Listing Permutations Done Recursively

Matt Bishop

MHI 289I, Fall Quarter 2025

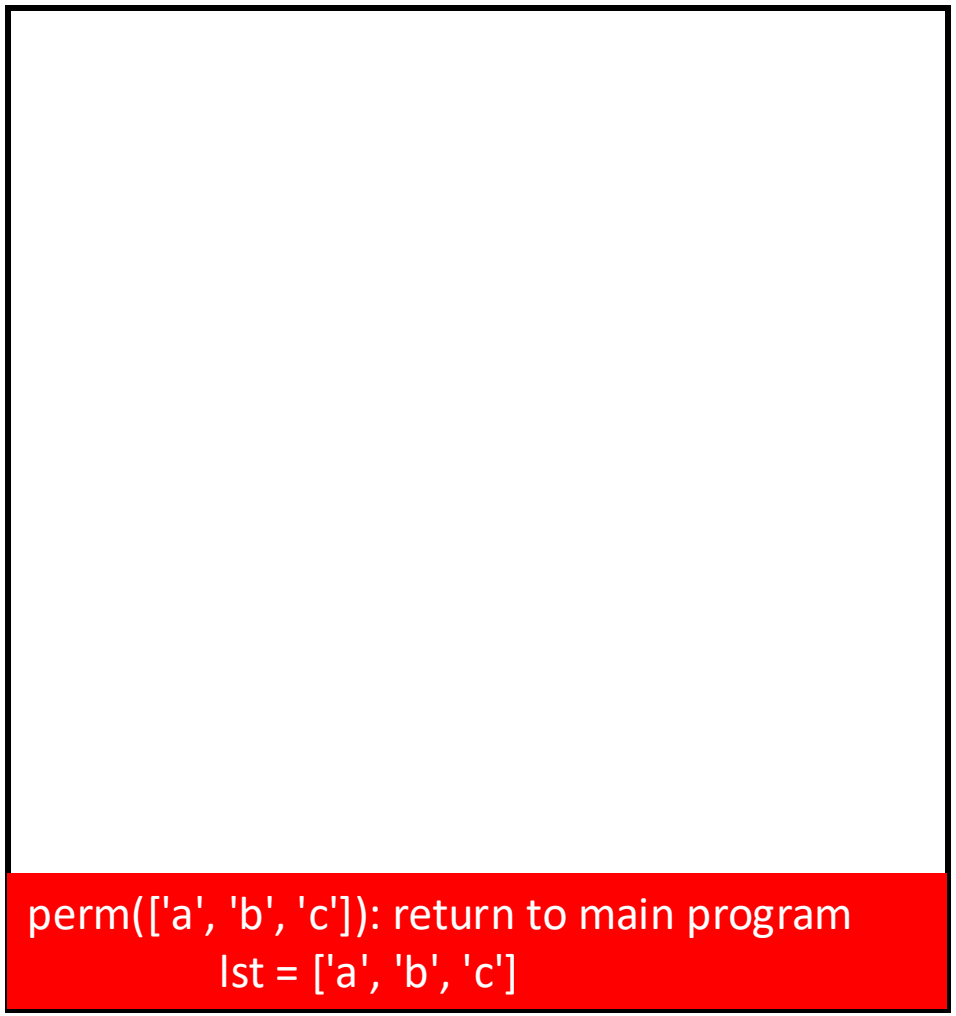
The Recursive n! Program

```
1:  def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:        remlst = lst[:i] + lst[i+1:]
7:        for p in perm(remlst):
8:              intlst.append([ lst[i] + p)
9:     return intlst
```

Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)

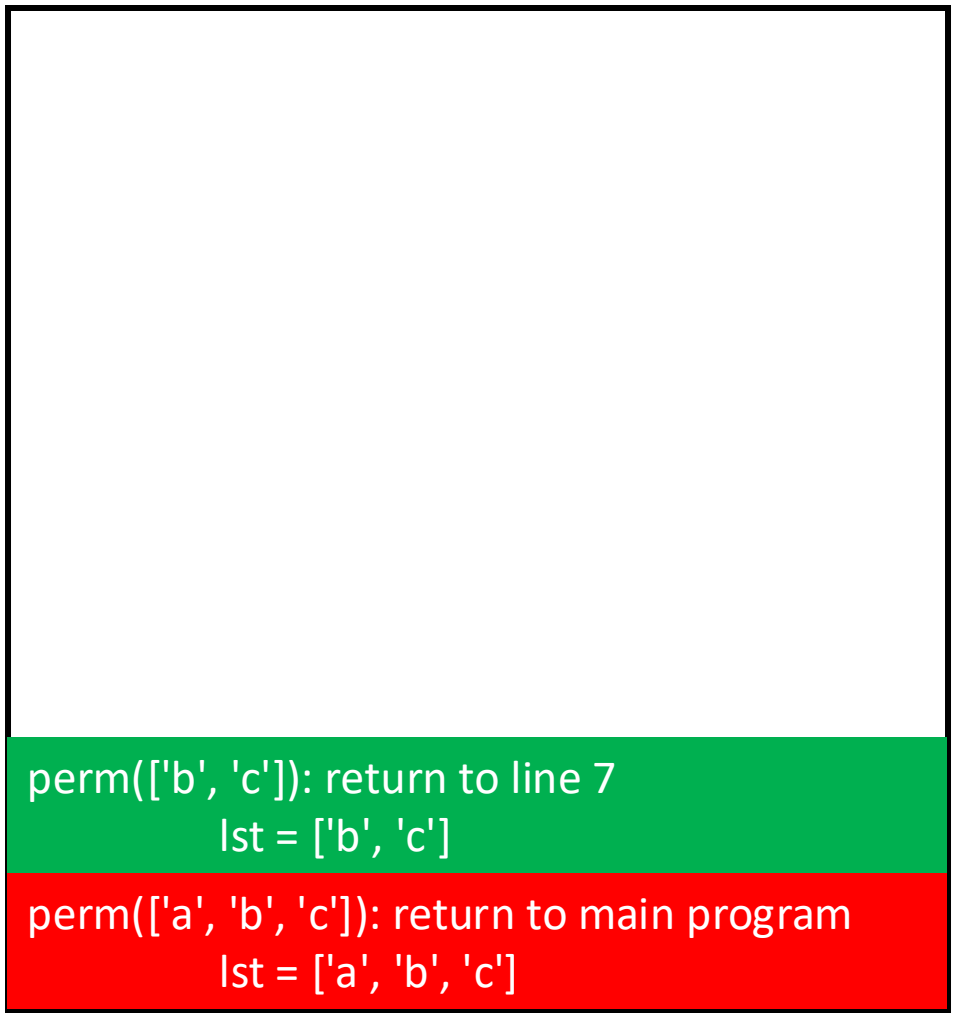
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)

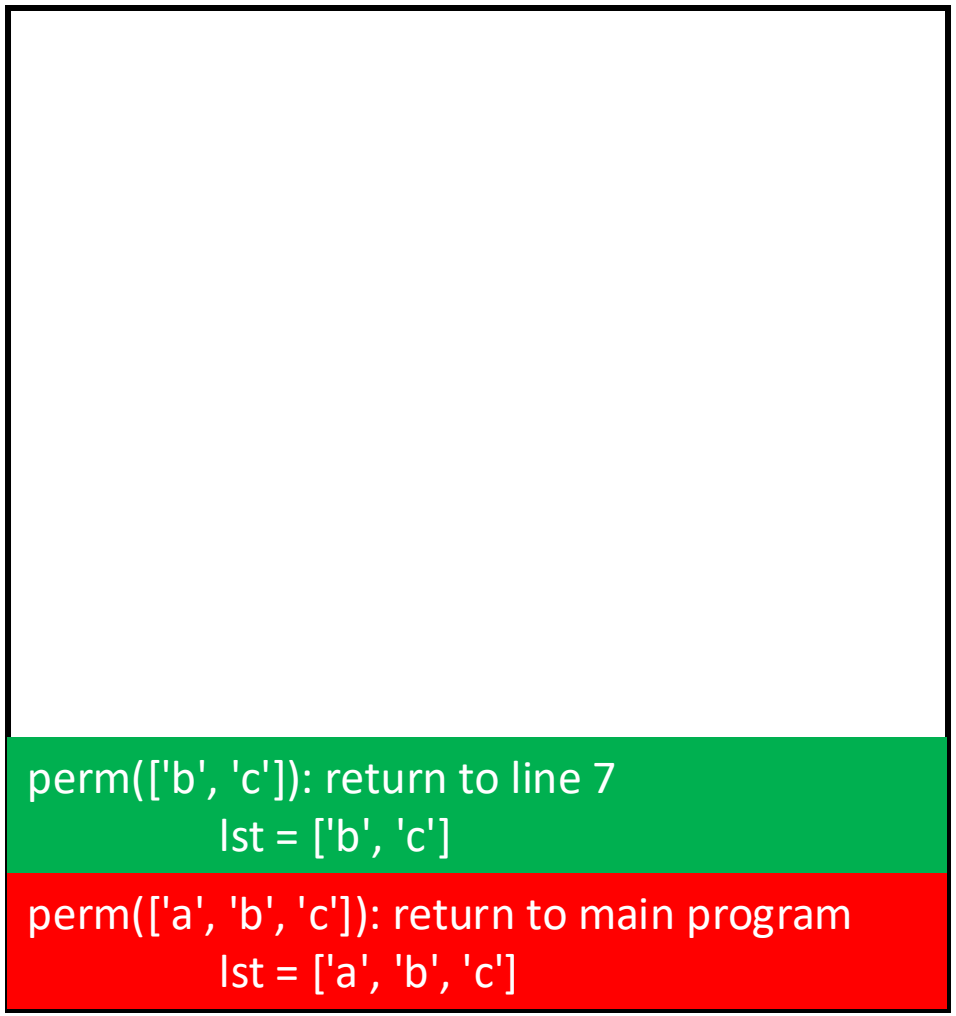
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c')] (line 7)
  - intlst ← [ ] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)

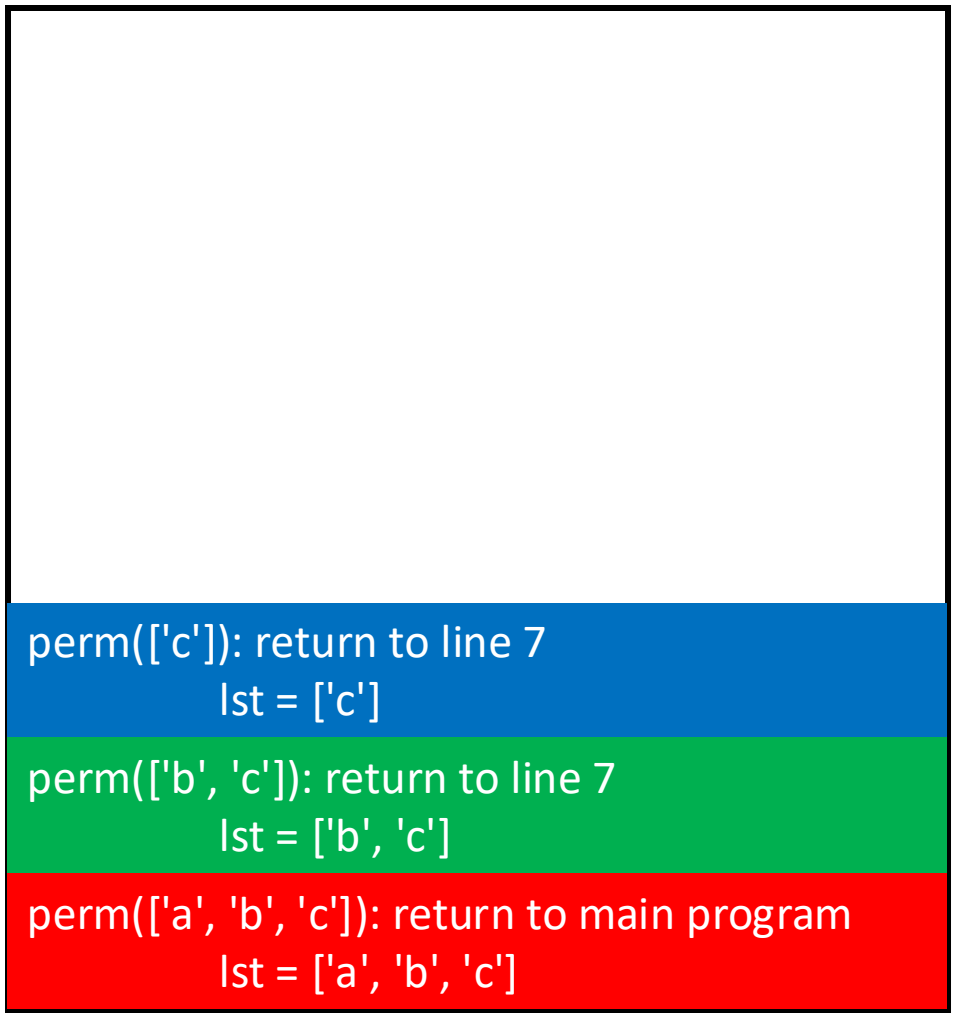
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c')] (line 7)
  - intlst ← [ ] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)
  - *call to* perm(['c']) (line 7)

