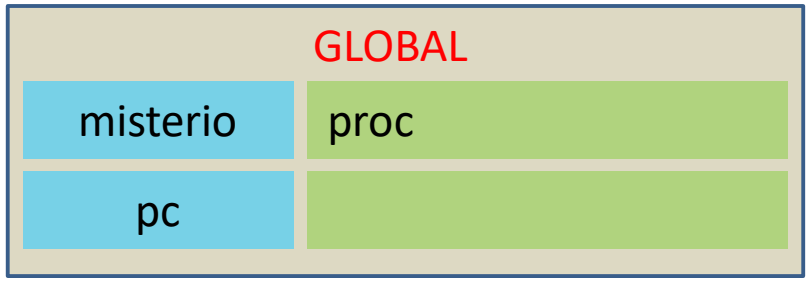
Examen 2 Problema 3 Misterio

Paso 1: Estado inicial.

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```



IMPRIME:

Nota: En cada llamada recursiva nos referiremos a los procedimientos recursivos con ‘ para ayudar a diferenciar

Examen 2 Problema 3 Misterio

Paso 2: Línea 10 inicia el ciclo for, ejecuta *misterio(5)* se crea el marco de pila de misterio.

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
	1	n	5
	0	pc	-

Examen 2 Problema 3 Misterio

Paso 3: Se evalúa la condición de la línea 2 *if n == 0*

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL

misterio	proc
x	-
pc	10

misterio

1	n	5
0	pc	2

IMPRIME:

Examen 2 Problema 3 Misterio

Paso 4: Como n != 0 **No se cumple** la condición y pasa a la línea 5, inicia el ciclo for, ejecuta *misterio(n-1)*, es decir, misterio(4), se crea el marco de pila de misterio'

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

Examen 2 Problema 3 Misterio

Paso 5: Se evalúa la condición de la línea 2 *if n == 0*

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

misterio			
	4	n	4
	3	pc	2

misterio			
	2	x	-
	1	n	5
	0	pc	~~2~~ 5

IMPRIME:

Examen 2 Problema 3 Misterio

Paso 6: Como n != 0 **No se cumple** la condición y pasa a la línea 5, inicia el ciclo for, ejecuta *misterio(n-1)*, es decir, misterio(3), se crea el marco de pila de misterio''

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
7	n	3
6	pc	-

misterio			
5	x	-
4	n	4
3	pc	~~2~~ 5

misterio			
2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 7: Se evalúa la condición de la línea 2 *if n == 0*

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
	7	n	3
	6	pc	2

misterio			
	5	x	-
	4	n	4
	3	pc	~~2~~ 5

misterio			
	2	x	-
	1	n	5
	0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 8: Como n != 0 **No se cumple** la condición y pasa a la línea 5, inicia el ciclo for, ejecuta *misterio(n-1)*, es decir, misterio(2), se crea el marco de pila de misterio'''

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
	10	n	2
	9	pc	-

misterio			
	8	x	-
	7	n	3
	6	pc	~~2~~ 5

misterio			
	5	x	-
	4	n	4
	3	pc	~~2~~ 5

misterio			
	2	x	-
	1	n	5
	0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 9: Se evalúa la condición de la línea 2 *if n == 0*

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
	10	n	2
	9	pc	2

misterio			
	8	x	-
	7	n	3
	6	pc	~~2~~ 5

misterio			
	5	x	-
	4	n	4
	3	pc	~~2~~ 5

misterio			
	2	x	-
	1	n	5
	0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 10: Como n != 0 **No se cumple** la condición y pasa a la línea 5, inicia el ciclo for, ejecuta *misterio(n-1)*, es decir, misterio(1), se crea el marco de pila de misterio’’’’

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
	13	n	1
	12	pc	-

misterio			
	11	x	-
	10	n	2
	9	pc	~~2~~ 5

misterio			
	8	x	-
	7	n	3
	6	pc	~~2~~ 5

misterio			
	5	x	-
	4	n	4
	3	pc	~~2~~ 5

misterio			
	2	x	-
	1	n	5
	0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 11: Se evalúa la condición de la línea 2 *if n == 0*

