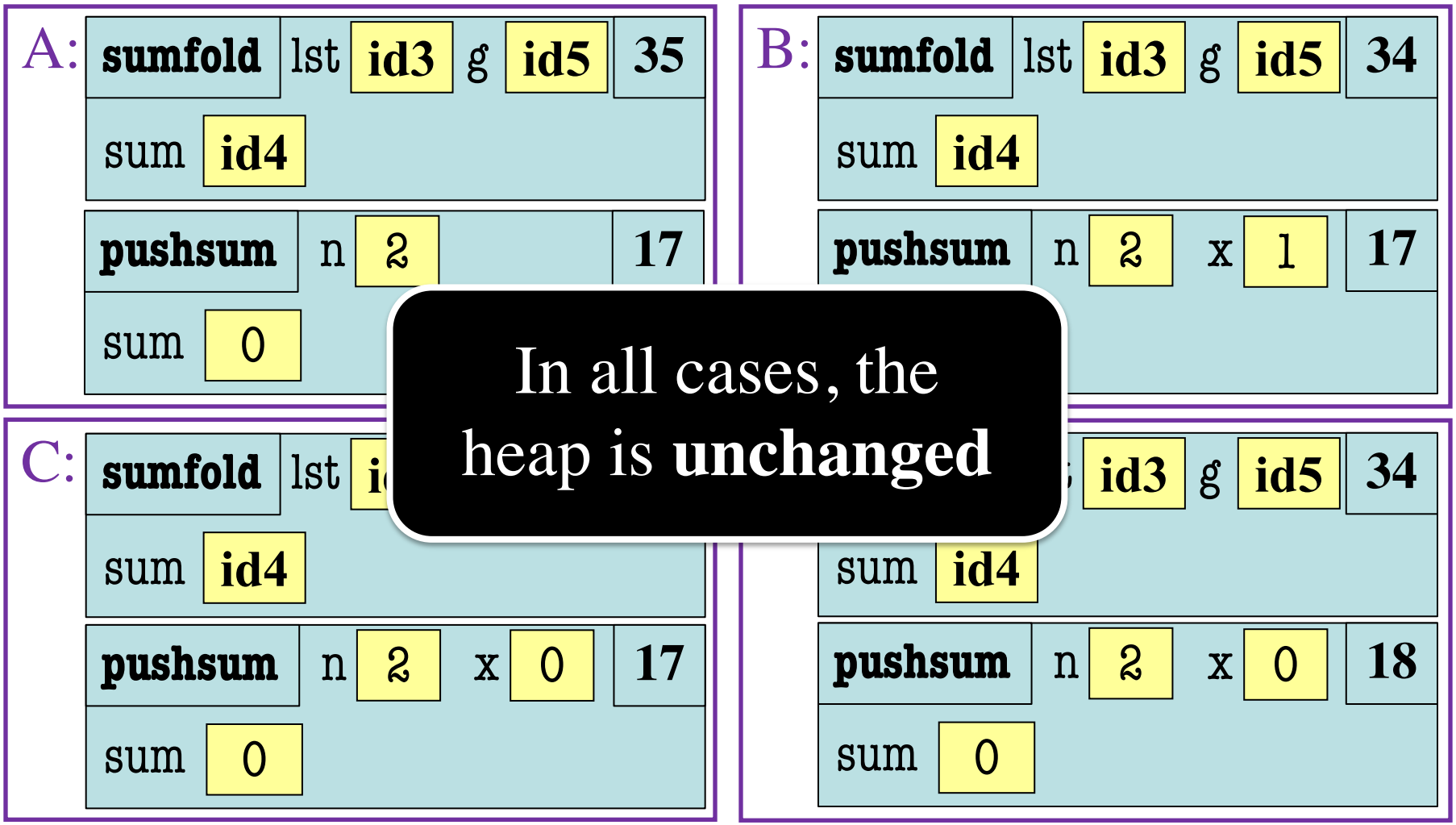
C:

sumfold	lst	**id3**	g	**id5**	34
sum	**id4**				
pushsum	n	2	x	0	17
sum	0				

D:

sumfold	lst	**id3**	g	**id5**	34
sum	**id4**				
pushsum	n	2	x	0	18
sum	0				

Which One is Closest to Your Answer?