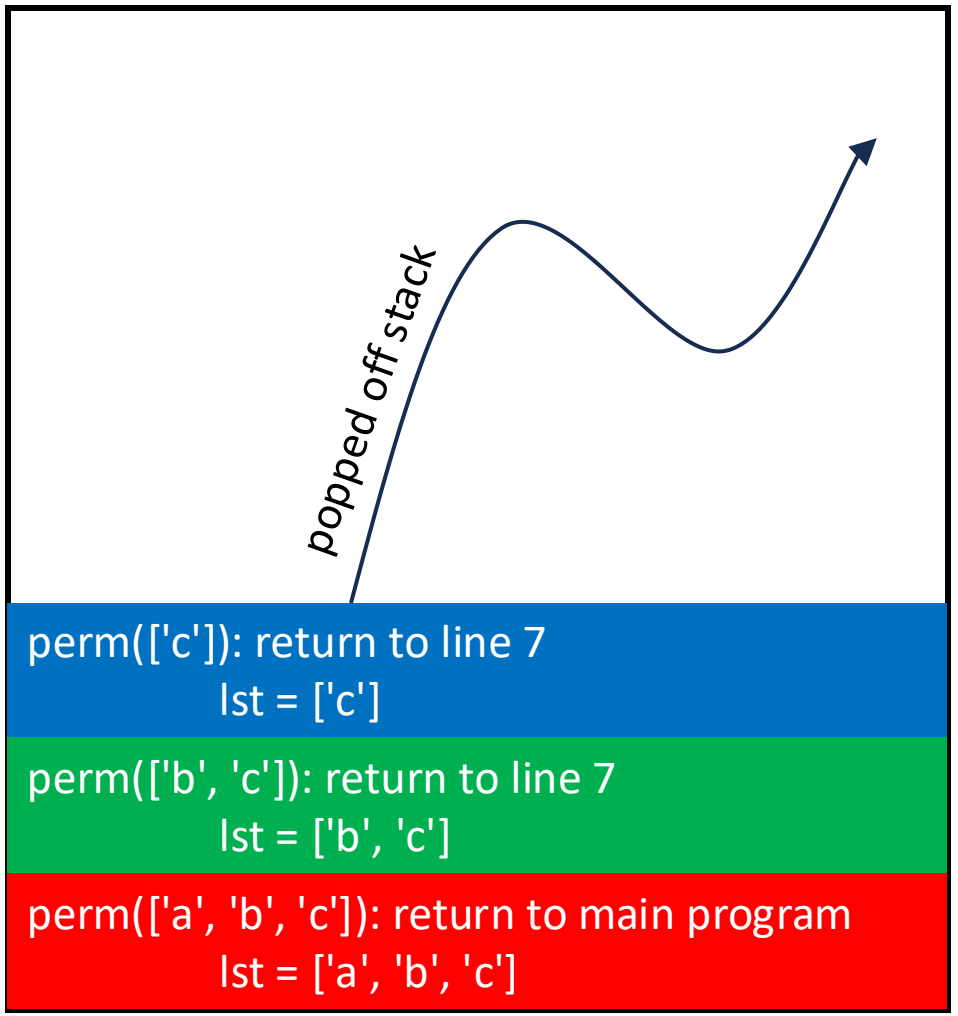
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)
  - intlst ← [ ] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)
  - *call to* perm(['c']) (line 7)
    - as len(lst) = 1, returns [['c']] (line 3)

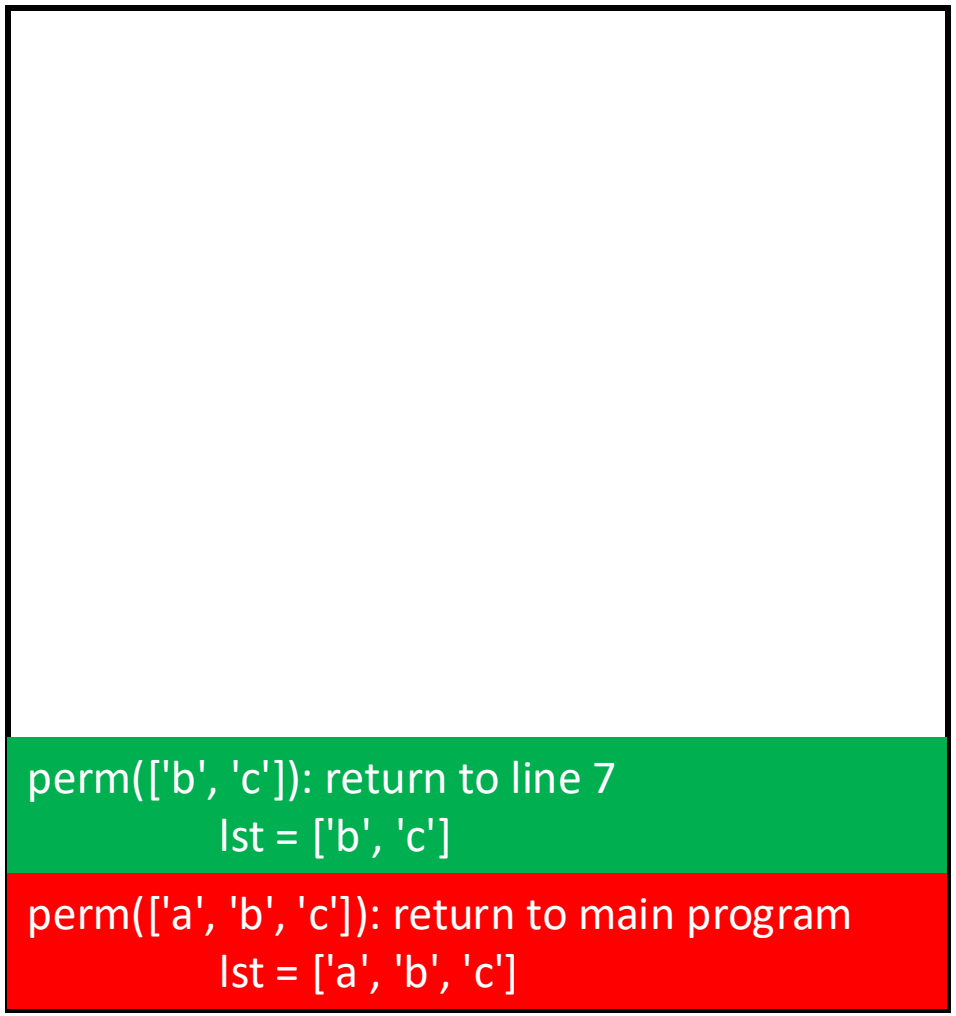
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)
  - p ← ['c'] (line 7)
  - intlst ← [['b'] + ['c']] *or* [['b', 'c']]

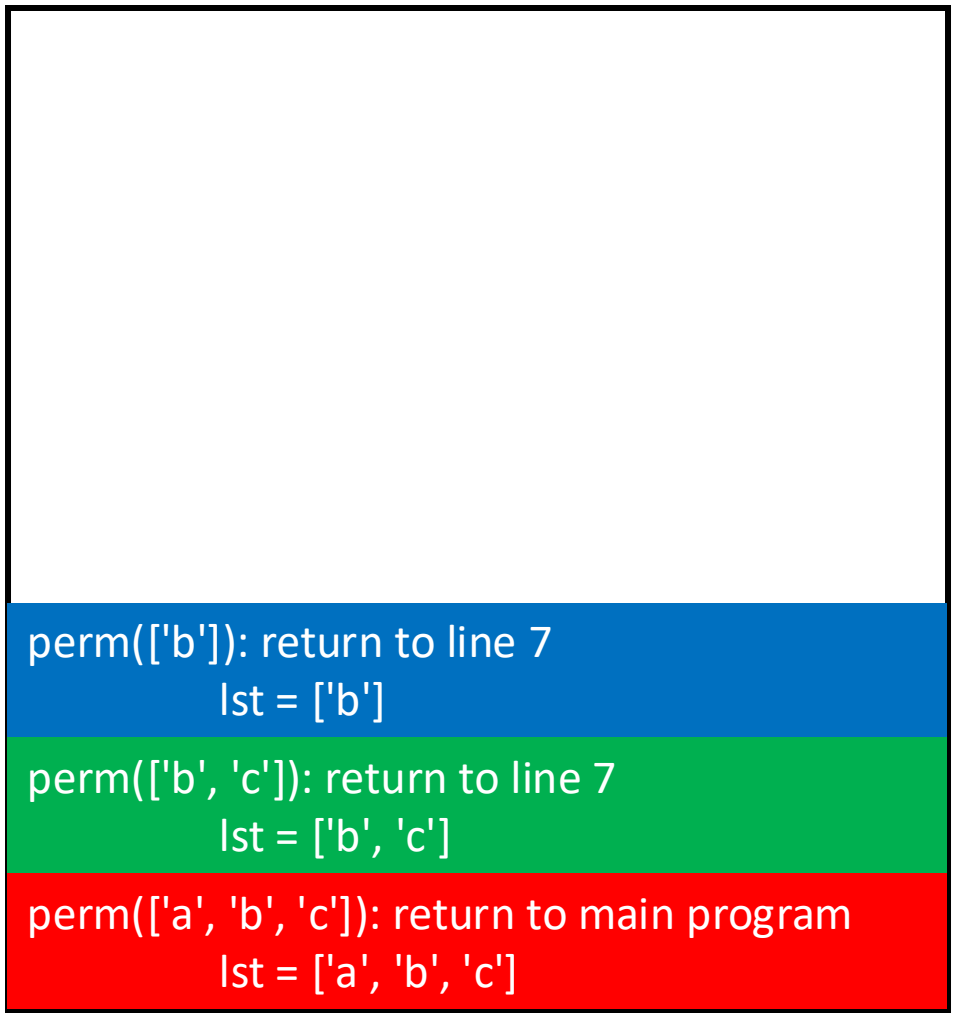
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)
  - intlst ← [['b', 'c']]
  - i ← 1 (line 5)
  - remlst ← ['b'] (line 6)
  - *call to* perm(['b']) (line 7)

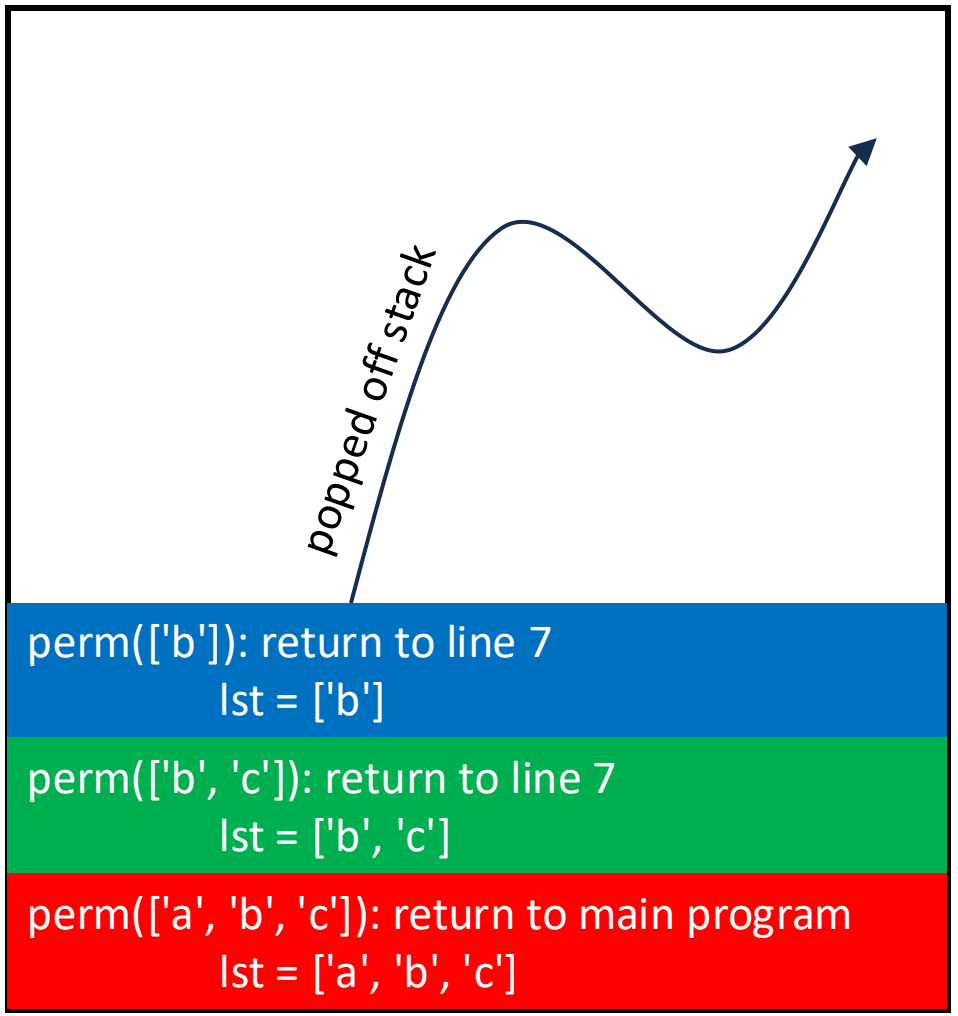
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)
  - intlst ← [['b', 'c']] (see slide 7)
  - i ← 1 (line 5)
  - remlst ← ['b'] (line 6)
  - *call to* perm(['b']) (line 7)
    - as len(lst) = 1, returns [['b']] (line 3)