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
	13	n	1
	12	pc	2

misterio			
	11	x	-
	10	n	2
	9	pc	~~2~~ 5

misterio			
	8	x	-
	7	n	3
	6	pc	~~2~~ 5

misterio			
	5	x	-
	4	n	4
	3	pc	~~2~~ 5

misterio			
	2	x	-
	1	n	5
	0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 12: Como n != 0 **No se cumple** la condición y pasa a la línea 5, inicia el ciclo for, ejecuta *misterio(n-1)*, es decir, misterio(0), se crea el marco de pila de misterio’’’’’

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
16	n	0
15	pc	-

misterio			
14	x	-
13	n	1
12	pc	~~2~~ 5

misterio			
11	x	-
10	n	2
9	pc	~~2~~ 5

misterio			
8	x	-
7	n	3
6	pc	~~2~~ 5

misterio			
5	x	-
4	n	4
3	pc	~~2~~ 5

misterio			
2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 13: Se evalúa la condición de la línea 2
if n == 0

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
16	n	0
15	pc	2

misterio			
14	x	-
13	n	1
12	pc	~~2~~ 5

misterio			
11	x	-
10	n	2
9	pc	~~2~~ 5

misterio			
8	x	-
7	n	3
6	pc	~~2~~ 5

misterio			
5	x	-
4	n	4
3	pc	~~2~~ 5

misterio			
2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 14: Como n == 0 **Sí se cumple** la condición y pasa a la línea 3, ejecuta *yield [1]*

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio			
16	n	0
15	pc	~~2~~ 3

misterio			
14	x	-
13	n	1
12	pc	~~2~~ 5

misterio			
11	x	-
10	n	2
9	pc	~~2~~ 5

misterio			
8	x	-
7	n	3
6	pc	~~2~~ 5

misterio			
5	x	-
4	n	4
3	pc	~~2~~ 5

misterio			
2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 15: Vuelve al marco de misterio’’’’ y pasa a la línea 6, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1			
12	pc	~~2~~ ~~5~~ 6	17	r	[]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 16: Pasa a la línea 7, entra al ciclo for, ejecuta *suspenso(0,x)*, es decir, suspenso(0,[1])

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	-
12	pc	~~2~~ ~~5~~ ~~6~~ 7	17	r	[]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 17: Sabemos que suspenso(0,[1]) primero hace *yield 0+1* y luego hace *yield 1*, por lo tanto, *y* itera en esos valores

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	1
12	pc	~~2~~ ~~5~~ ~~6~~ 7	17	r	[]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 18: Pasa a la línea 8, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ 8	17	r	~~[]~~ [1]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 19: Vuelve a la línea 7, en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8~~ 7	17	r	~~[]~~ [1]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 20: Pasa a la línea 8, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ 8	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 21: Sale del ciclo for que itera en suspenso, pasa a la línea 9, ejecuta *yield r*, es decir, yield [1,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	-
10	n	2
9	pc	~~2~~ 5

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 22: Vuelve al marco de misterio''' y pasa a la línea 6, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2			
9	pc	~~2~~ ~~5~~ 6	19	r	[]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 23: Pasa a la línea 7, entra al ciclo for, ejecuta *suspenso(0,x)*, es decir, suspenso(0,[1,1])

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	-
9	pc	~~2~~ ~~5~~ ~~6~~ 7	19	r	[]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 24: Sabemos que suspenso(0,[1,1]) primero hace *yield 0+1,* luego hace *yield 1+1*, y luego hace *yield 1* por lo tanto, *y* itera en esos valores

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0			
15	pc	~~2~~ 3			

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	1
9	pc	~~2~~ ~~5~~ ~~6~~ 7	19	r	[]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 25: Pasa a la línea 8, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	1
9	pc	~~2 5 6 7~~ 8	19	r	~~[]~~ [1]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 26: Vuelve a la línea 7, en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ 2
9	pc	~~2 5 6 7 8~~ 7	19	r	~~[]~~ [1]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 27: Pasa a la línea 8, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ 2
9	pc	~~2 5 6 7 8 7~~ 8	19	r	~~[]~~ ~~[1]~~ [1,2]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 28: Vuelve a la línea 7, en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8~~ 7	19	r	~~[]~~ ~~[1]~~ [1,2]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 29: Pasa a la línea 8, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ 8	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 30: Sale del ciclo for que itera en suspenso, pasa a la línea 9, ejecuta *yield r*, es decir, yield [1,2,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	-
7	n	3
6	pc	~~2~~ 5