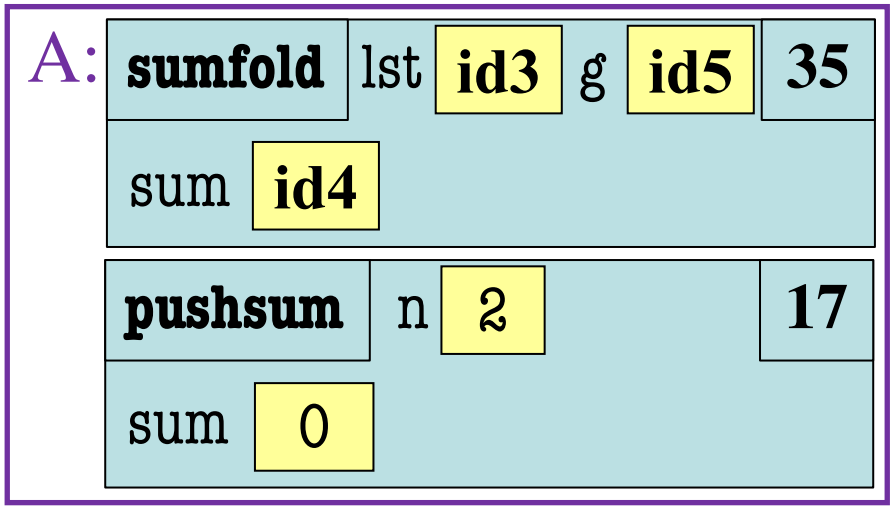
Activity: Call Frame Time

Coroutine

```
def pushsum(n):
    """Yields sum of all sent"""
16  sum = 0
17  for x in range(n):
18      val = (yield sum)
19      sum = sum+val
20  yield sum
```

Function Call

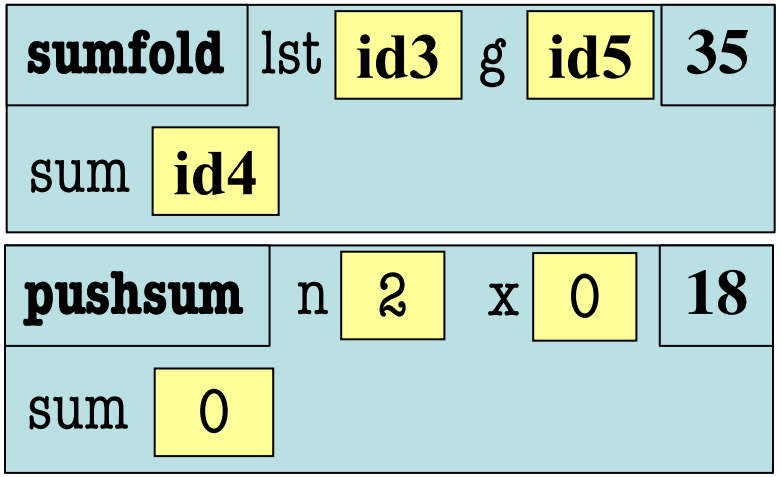
```
>>> x = sumfold([1,2])
```



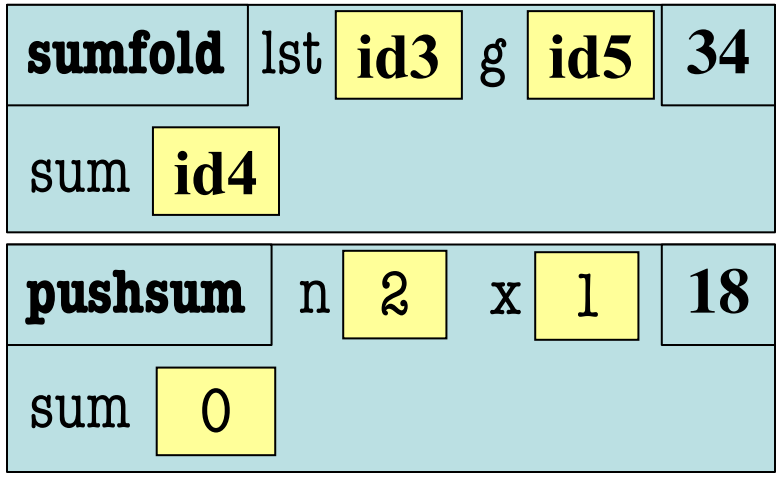
What is the **next step**?

Which One is Closest to Your Answer?

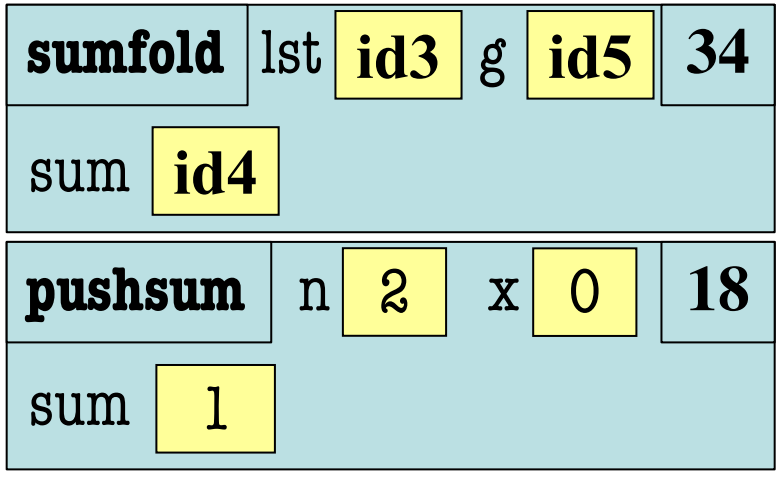
A:



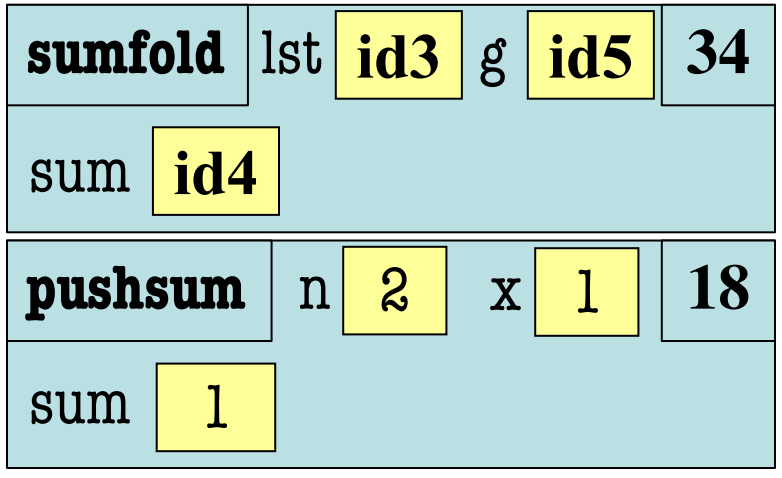
B:



C:



D:



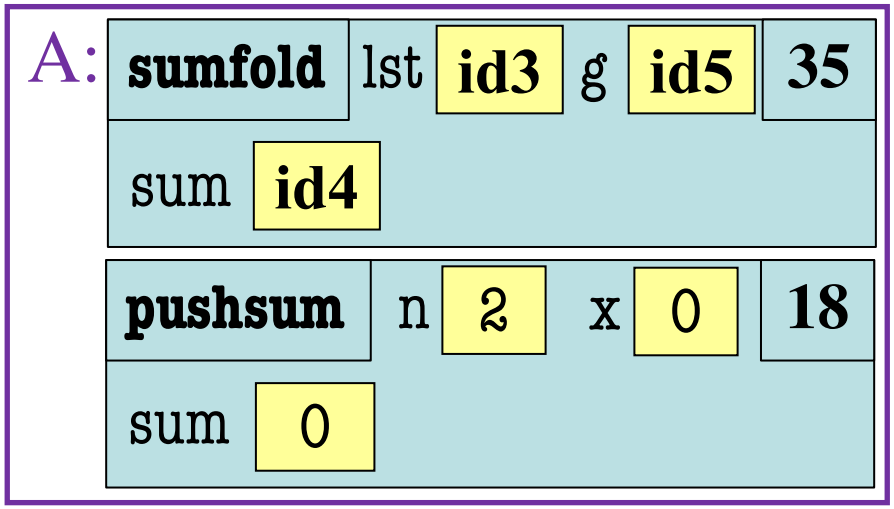
Activity: Call Frame Time

Coroutine

```
def pushsum(n):
    """Yields sum of all sent"""
16  sum = 0
17  for x in range(n):
18      val = (yield sum)
19      sum = sum+val
20  yield sum
```