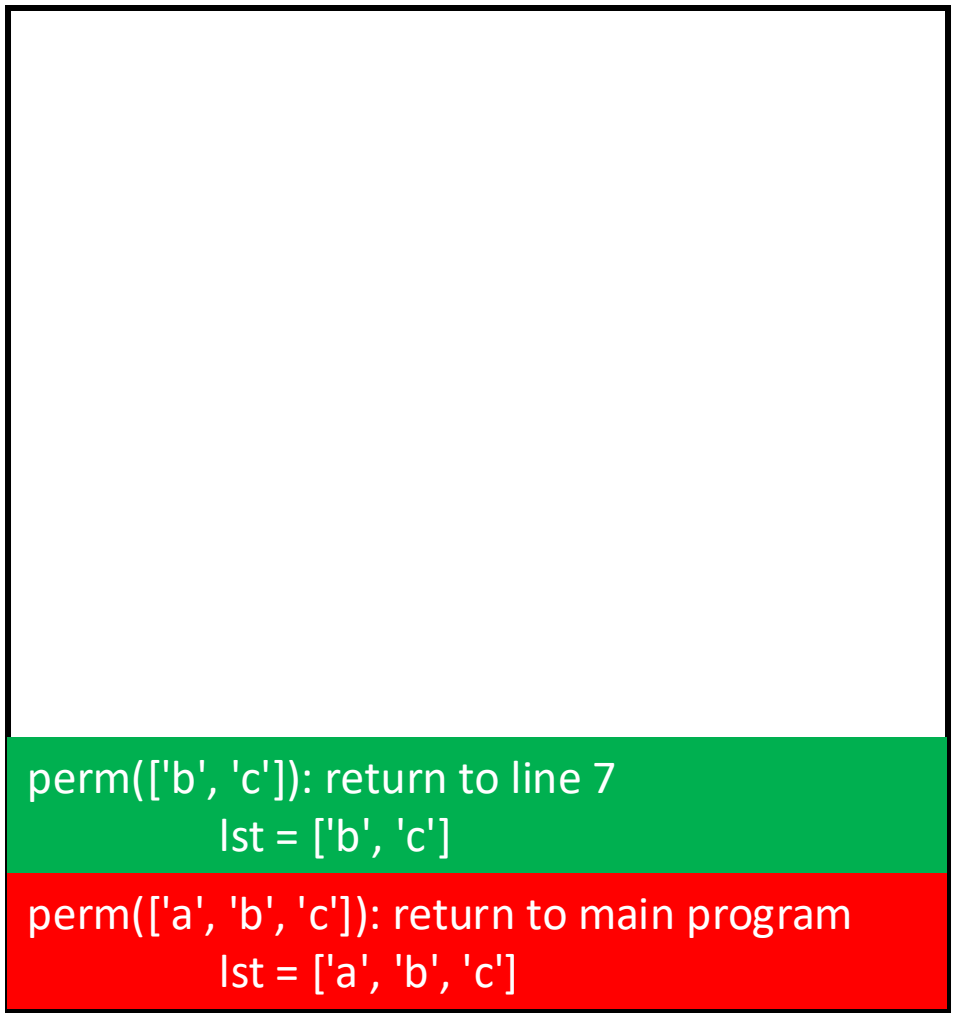
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)
  - i ← 1 (line 5)
  - remlst ← ['b'] (line 6)
  - p ← ['b']
  - intlst ← [['b', 'c'], ['c'] + ['b']] *or* [['b', 'c'], ['c', 'b']] (line 9)

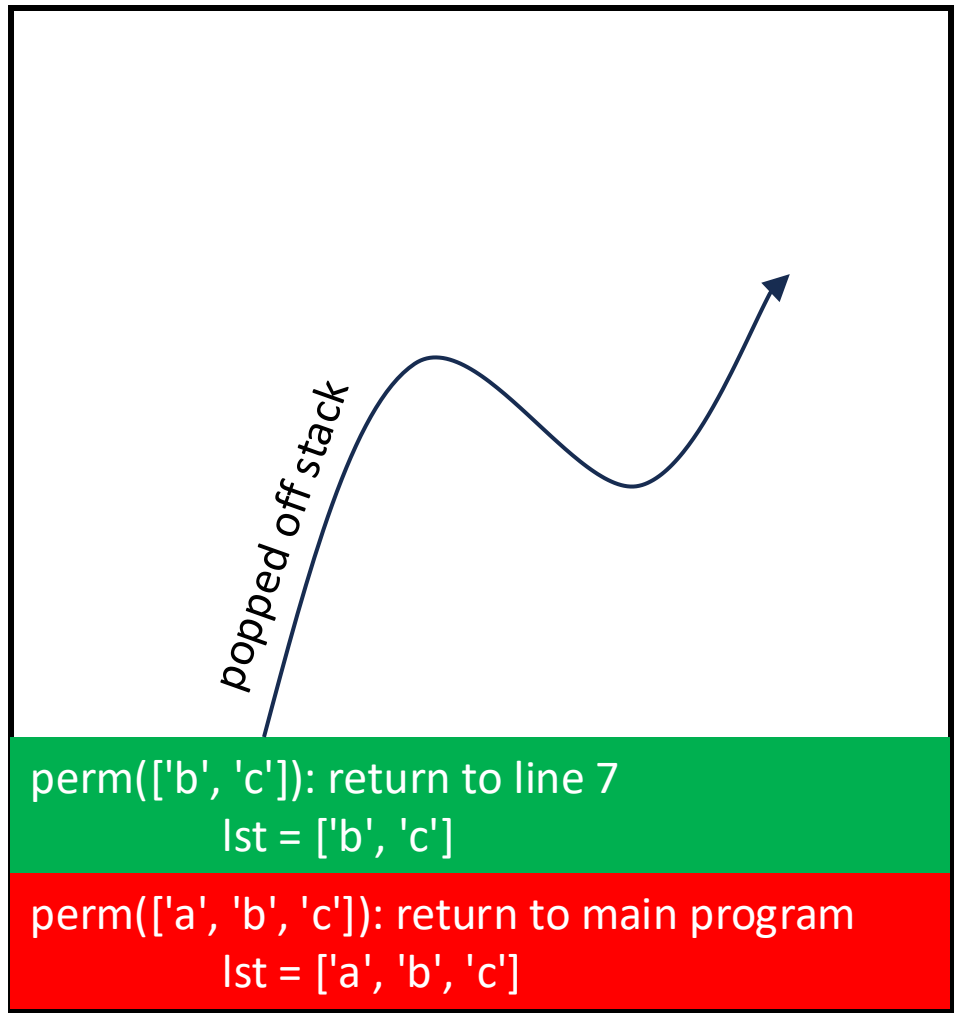
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [] (line 4)
- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- *call to* perm(['b', 'c']) (line 7)
  - returns [['b', 'c'], ['c', 'b']]

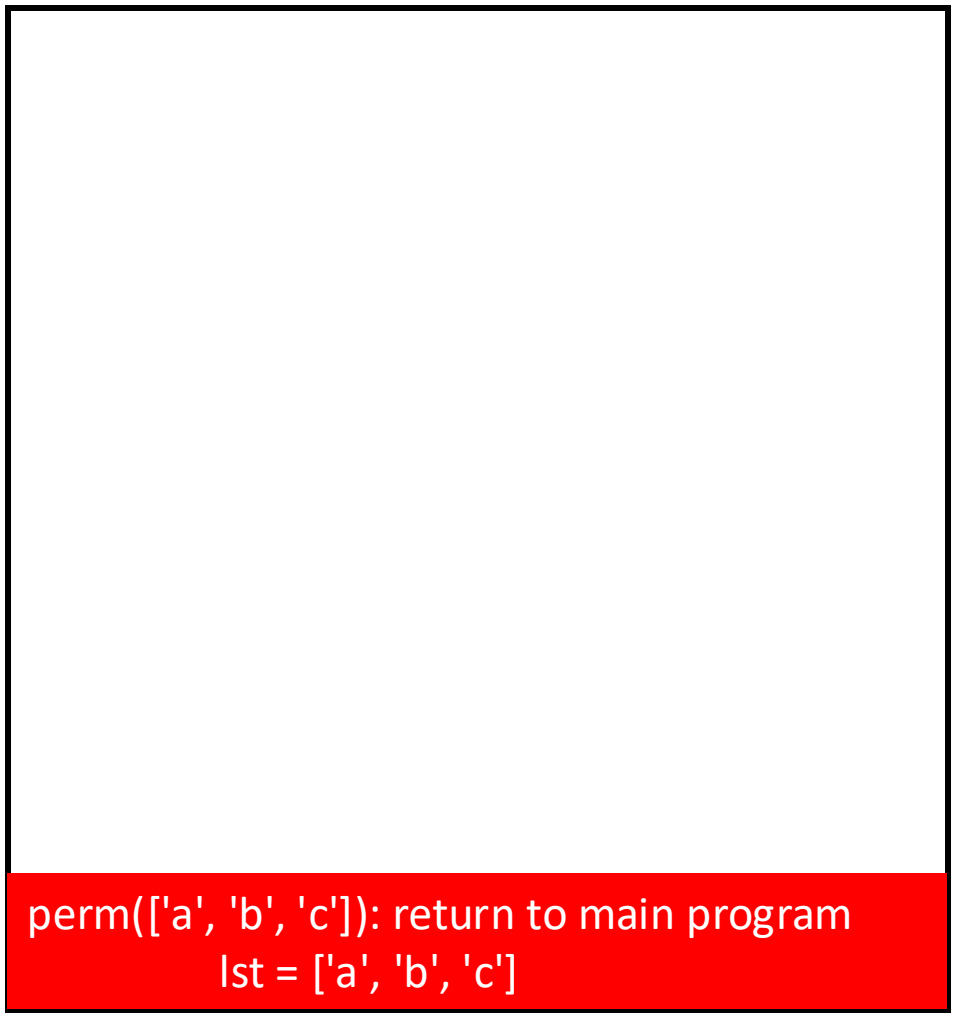
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- i ← 0 (line 5)
- remlst ← ['b', 'c'] (line 6)
- p ← [['b', 'c'], ['c', 'b']] (line 7)
- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]