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 31: Vuelve al marco de misterio'' y pasa a la línea 6, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0			
15	pc	~~2~~ 3			

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3			
6	pc	~~2 5~~ 6	21	r	[]

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 32: Pasa a la línea 7, entra al ciclo for, ejecuta *suspenso(0,x)*, es decir, suspenso(0,[1,2,1])

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[] [1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ 8 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	-
6	pc	~~2 5 6~~ 7	21	r	[]

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 33: Sabiendo ya el comportamiento de este ciclo, es claro que *y* itera sobre 1, 3, 3, 1, por lo tanto *r* queda como [1,3,3,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[] [1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ 8 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ 7-8	21	r	[1,3,3,1]

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 34: Sale del ciclo for que itera en suspenso, pasa a la línea 9, ejecuta *yield r*, es decir, yield [1,3,3,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5 6 7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	-
4	n	4
3	pc	~~2~~ 5

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 35: Vuelve al marco de misterio' y pasa a la línea 6, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ ~~7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4			
3	pc	~~2 5~~ 6	23	r	[]

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 36: Pasa a la línea 7, entra al ciclo for, ejecuta *suspenso(0,x)*, es decir, suspenso(0,[1,3,3,1])

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ ~~7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	-
3	pc	~~2 5 6~~ 7	23	r	[]

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 37: Sabiendo ya el comportamiento de este ciclo, es claro que *y* itera sobre 1, 4, 6, 4, 1, por lo tanto *r* queda como [1,4,6,4,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ ~~7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6~~ 4 1
3	pc	~~2 5 6~~ 7-8	23	r	[1,4,6,4,1]

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 38: Sale del ciclo for que itera en suspenso, pasa a la línea 9, ejecuta *yield r*, es decir, yield [1,4,6,4,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	19	r	~~[]~~[1]~~[1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	-
1	n	5
0	pc	~~2~~ 5

Examen 2 Problema 3 Misterio

Paso 39: Vuelve al marco de misterio y pasa a la línea 6, asigna a r

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[] [1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ 8 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5 6 7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6 4~~ 1
3	pc	~~2 5 6 7 8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5			
0	pc	~~2 5~~ 6	25	r	[]

Examen 2 Problema 3 Misterio

Paso 40: Pasa a la línea 7, entra al ciclo for, ejecuta *suspenso(0,x)*, es decir, suspenso(0,[1,4,6,4,1])

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0			
15	pc	~~2~~ 3			

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ ~~7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6 4~~ 1
3	pc	~~2 5 6 7 8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5			
0	pc	~~2 5 6~~ 7	25	r	[]

