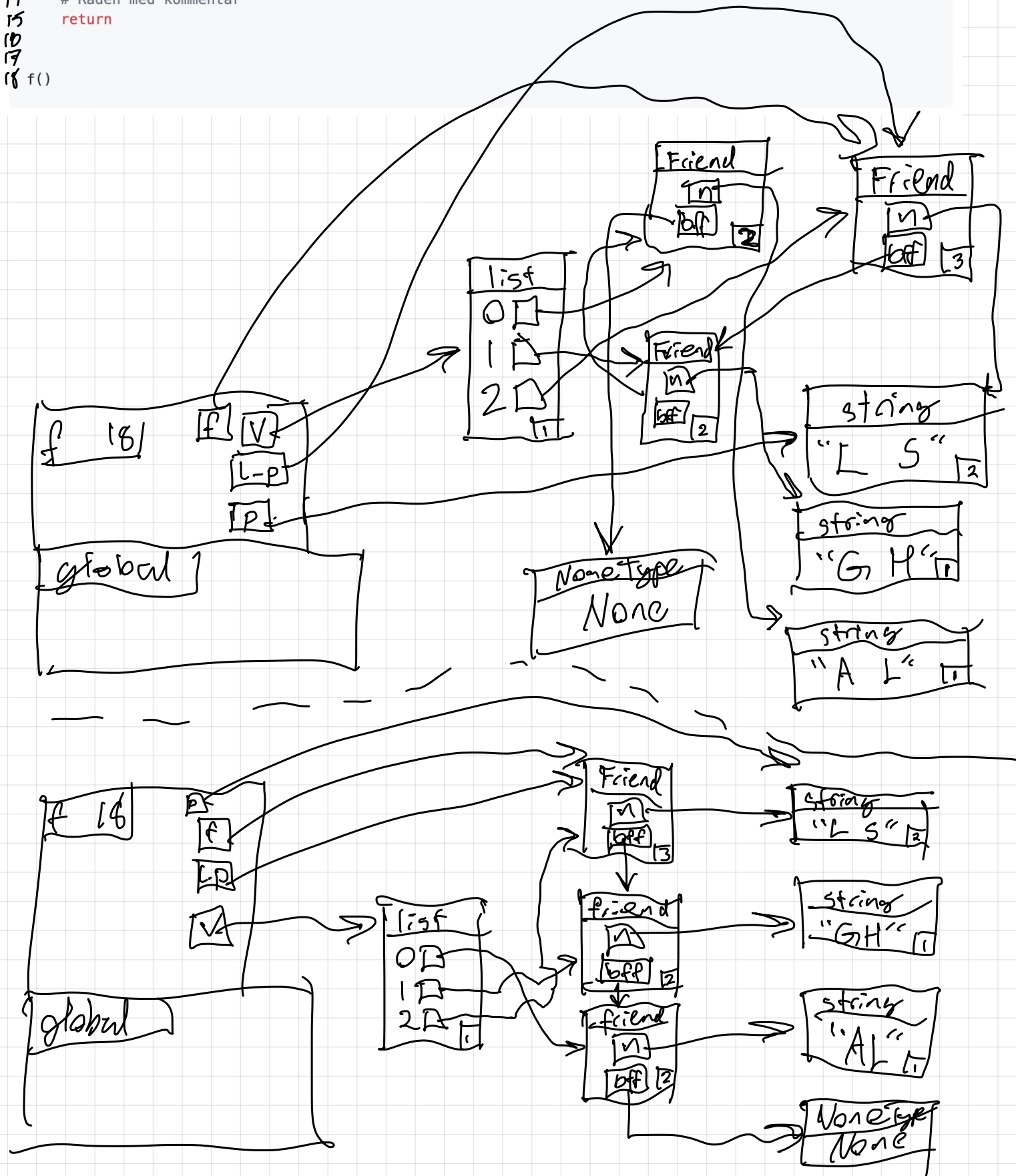


```

1 class Friend:
2     def __init__(self, name, bff):
3         self.name = name
4         self.bff = bff
5
6
7 def f():
8     v = []
9     last_person = None
10    for person in ["Ada Lovelace", "Grace Hopper", "Lisa Su"]:
11        f = Friend(person, last_person)
12        last_person = f