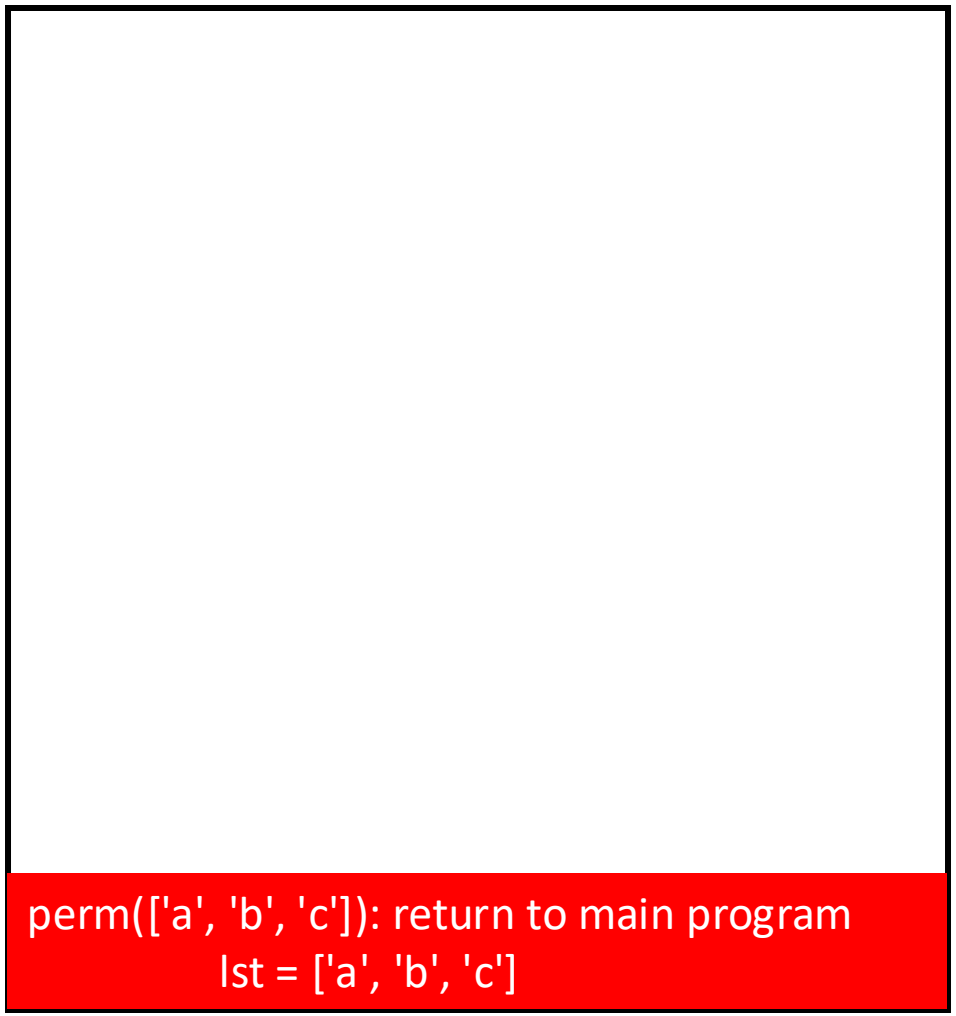
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)

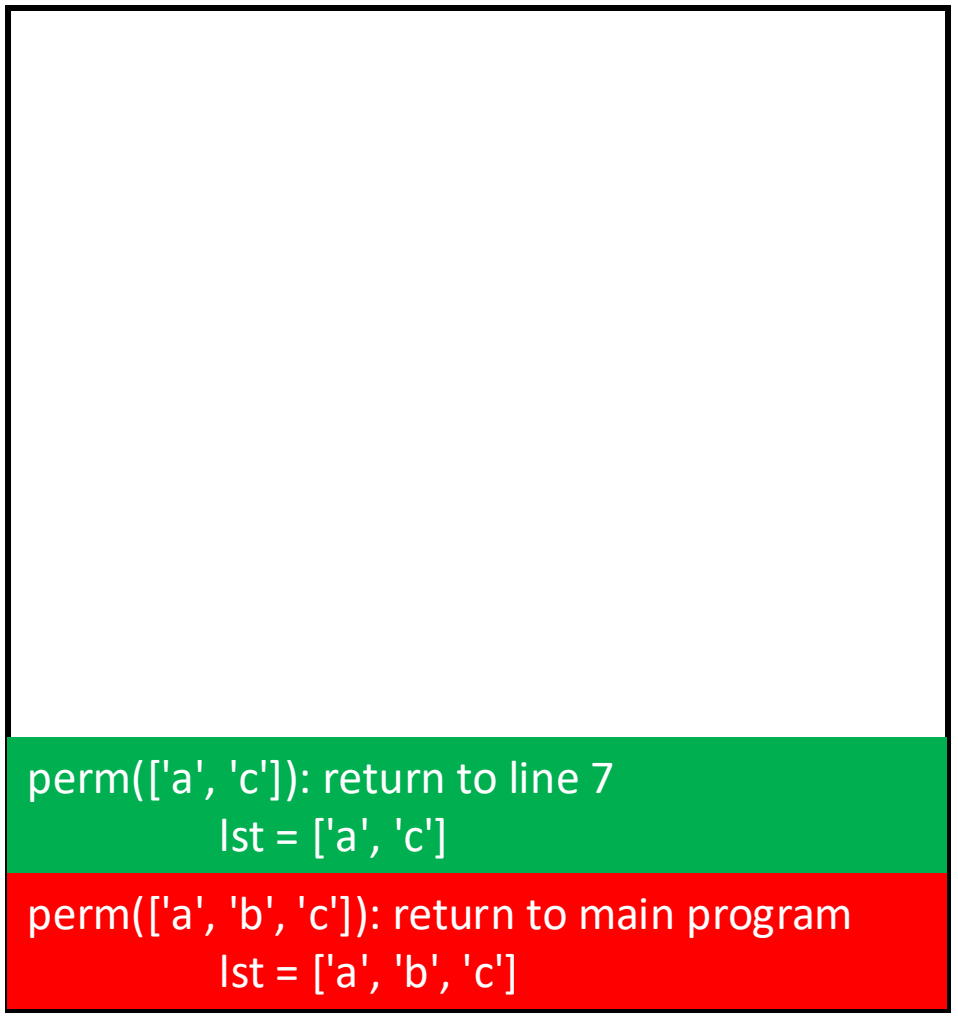
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c')] (line 7)

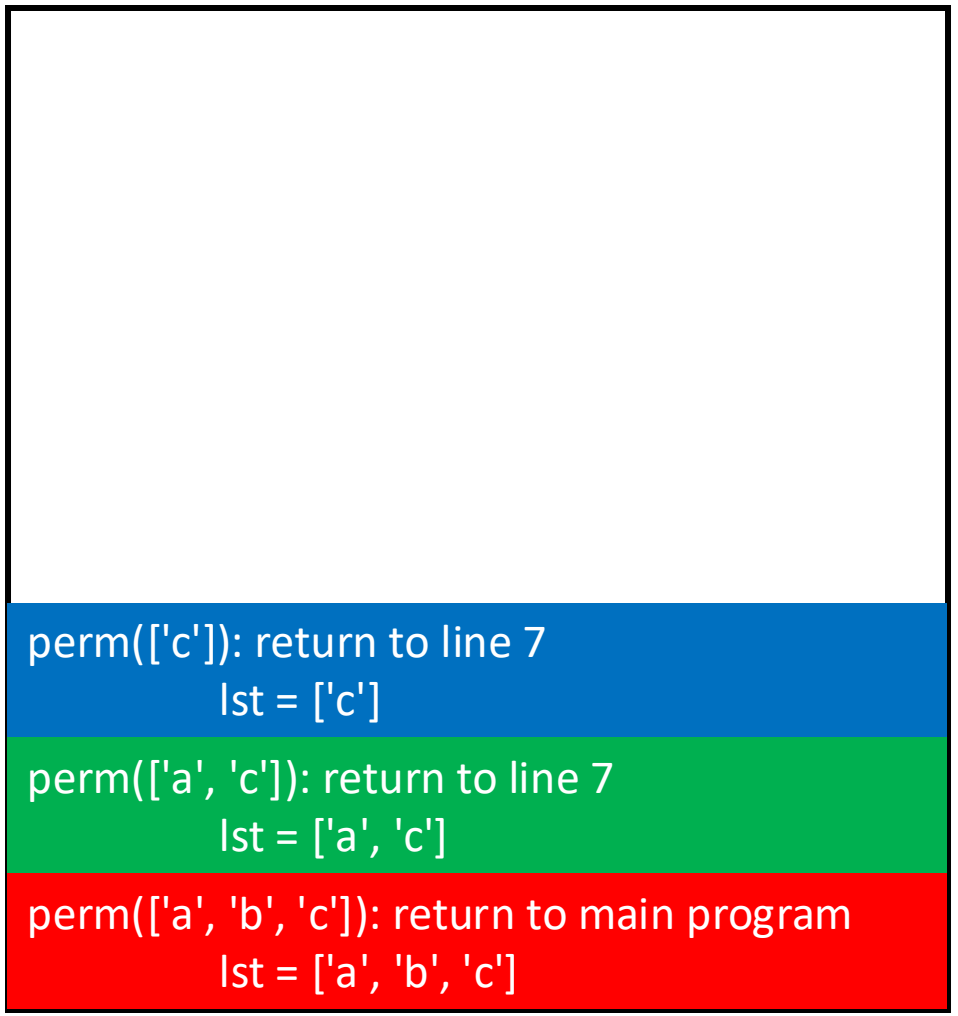
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - intlst ← [] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)
  - *call to* perm(['c']) (line 7)

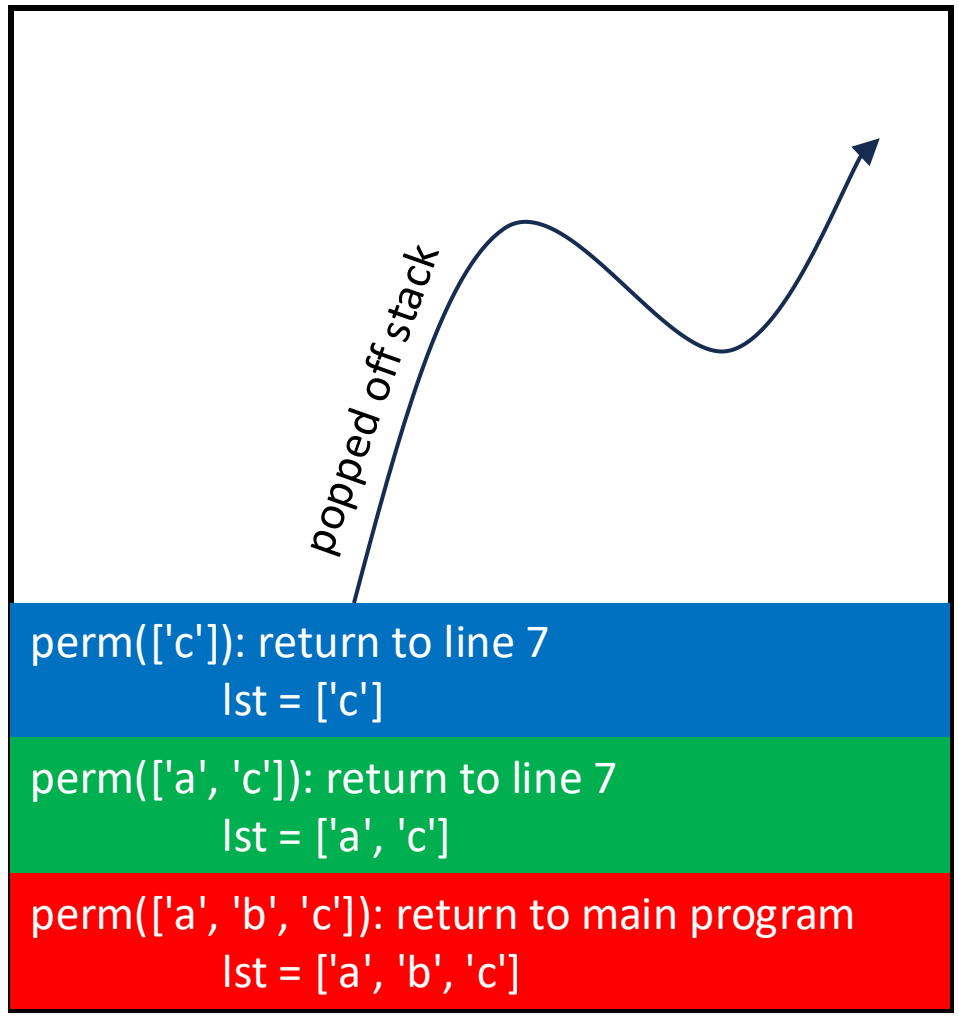
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - intlst ← [] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)
  - *call to* perm(['c']) (line 7)
    - as len(lst) = 1, returns [['c']] (line 3)