Examen 2 Problema 3 Misterio

Paso 41: Sabiendo ya el comportamiento de este ciclo, es claro que *y* itera sobre 1, 5, 10, 10, 5, 1, por lo tanto *r* queda como [1,5,10,10,5,1]

```
1  def misterio(n):
2        if n == 0:
3              yield [1]
4        else:
5              for x in misterio(n-1):
6                    r = []
7                    for y in suspenso(0, x):
8                          r = [*r, y]
9                    yield r
10 for x in misterio(5):
11       print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0			
15	pc	~~2~~ 3			

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ ~~7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6 4~~ 1
3	pc	~~2 5 6 7 8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1 5 10 10 5~~ 1
0	pc	~~2 5 6~~ 7-8	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 42: Sale del ciclo for que itera en suspenso, pasa a la línea 9, ejecuta *yield r*, es decir, yield [1,5,10,10,5,1]

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	-
pc	10

IMPRIME:

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[] [1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ 8 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5 6 7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6 4~~ 1
3	pc	~~2 5 6 7 8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1 5 10 10 5~~ 1
0	pc	~~2 5 6 7 8~~ 9	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 43: Vuelve al marco global y pasa a la línea 11, imprime: **x = [1, 5, 10, 10, 5, 1]**

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ 11

IMPRIME: **[1,5,10,10,5,1]**

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7~~ ~~8~~ 9	19	r	~~[][1][1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5~~ ~~6~~ ~~7 8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6~~ 4 1
3	pc	~~2 5 6 7 8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1 5 10 10 5~~ 1
0	pc	~~2 5 6 7 8~~ 9	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 44: Vuelve a la línea 10, en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	19	r	~~[]~~[1]~~[1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 45: Vuelve al marco de misterio, vuelve a la línea 5 en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	19	r	~~[]~~[1]~~[1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 46: Vuelve al marco de misterio', vuelve a la línea 5 en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	19	r	~~[]~~[1]~~[1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ 9	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 47: Vuelve al marco de misterio'', vuelve a la línea 5 en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0			
15	pc	~~2~~ 3			

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	19	r	~~[]~~~~[1]~~~~[1,2]~~ [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 48: Vuelve al marco de misterio’’’, vuelve a la línea 5 en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ 9	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~9~~ 5	19	r	~~[]~~~~[1]~~~~[1,2]~~ [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 49: Vuelve al marco de misterio’’’’, vuelve a la línea 5 en el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0
15	pc	~~2~~ 3

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8 9~~ 5	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1 2~~ 1
9	pc	~~2 5 6 7 8 7 8 7 8 9~~ 5	19	r	~~[][1][1,2]~~ [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1 3 3~~ 1
6	pc	~~2 5 6 7 8 9~~ 5	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1 4 6 4~~ 1
3	pc	~~2 5 6 7 8 9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1 5 10 10 5~~ 1
0	pc	~~2 5 6 7 8 9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 50: Vuelve al marco de misterio’’’’’, termina la ejecución de misterio()

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

16	n	0
15	pc	~~2~~ ~~3~~

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~9~~ 5	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~9~~ 5	19	r	~~[]~~[1]~~[1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 51: Se libera el marco de misterio’’’’’. Vuelve al marco de misterio’’’’, termina el ciclo for, termina la ejecución de misterio()

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

~~misterio~~ 0 ~~pc 2 3~~

misterio

14	x	[1]			
13	n	1	18	y	~~1~~ 1
12	pc	~~2 5 6 7 8 7 8 9 5~~	17	r	~~[]~~ ~~[1]~~ [1,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2 5 6 7 8 7 8 7 8 9~~ 5	19	r	~~[]~~[1]~~[1,2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 52: Se libera el marco de misterio’’’’. Vuelve al marco de misterio’’’, termina el ciclo for, termina la ejecución de misterio()

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

11	x	[1,1]			
10	n	2	20	y	~~1~~ ~~2~~ 1
9	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~7~~ ~~8~~ ~~9~~ ~~5~~	19	r	~~[]~~[~~1~~][~~1~~,~~2~~] [1,2,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

Examen 2 Problema 3 Misterio

Paso 53: Se libera el marco de misterio'''. Vuelve al marco de misterio'', termina el ciclo for, termina la ejecución de misterio()

```
1  def misterio(n):
2        if n == 0:
3              yield [1]
4        else:
5              for x in misterio(n-1):
6                    r = []
7                    for y in suspenso(0, x):
8                          r = [*r, y]
9                    yield r
10 for x in misterio(5):
11       print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]

misterio

8	x	[1,2,1]			
7	n	3	22	y	~~1~~ ~~3~~ ~~3~~ 1
6	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~5~~	21	r	[1,3,3,1]

misterio

5	x	[1,3,3,1]			
4	n	4	24	y	~~1~~ ~~4~~ ~~6~~ ~~4~~ 1
3	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	23	r	[1,4,6,4,1]

misterio

2	x	[1,4,6,4,1]			
1	n	5	26	y	~~1~~ ~~5~~ ~~10~~ ~~10~~ ~~5~~ 1
0	pc	~~2~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ 5	25	r	[1,5,10,10,5,1]