Function Call

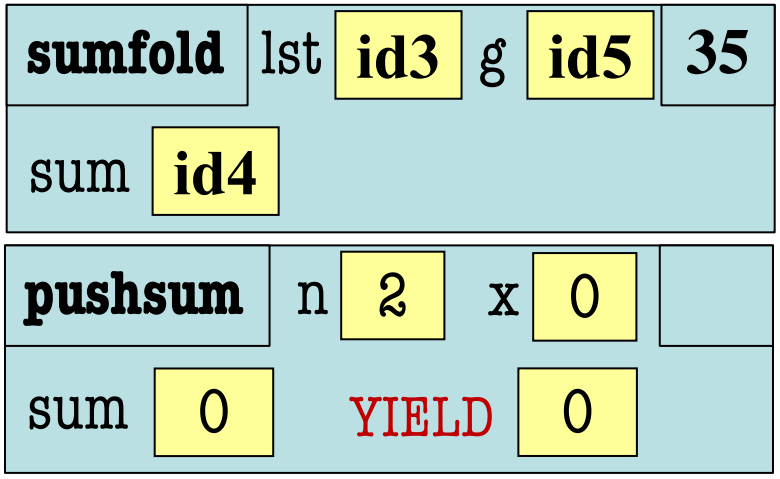
```
>>> x = sumfold([1,2])
```



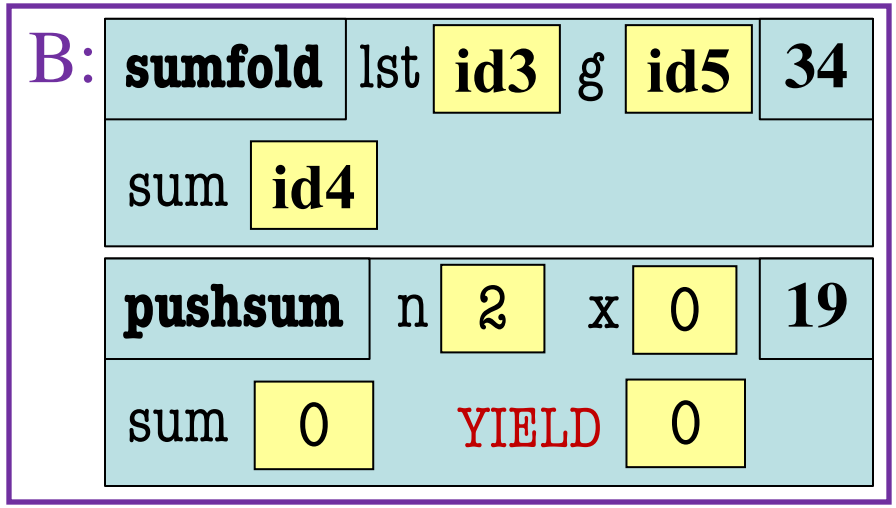
What is the **next step**?

Which One is Closest to Your Answer?

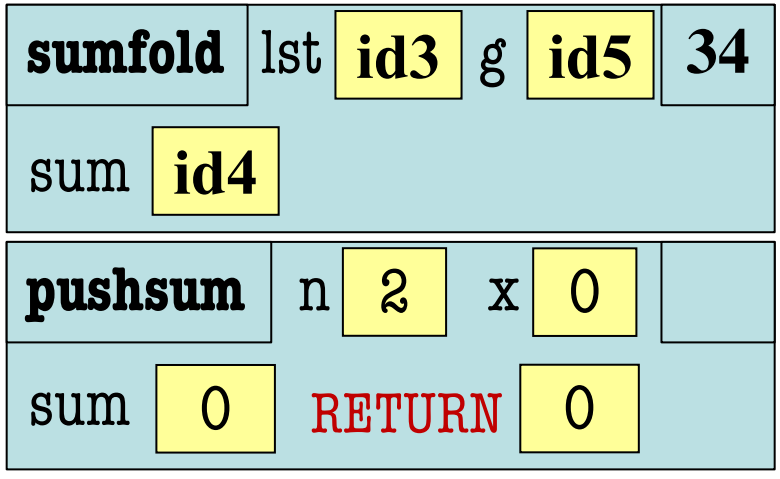
A:



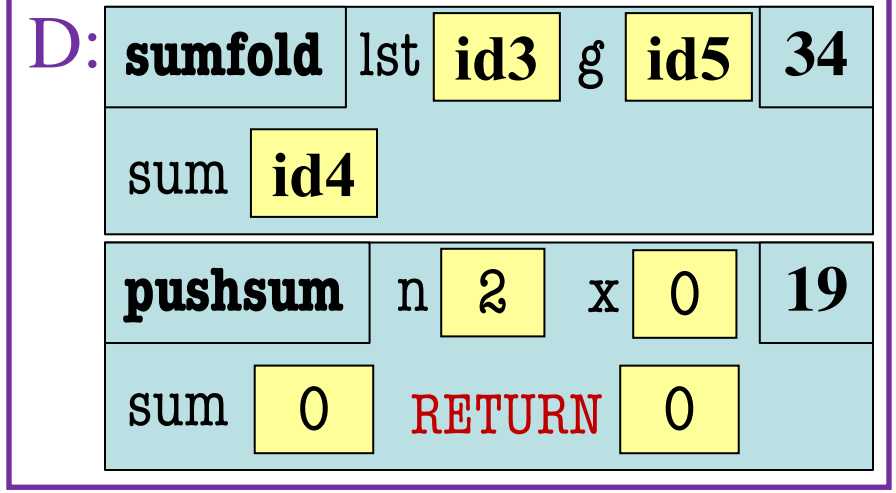
B:



C:



D:



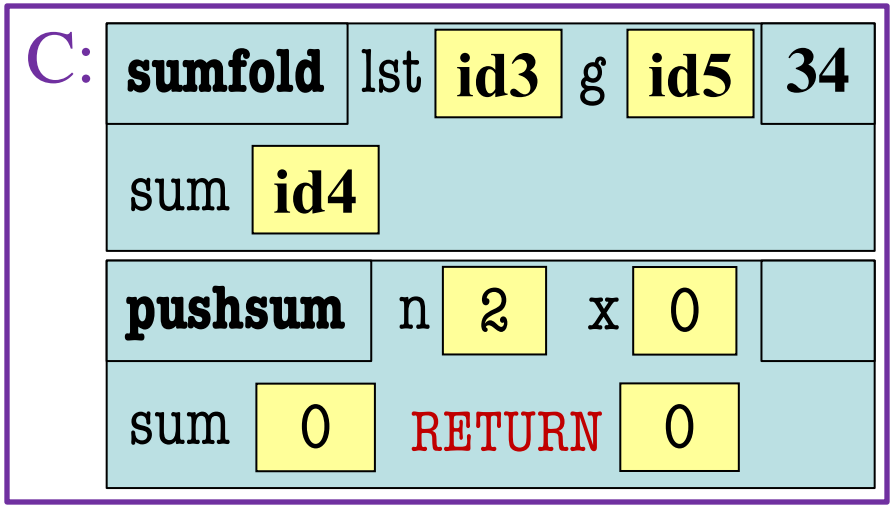
Activity: Call Frame Time

Parent Function

```
def sumfold(lst):
    """Returns list of sums"""
32  sum = []
33  g = pushsum(len(lst))
34  next(g)
35  for x in lst:
36      a = g.send(x)
37      sum.append(a)
38  return sum
```

Function Call

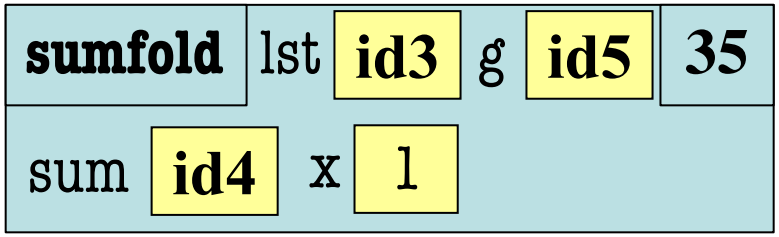
```
>>> x = sumfold([1,2])
```



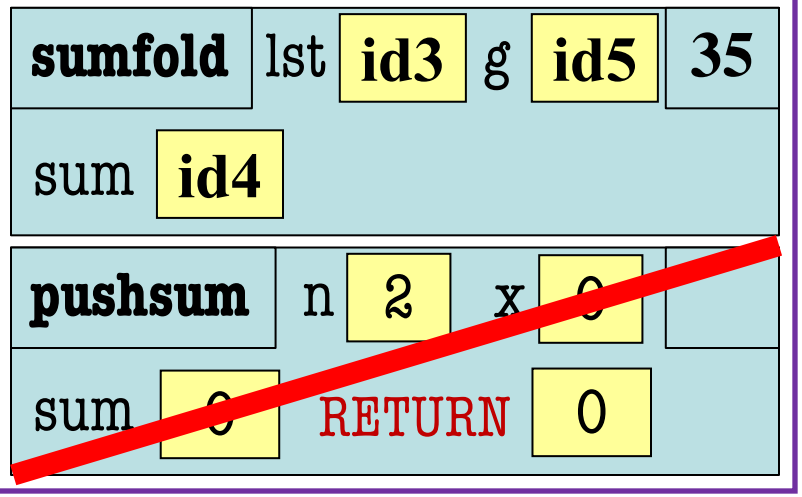
What is the **next step**?

Which One is Closest to Your Answer?

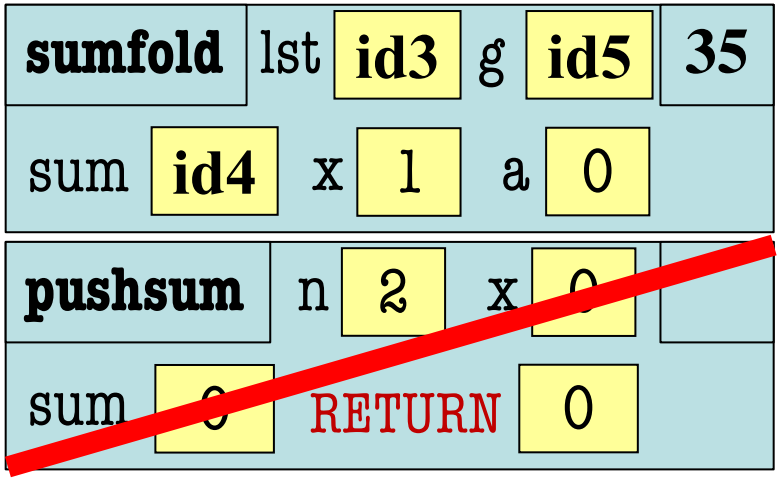
A:



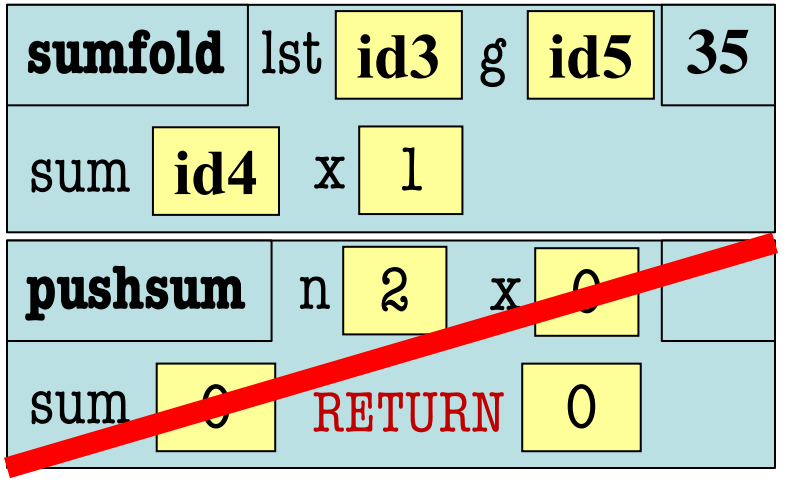
B:



C:



D:



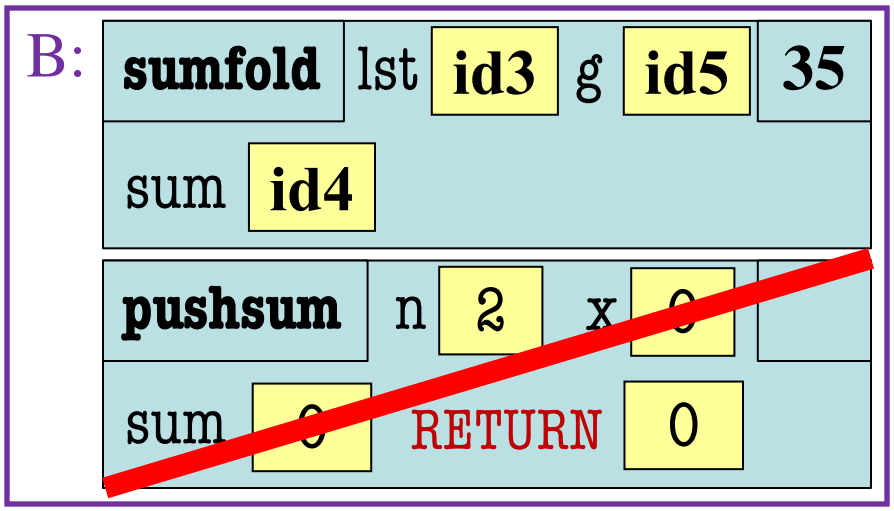
Activity: Call Frame Time

Parent Function

```
    def sumfold(lst):
        """Returns list of sums"""
32      sum = []
33      g = pushsum(len(lst))
34      next(g)
35      for x in lst:
36          a = g.send(x)
37          sum.append(a)
38      return sum
```

Function Call

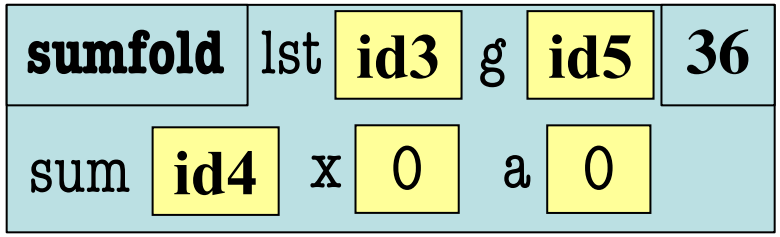
```
>>> x = sumfold([1,2])
```



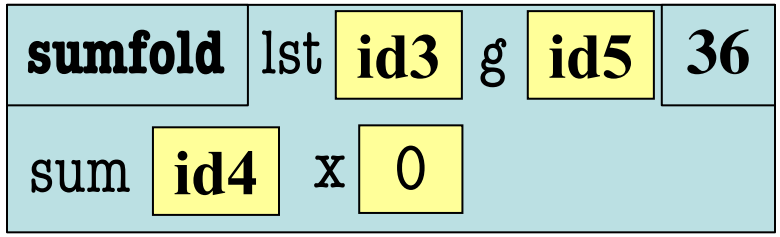
What is the **next step**?

Which One is Closest to Your Answer?