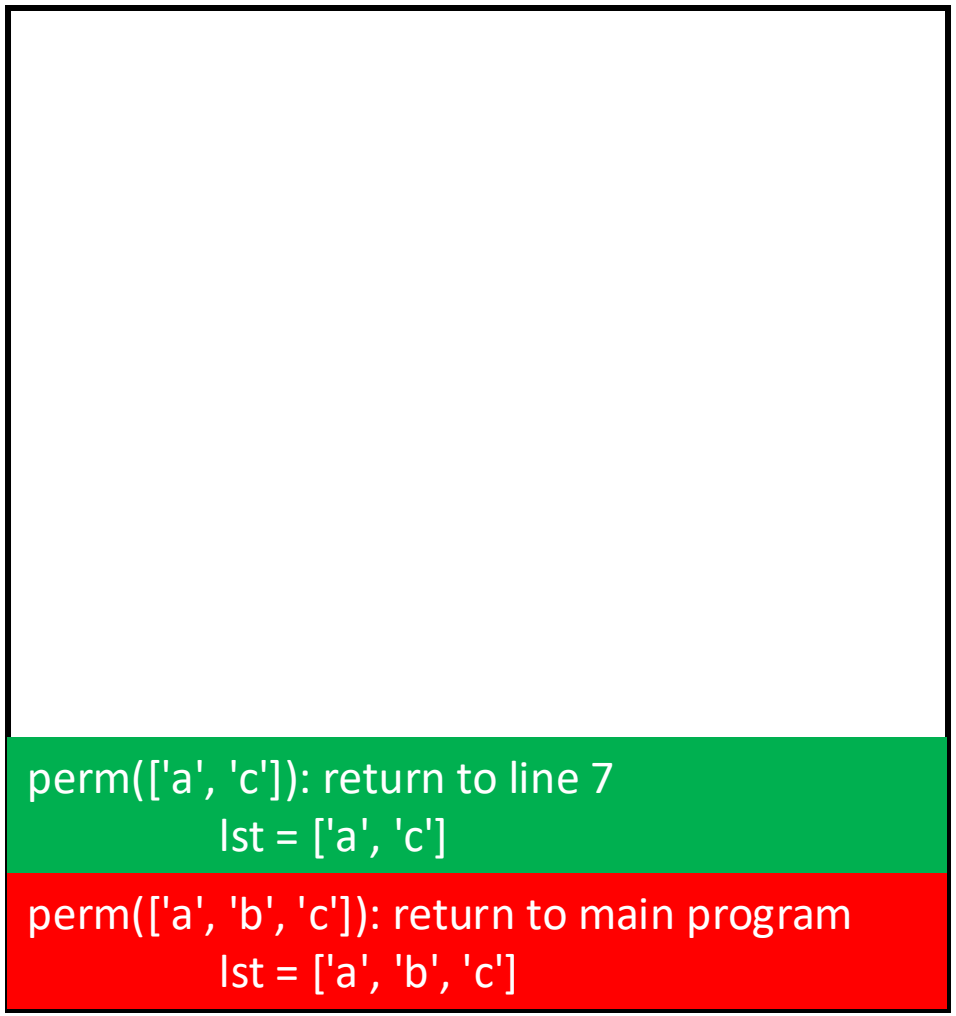
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - i ← 0 (line 5)
  - remlst ← ['c'] (line 6)
  - p ← ['c'] (line 7)
  - intlst ← [['a'] + ['c']] *or* [['a', 'c']] (line 9)

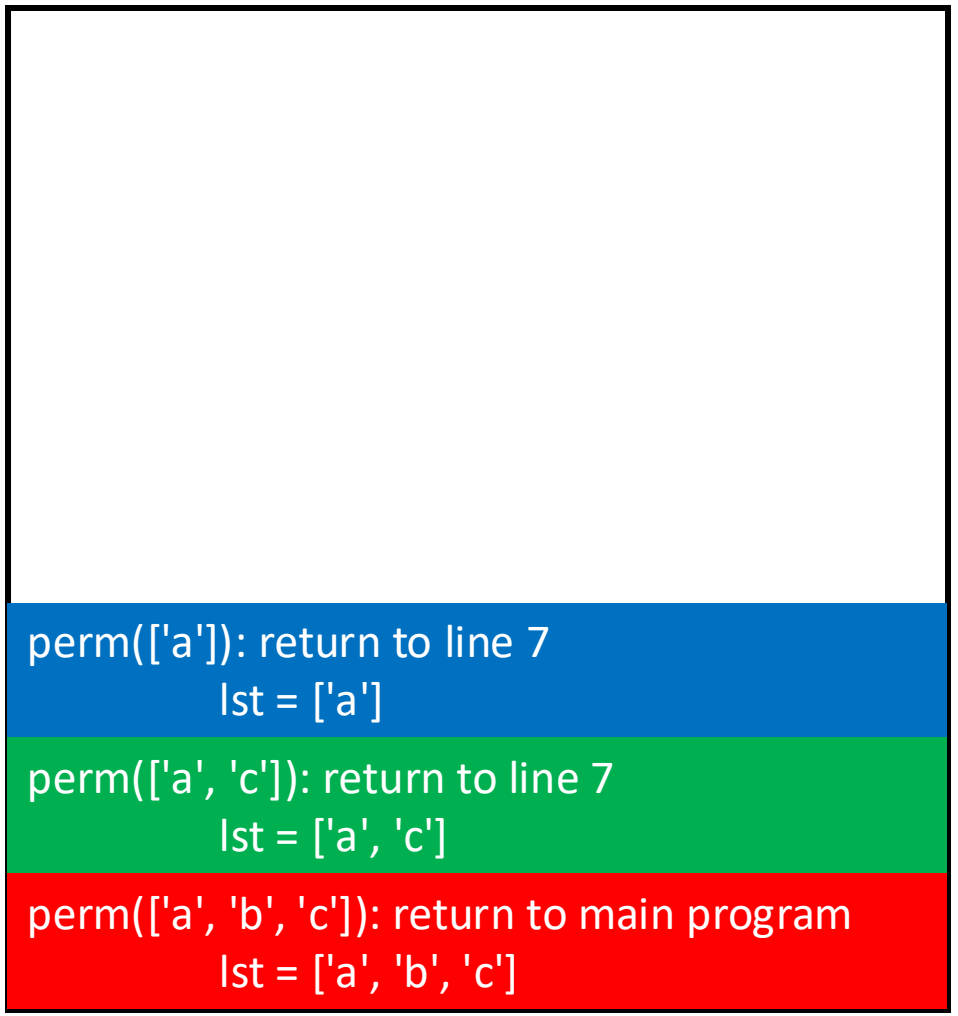
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - intlst ← [['a', 'c']]
  - i ← 1 (line 5)
  - remlst ← ['a'] (line 6)
  - *call to* perm(['a']) (line 7)

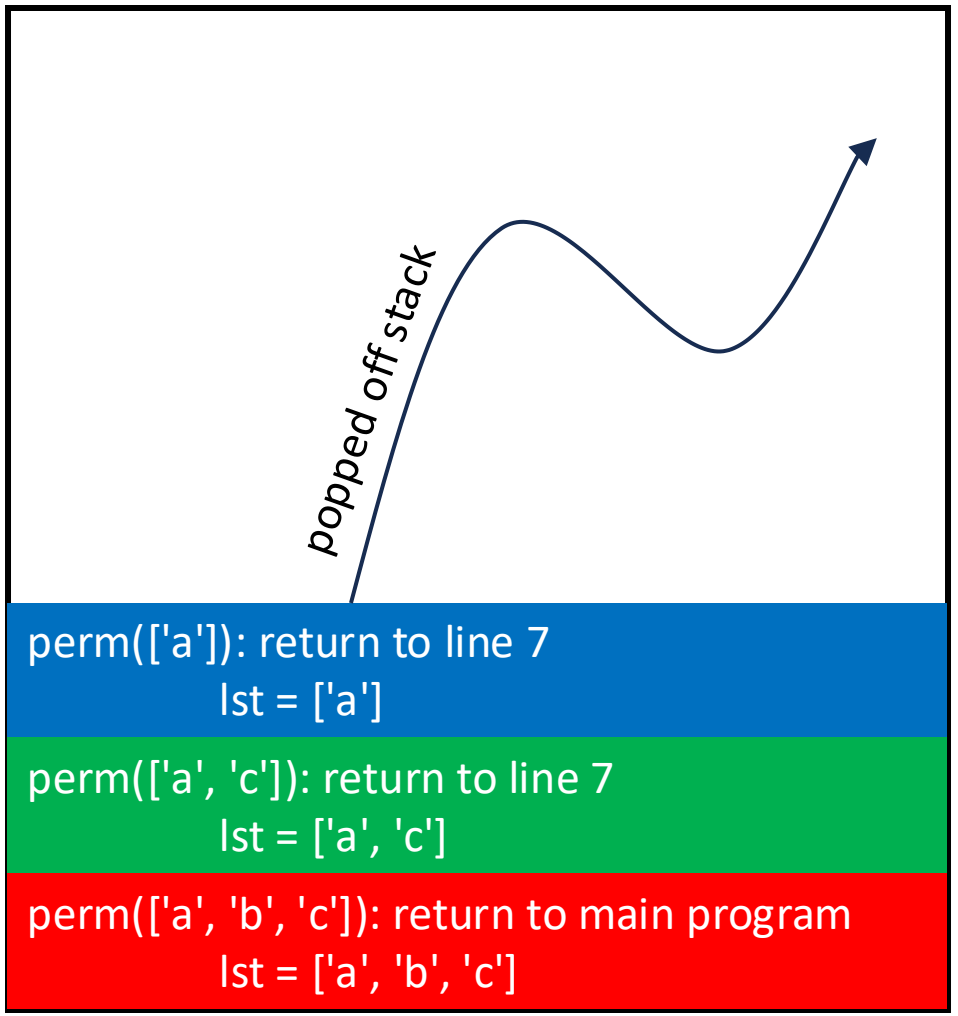
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - intlst ← [['a', 'c']] (line 9)
  - i ← 1 (line 5)
  - remlst ← ['a'] (line 6)
  - *call to* perm(['a']) (line 7)
    - as len(lst) = 1, returns [['a']] (line 3)

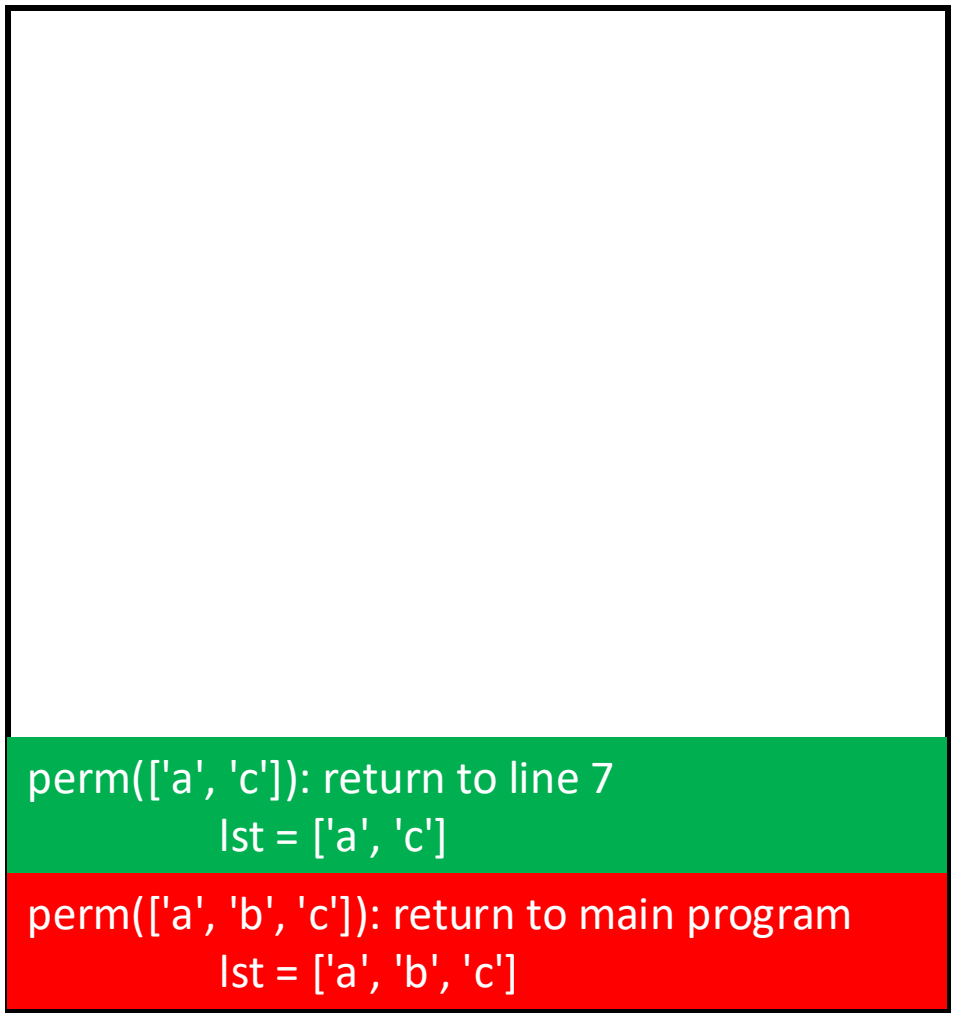
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - i ← 1 (line 5)
  - remlst ← ['a'] (line 6)
  - p ← ['a'] (line 7)
  - intlst ← [['a', 'c'], ['c'] + ['a']] *or* [['a', 'c'], ['c', 'a']]