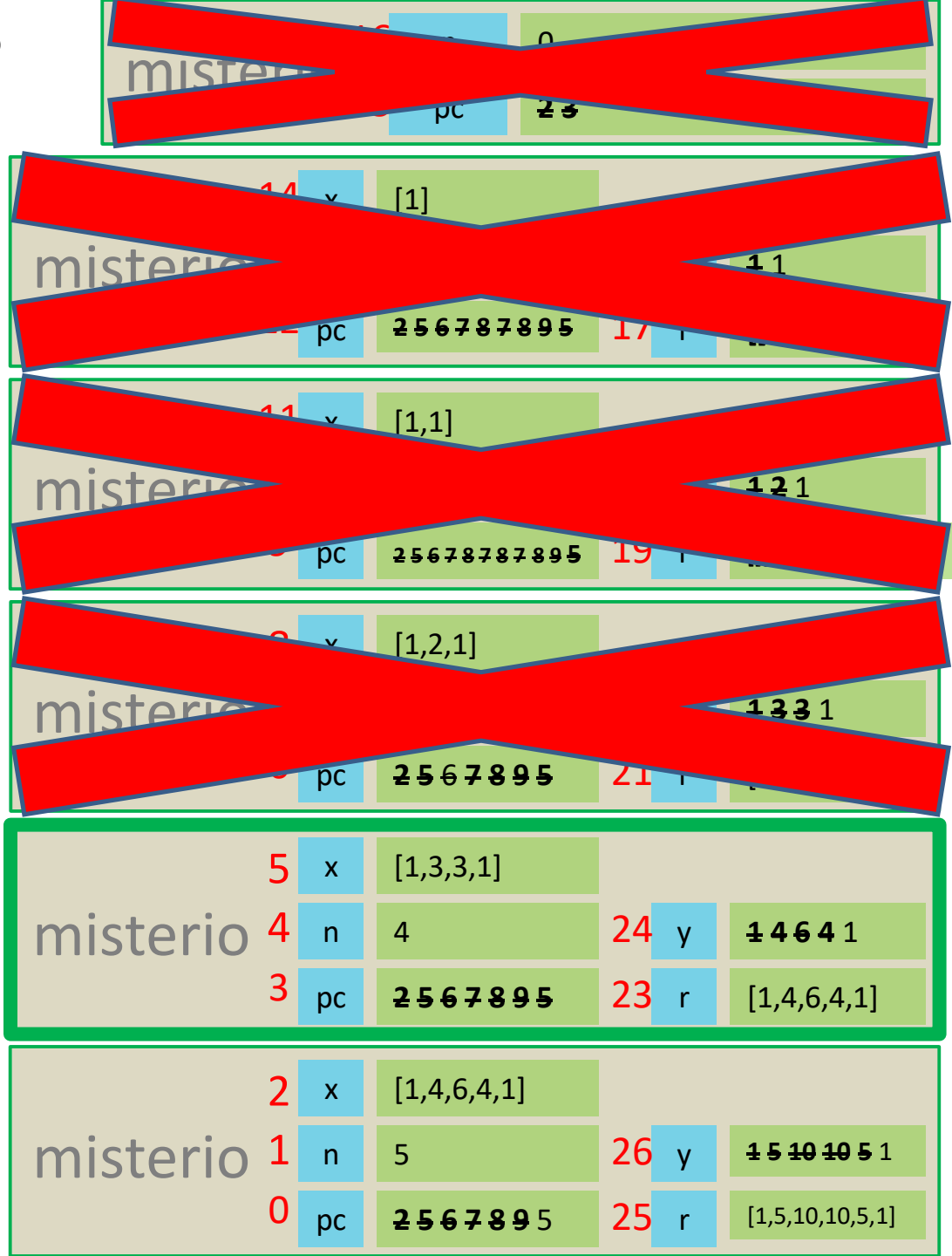
Examen 2 Problema 3 Misterio

Paso 54: Se libera el marco de misterio''. Vuelve al marco de misterio', termina el ciclo for, termina la ejecución de misterio()

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]