13        v.append(f)
14    # Raden med kommentar
15    return
16
17 f()
18

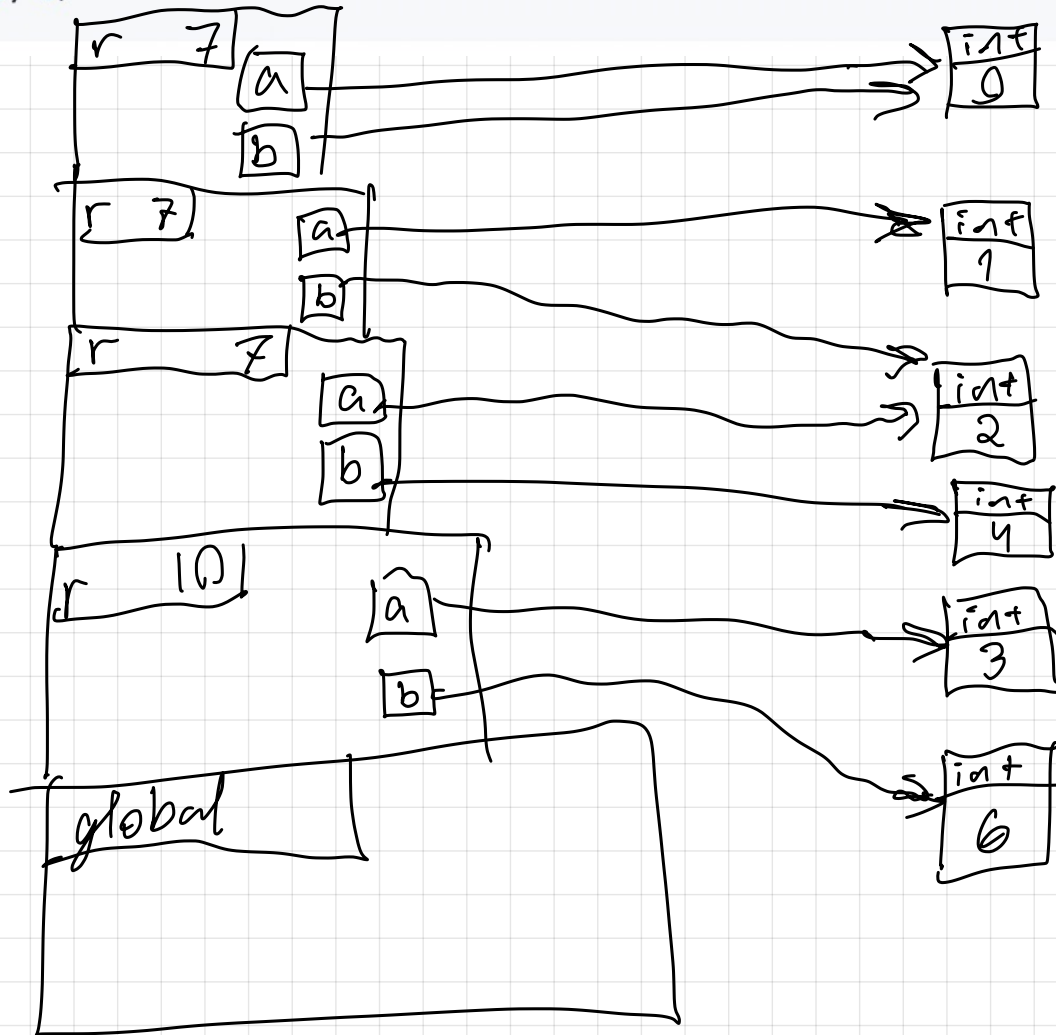
```



```

1 def r(a, b):
2     if a <= 0:
3         # Raden med kommentar
4         return 1
5     if b <= 0:
6         return 2
7     return r(a-1, b-2)
8
9
10 r(3, 6)

```



return 1

r(0, 0)

r(1, 2)