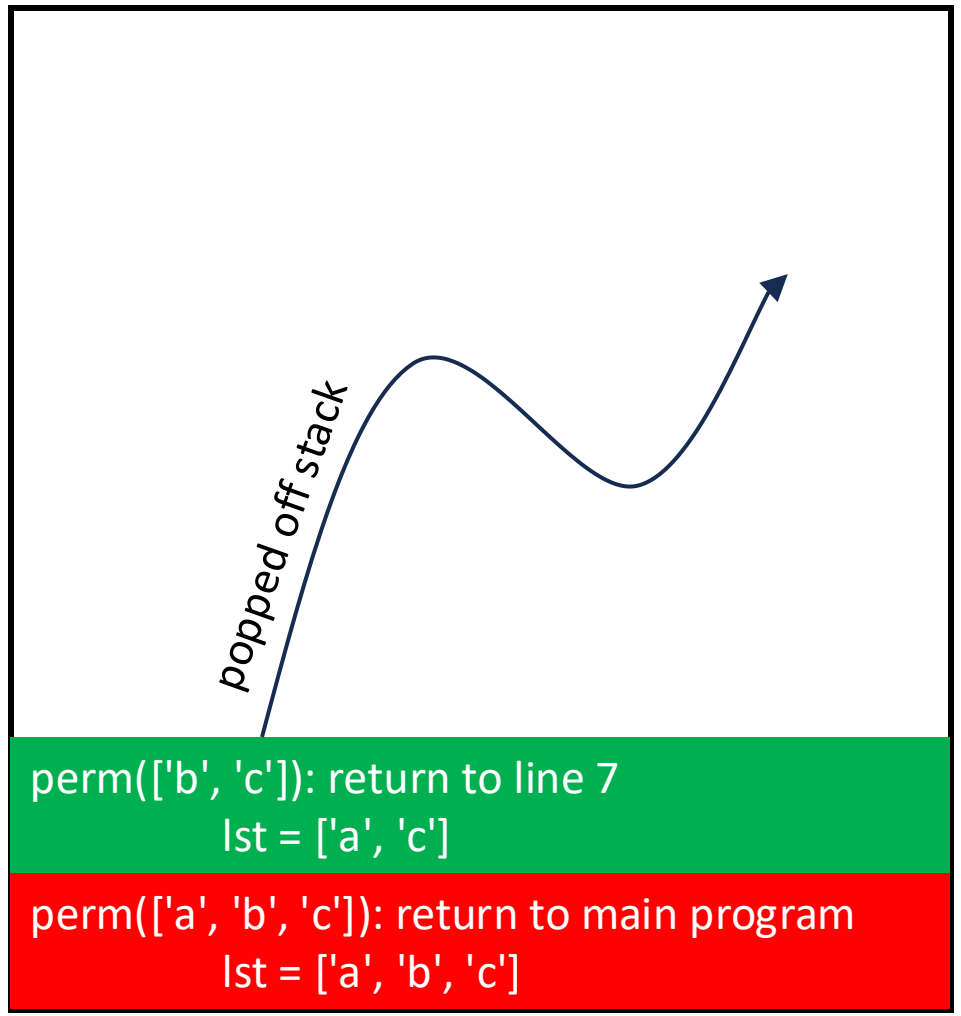
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a']]
- i ← 1 (line 5)
- remlst ← ['a', 'c'] (line 6)
- *call to* perm(['a', 'c']) (line 7)
  - return [['a', 'c'], ['c', 'a']]

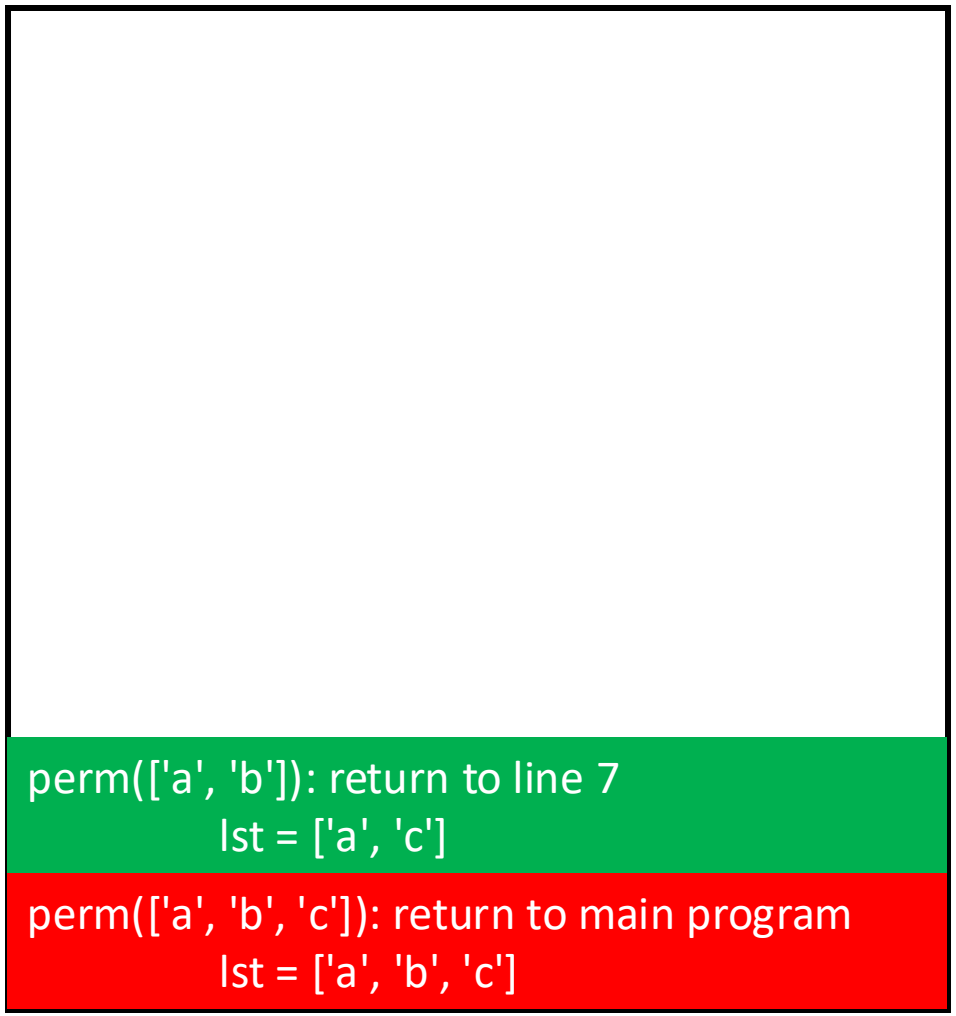
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])

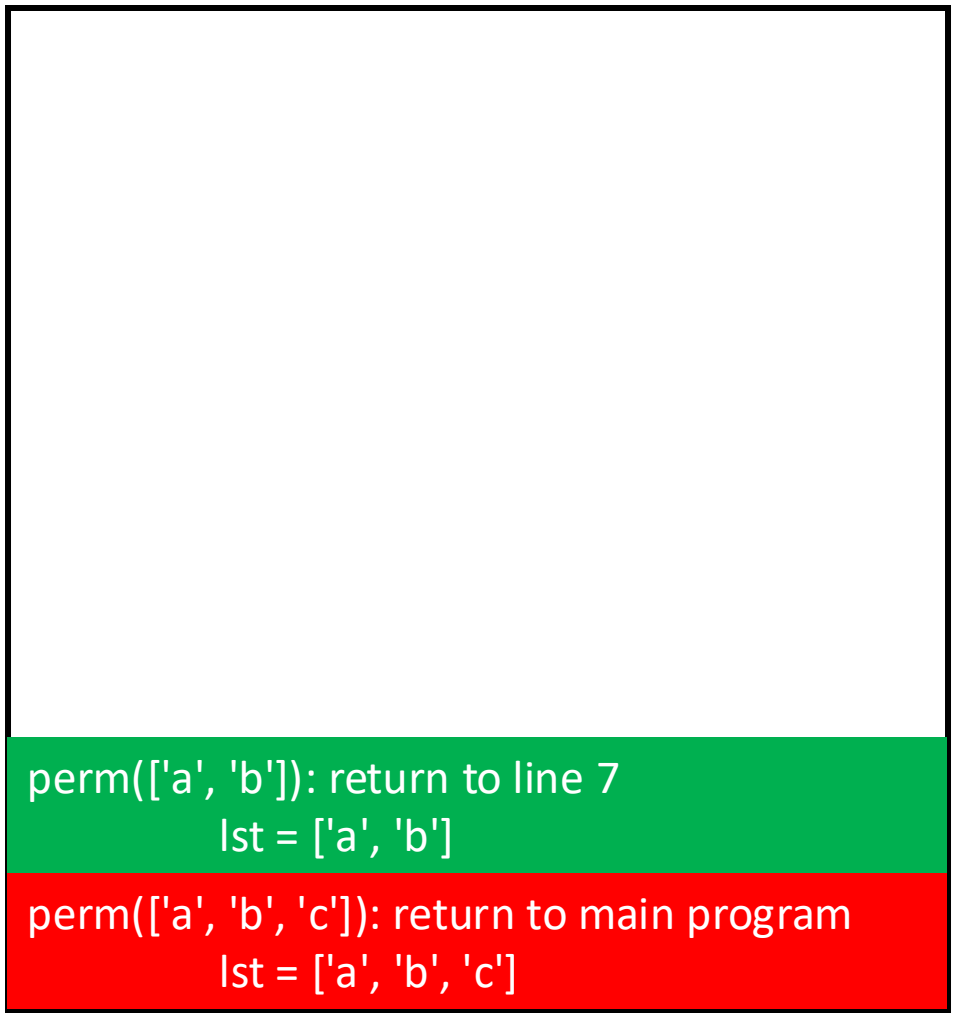
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - intlst ← [ ] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['b'] (line 6)

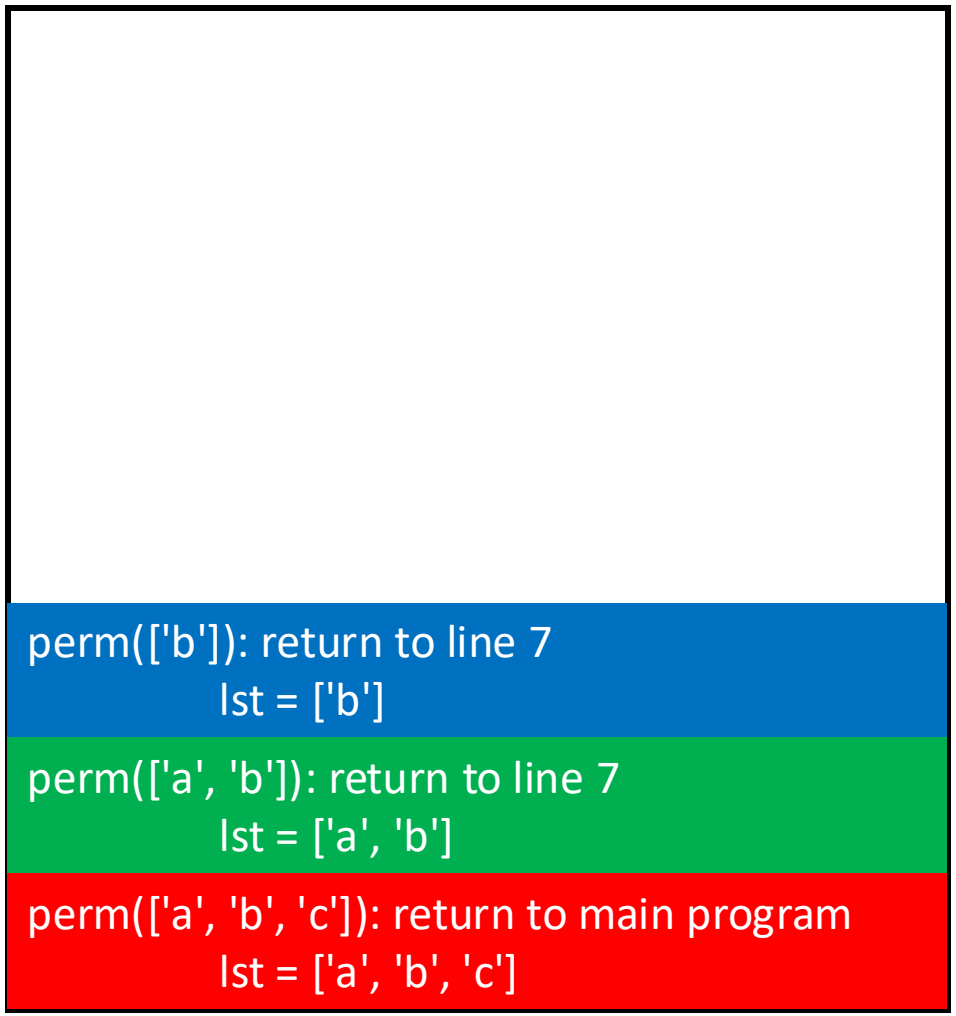
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - intlst ← [ ] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['b'] (line 6)
  - *call to* perm(['b']) (line 7)