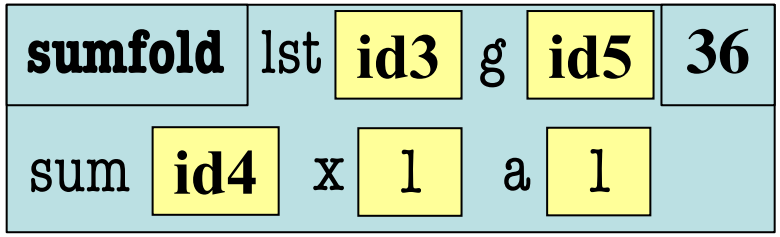
A:



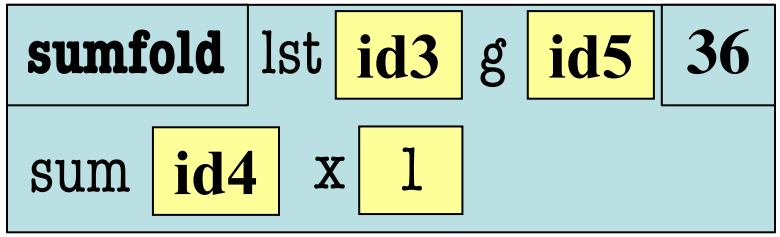
B:



C:



D:



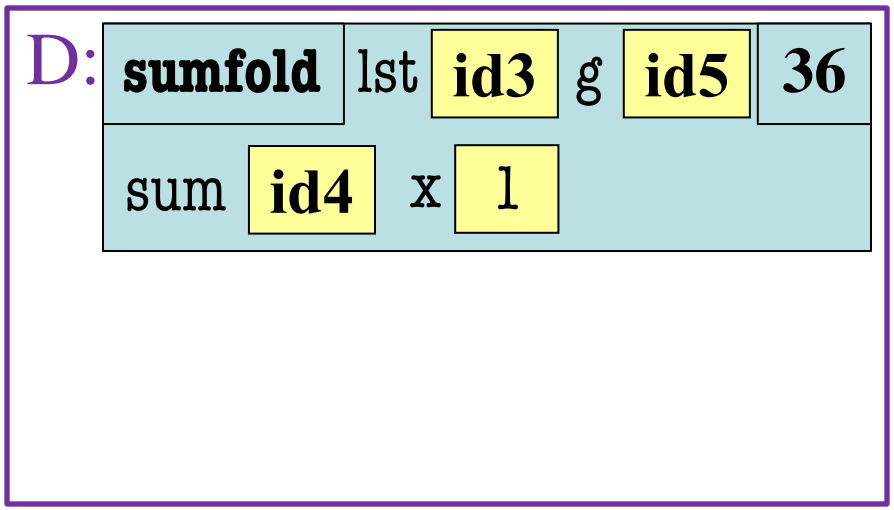
Activity: Call Frame Time

Parent Function

```
def sumfold(lst):
    """Returns list of sums"""
32  sum = []
33  g = pushsum(len(lst))
34  next(g)
35  for x in lst:
36      a = g.send(x)
37      sum.append(a)
38  return sum
```

Function Call

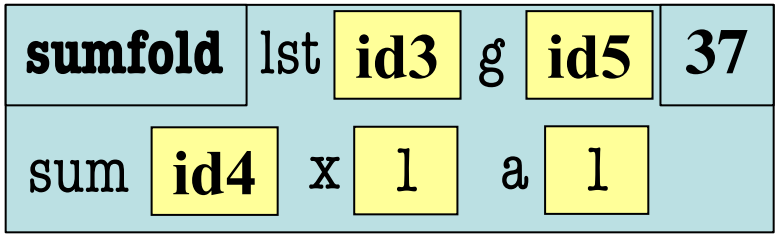
```
>>> x = sumfold([1,2])
```



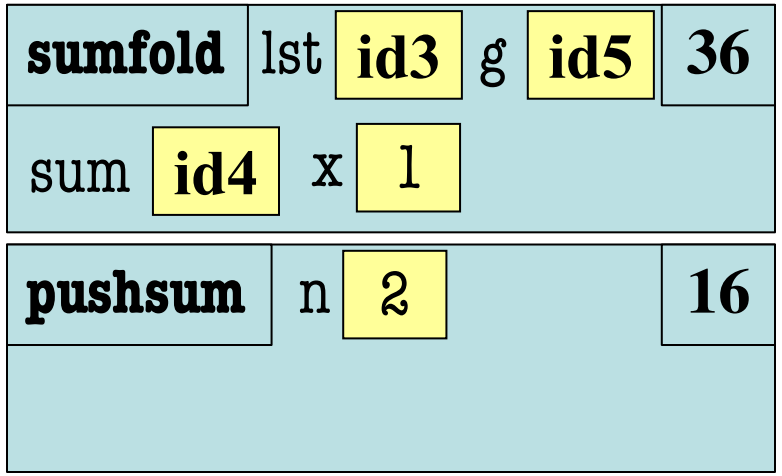
What is the **next step**?

Which One is Closest to Your Answer?