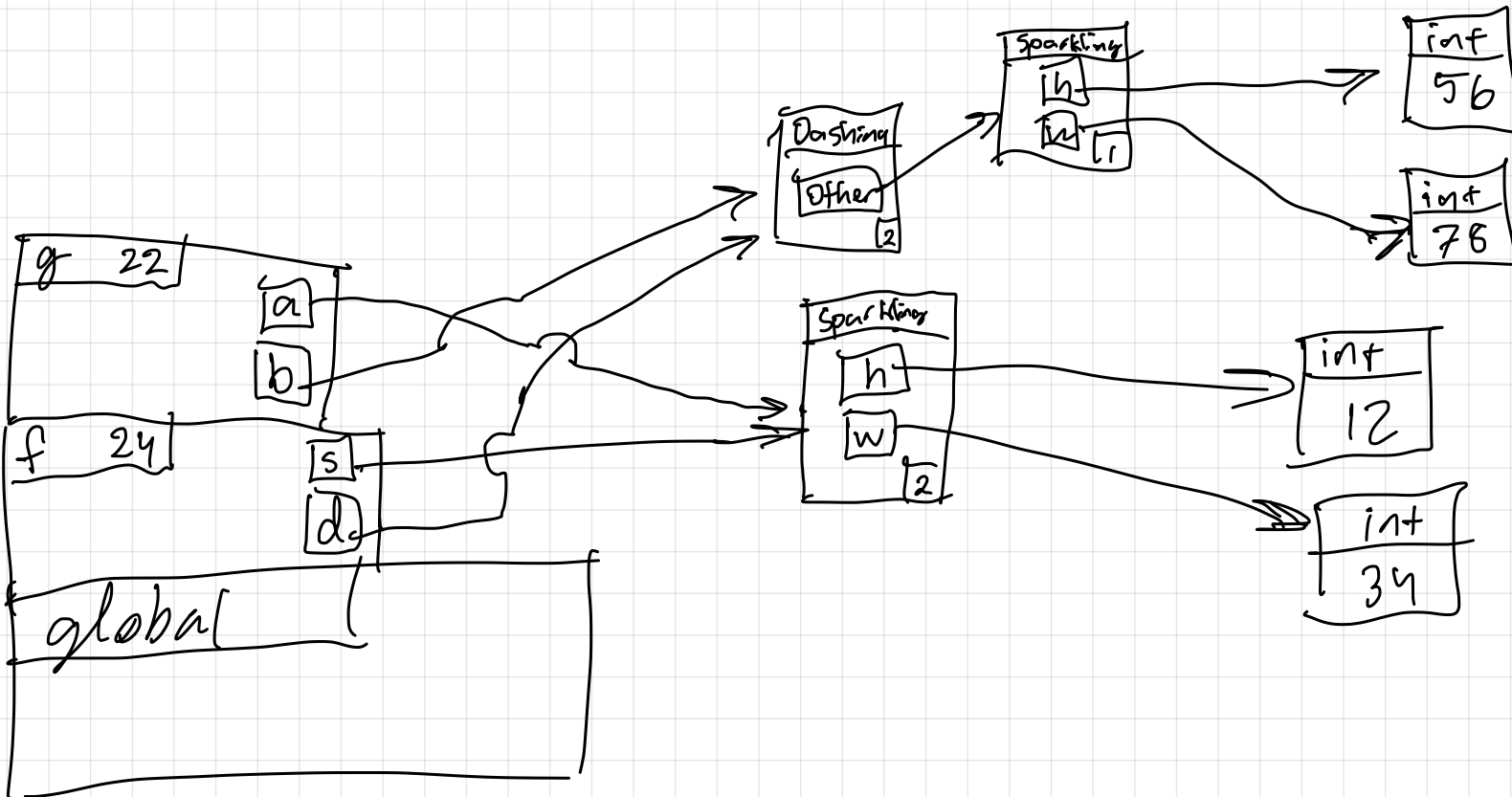
r(2, 4)

r(3, 6)

```

1 class Sparkling:
2     def __init__(self, height, width):
3         self.height = height
4         self.width = width
5
6 class Dashing:
7     def __init__(self, other):
8         self.other = other
9
10
11 def g(a, b):
12     b.other = Sparkling(56, 78)