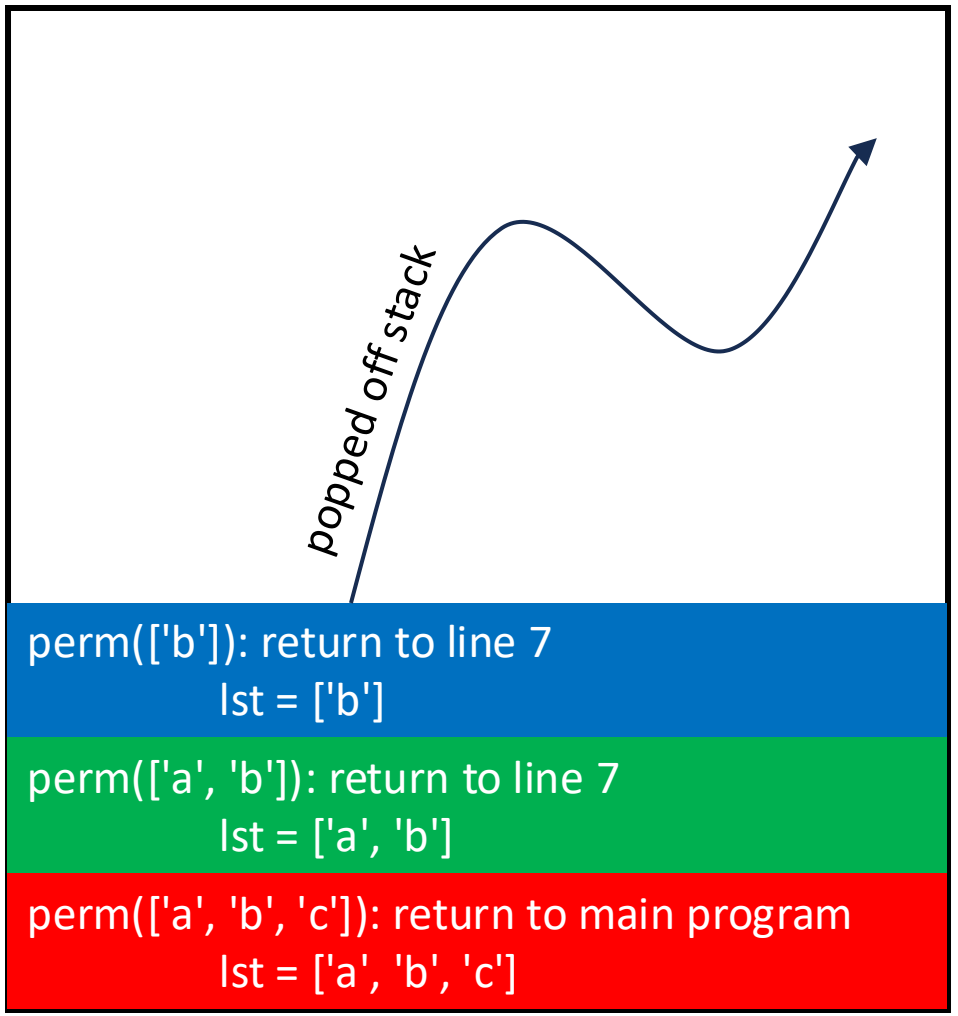
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - intlst ← [ ] (line 4)
  - i ← 0 (line 5)
  - remlst ← ['b'] (line 6)
    - as len(lst) = 1, returns [['b']] (line 3)

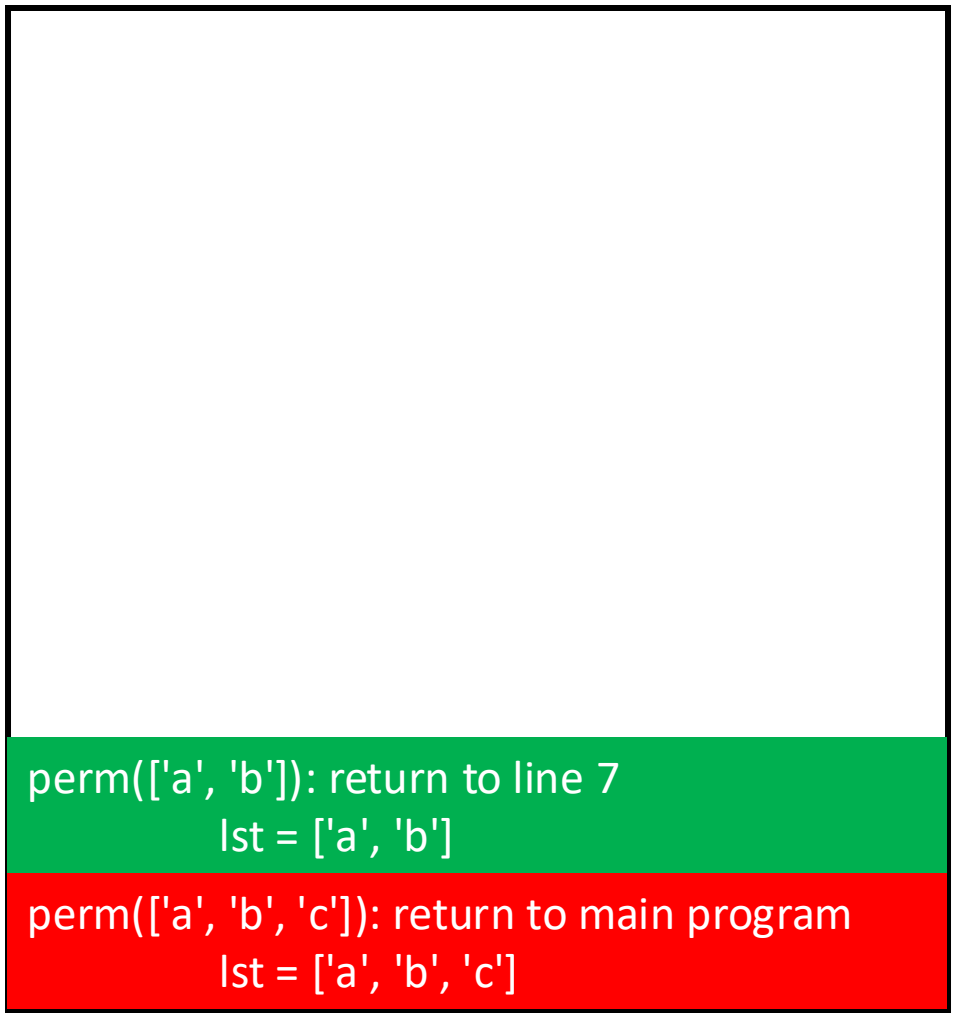
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - i ← 0 (line 5)
  - remlst ← ['b'] (line 6)
  - *call to* perm(['b']) (line 7)
  - intlst ← [['a'] + [b']] *or* [['a', 'b']] (line 9)

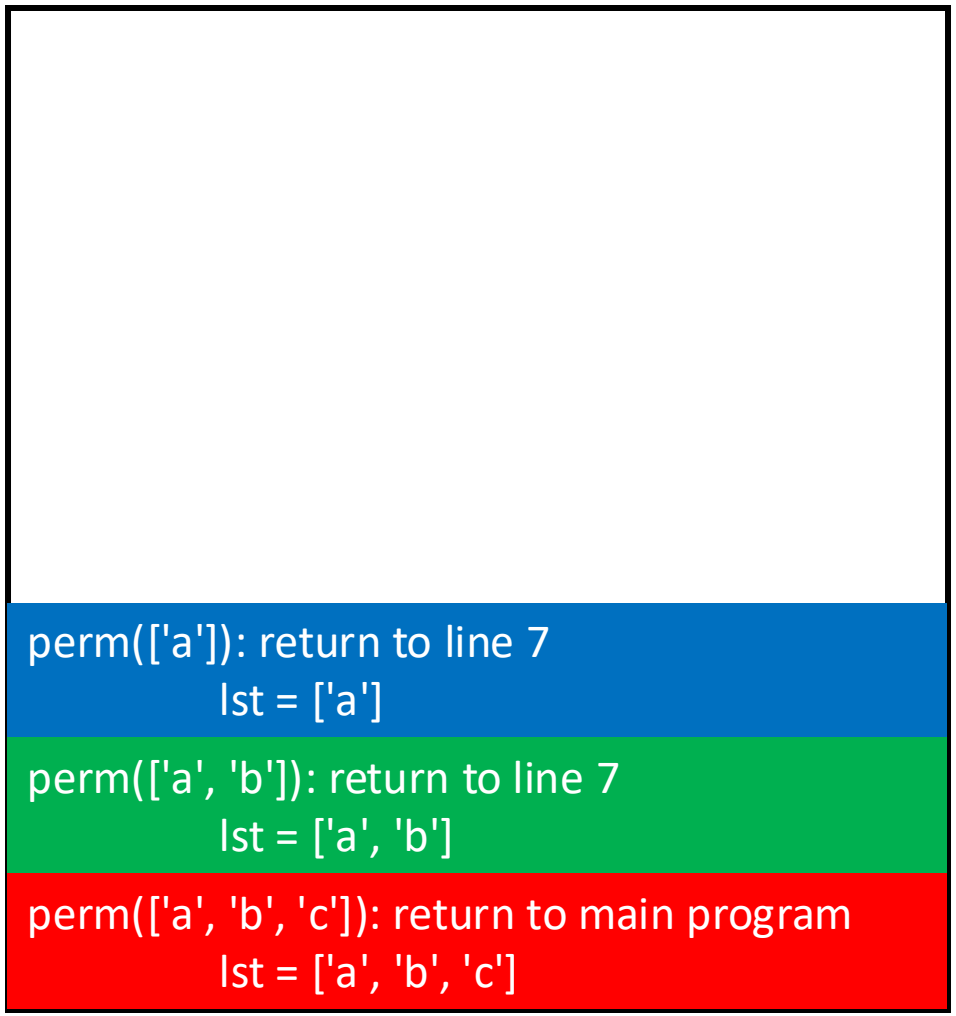
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - intlst ← [['a', 'b']] (see previous slide)
  - i ← 1 (line 5)
  - remlst ← ['a'] (line 6)
  - *call to* perm(['a']) (line 7)