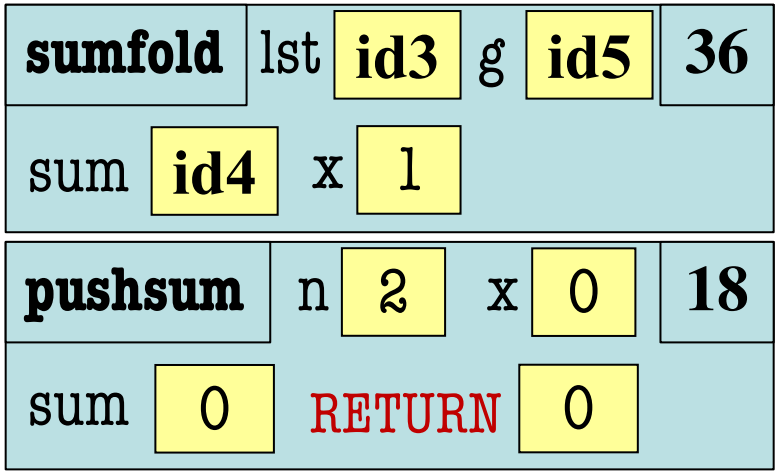
A:



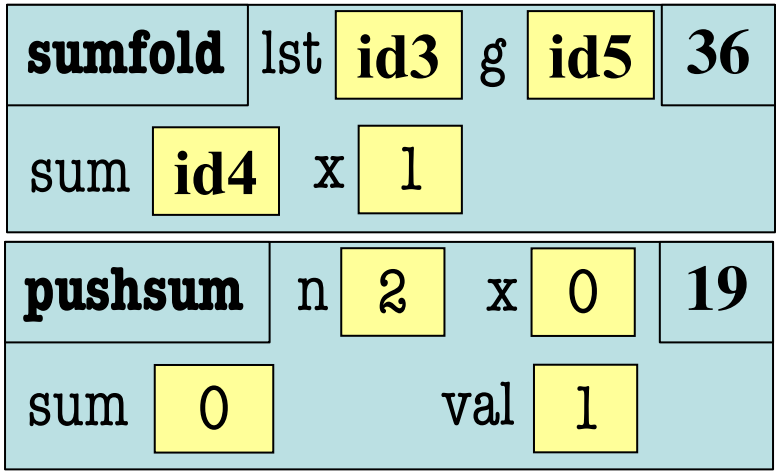
B:



C:



D:



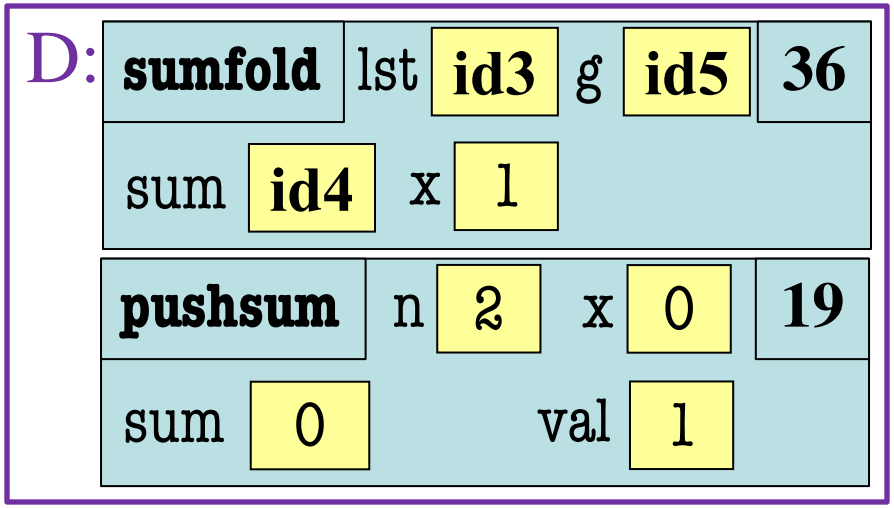
Activity: Call Frame Time

Coroutine

```
def pushsum(n):
    """Yields sum of all sent"""
16  sum = 0
17  for x in range(n):
18      val = (yield sum)
19      sum = sum+val
20  yield sum
```

Function Call

```
>>> x = sumfold([1,2])
```



Try the rest on your own

Demo: Writing a Coroutine

```
def chunkify(input):
    """Coroutine to break a list into chunks.

    Each call to send is the number of elements to chunk. At each call,
    it yields a new list of the size of the number of elements in send.

    If the size sent is not an int or is <= 0, it yields the empty list

    Parameter input: The data to process
    Precondition: input is an iterable"""
    pass
```

Demo: Writing a Coroutine

```
def chunkify(input):
    """Coroutine to break a list into chunks.

    Each call to send is the number of elements to chunk. At each call,
    it yields a new list of the size of the number of elements in send.

    If the size sent is not an int or is <= 0, it yields the empty list

    Parameter input: The data to proces
    Precondition: input is an iterable"""
    pass
```

Kind of like a **precondition**

Demo: Writing the Parent/Driver

```
def printit(lst,step):
    """Prints the contents of lst in groups of size step

    Each group is printed on a line by itself.

    Parameter lst: The list to process
    Precondition: lst is a list

    Parameter step:  The number of elements to print at a time
    Precondition: step is an int > 0"""
    pass
```

Questions?