13     # Raden med kommentar
14     return
15
16 def f():
17     s = Sparkling(12, 34)
18     d = Dashing(s)
19     g(s, d)
20
21 f()

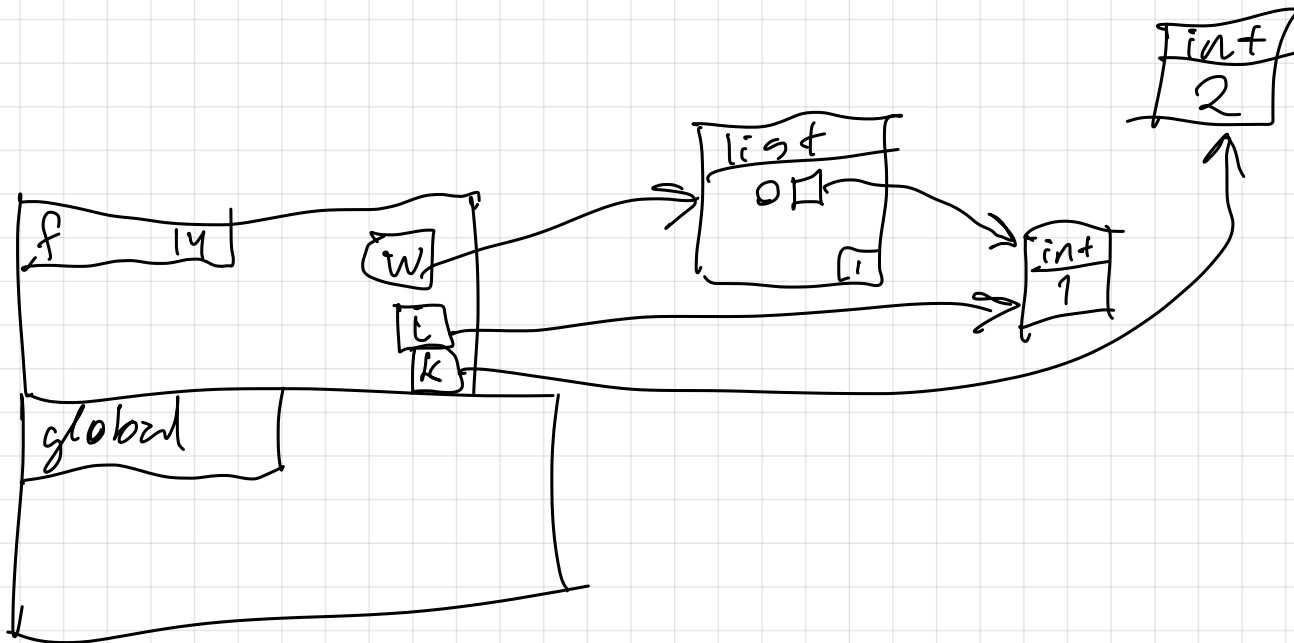
```



```

1 def g(v, i):
2     v.append(i)
3     i += 1
4     return i
5
6
7 def f():
8     w = []
9     i = 1
10    k = g(w, i)
11    # Raden med kommentar
12
13