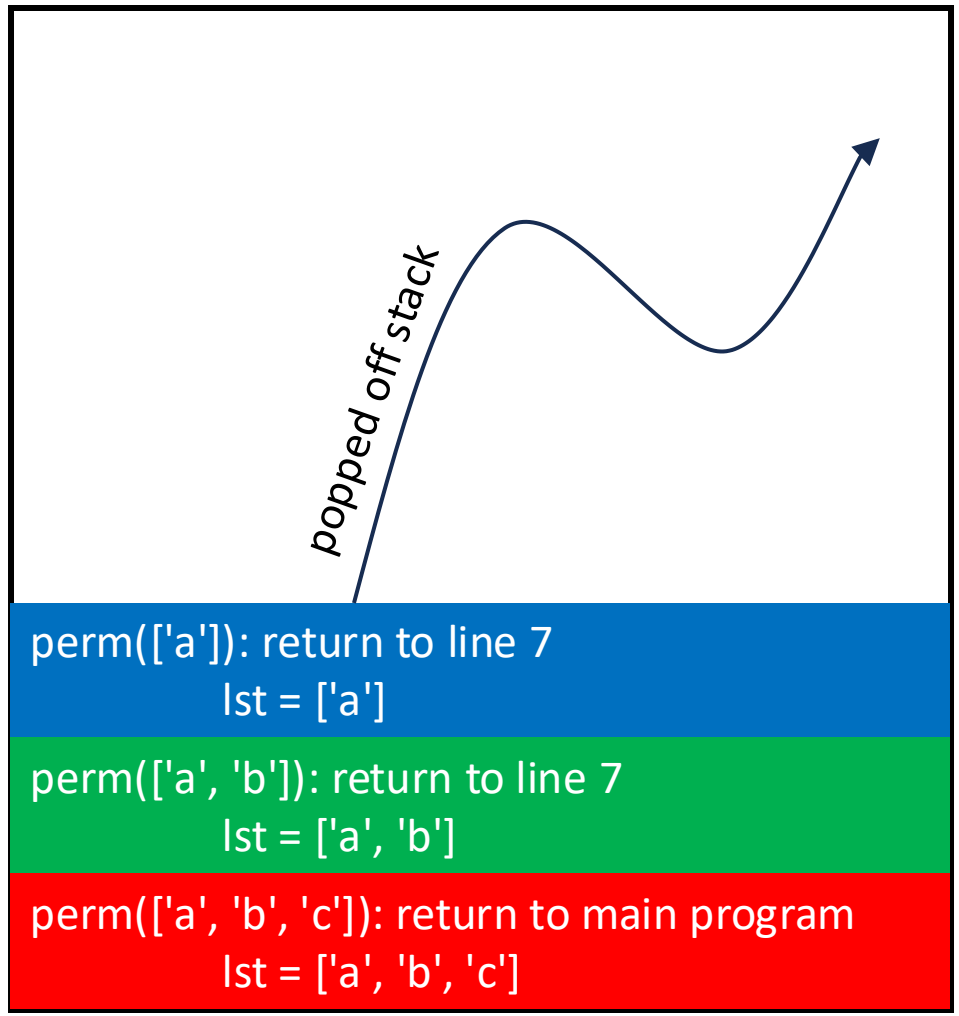
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - intlst ← [['a', 'b']] (see previous slide)
  - i ← 1 (line 5)
  - remlst ← ['a'] (line 6)
  - *call to* perm(['a']) (line 7)
    - as len(lst) = 1, returns [['a']] (line 3)

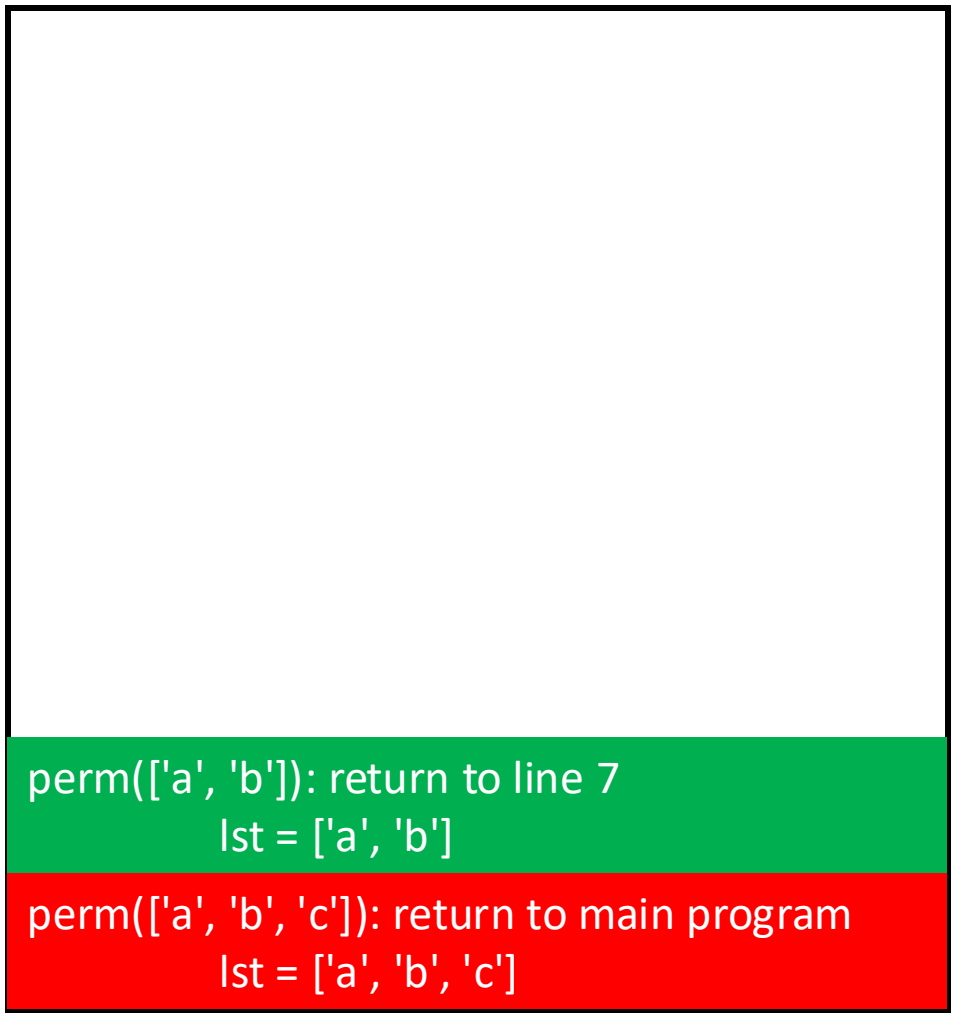
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - i ← 1 (line 5)
  - remlst ← ['a'] (line 6)
  - intlst ← [['a', 'b'] , ['b'] + ['a']] *or* [['a', 'b'], ['b', 'a']] (line 9)

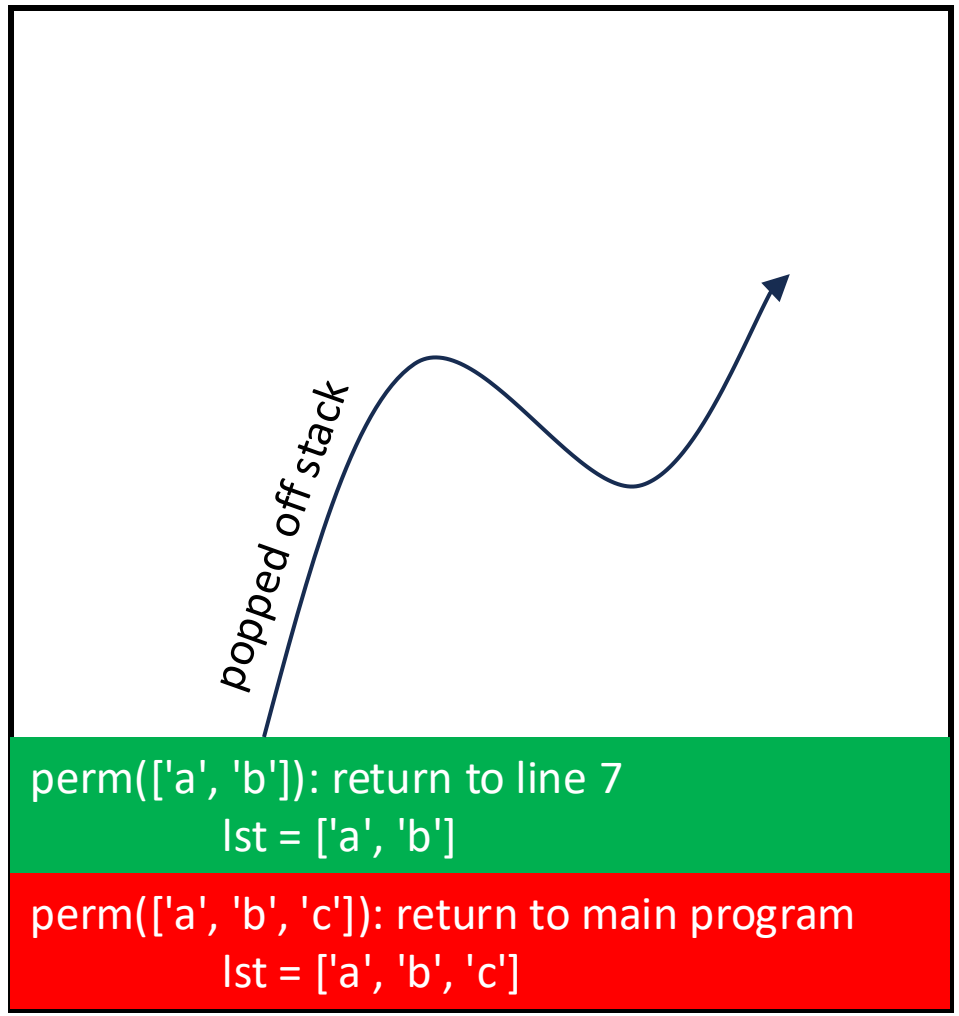
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - returns [['a', 'b'], ['b', 'a']]

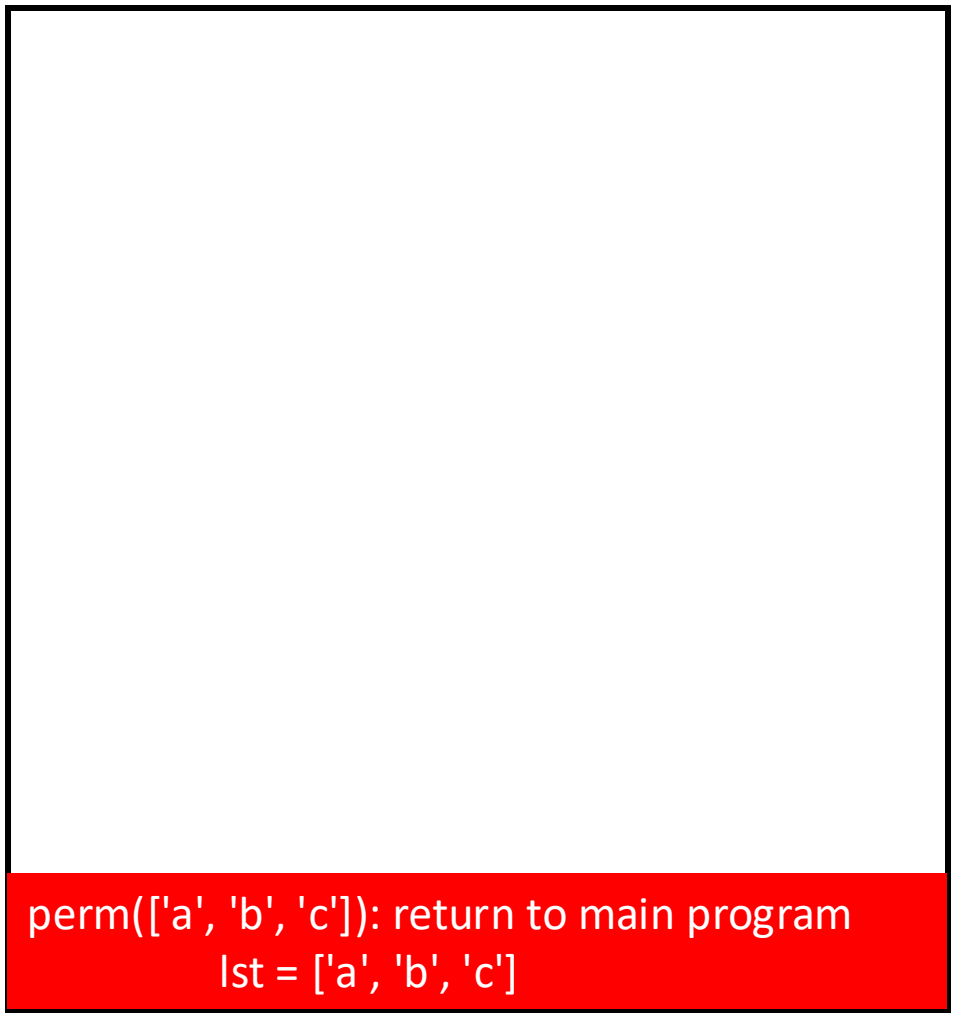
```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- returns [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b'], ['a', 'b', 'c'], ['b', 'a', 'c']]

```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```



Initial call to perm: perm(lst ← ['a', 'b', 'c'])

- intlst ← [['b', 'c', 'a'], ['c', 'b', 'a'], ['a', 'c', 'b'], ['c', 'a', 'b']]
- i ← 2 (line 5)
- remlst ← ['a', 'b'] (line 6)
- *call* perm(['a', 'b'])
  - returns [['a', 'b'], ['b', 'a']]

```
1: def perm(lst):
2:     if len(lst) == 0: return [ ]
3:     if len(lst) == 1: return [ lst ]
4:     intlst = [ ]
5:     for i in range(len(lst)):
6:         remlst = lst[:i] + lst[i+1:]
7:         for p in perm(remlst):
8:             intlst.append([ lst[i] + p)
9:     return intlst
```