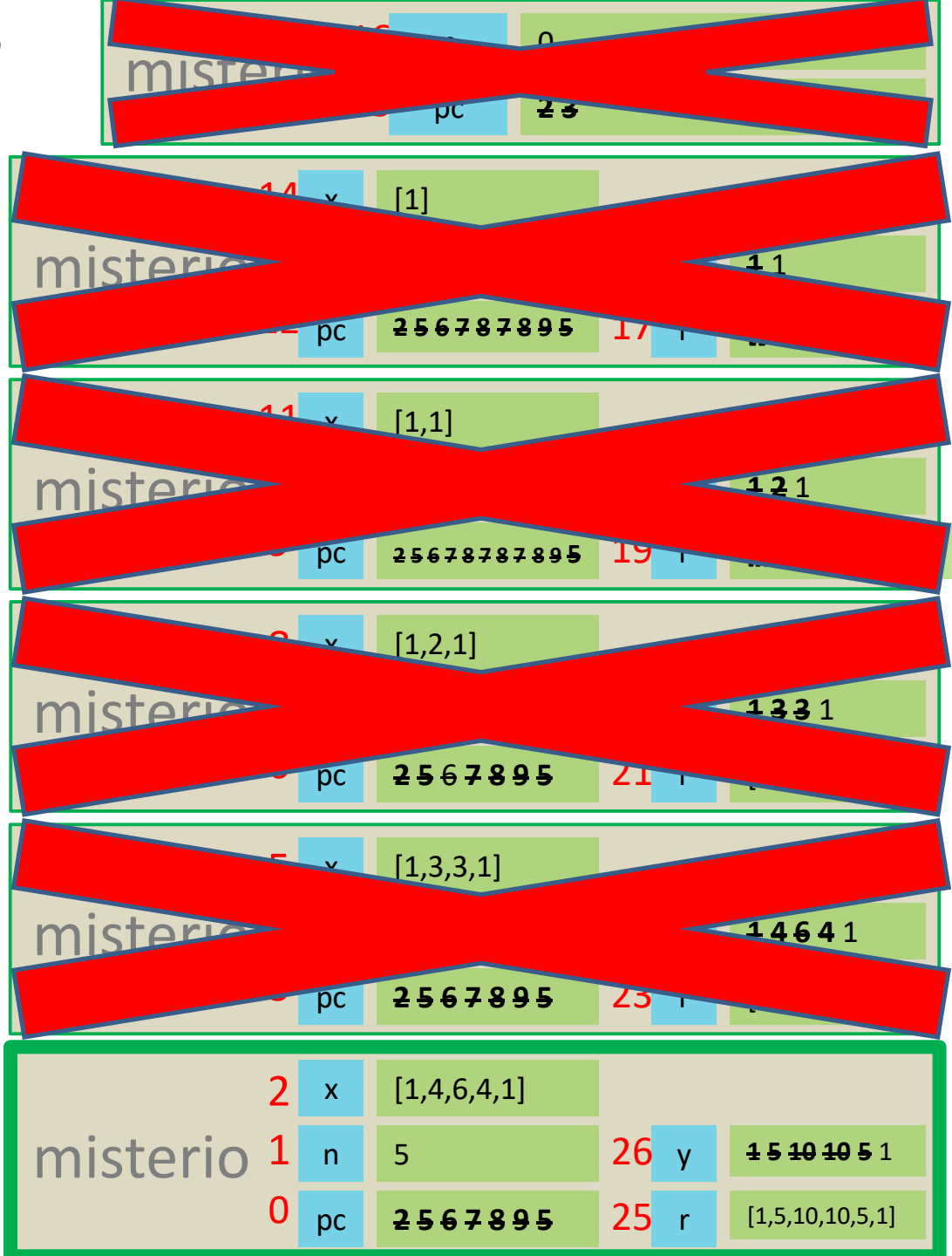
Examen 2 Problema 3 Misterio

Paso 55: Se libera el marco de misterio'. Vuelve al marco de misterio, termina el ciclo for, termina la ejecución de misterio()

```
1  def misterio(n):
2        if n == 0:
3              yield [1]
4        else:
5              for x in misterio(n-1):
6                    r = []
7                    for y in suspenso(0, x):
8                          r = [*r, y]
9                    yield r
10 for x in misterio(5):
11       print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ 10

IMPRIME: [1,5,10,10,5,1]



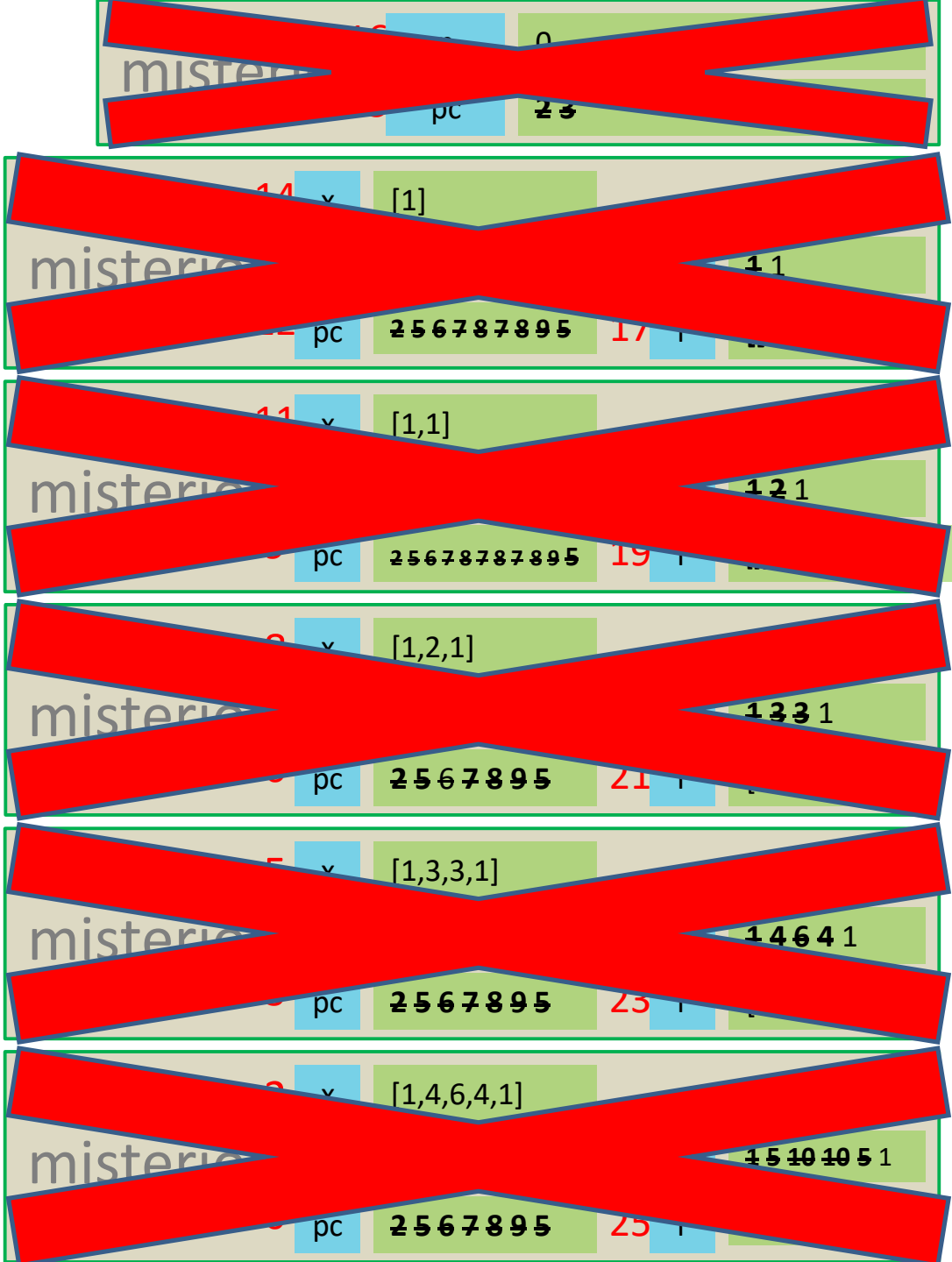
Examen 2 Problema 3 Misterio

Paso 56: Se libera el marco de misterio.
Vuelve al marco global, termina el ciclo for

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ ~~10~~

IMPRIME: [1,5,10,10,5,1]



Examen 2 Problema 3 Misterio

Paso 57: Termina la ejecución del programa

```
1  def misterio(n):
2          if n == 0:
3                  yield [1]
4          else:
5                  for x in misterio(n-1):
6                          r = []
7                          for y in suspenso(0, x):
8                                  r = [*r, y]
9                          yield r
10 for x in misterio(5):
11         print(x)
```

GLOBAL	
misterio	proc
x	[1,5,10,10,5,1]
pc	~~10~~ ~~11~~ ~~10~~

IMPRIME: [1,5,10,10,5,1]