14 f()

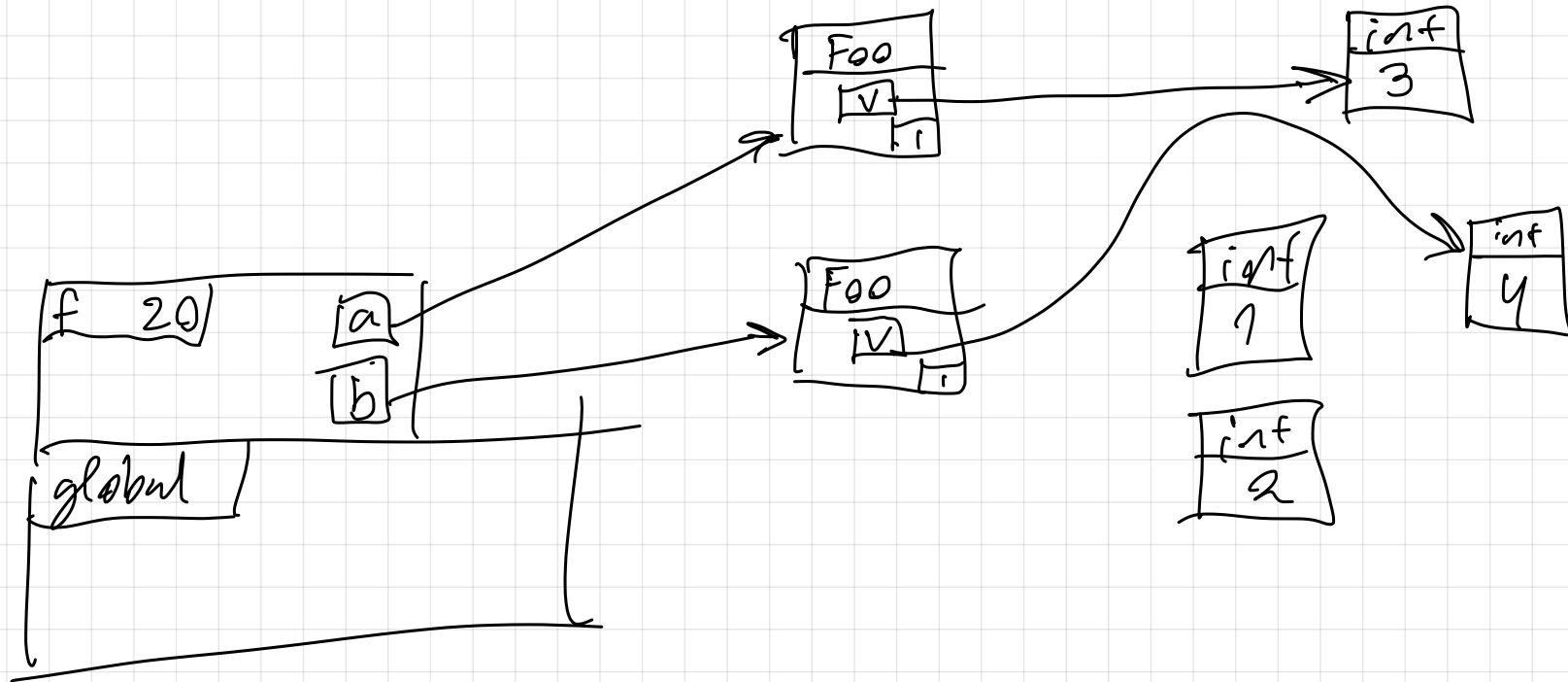
```



```

1 class Foo:
2     def __init__(self, value):
3         self.value = value
4
5
6 def g(a, b):
7     a.value = 3
8     b.value = 4
9     return
10
11
12 def f():
13     a = Foo(1)
14     b = Foo(2)
15     g(a, b)
16     # Raden med kommentar
17     return
18
19
20 f()

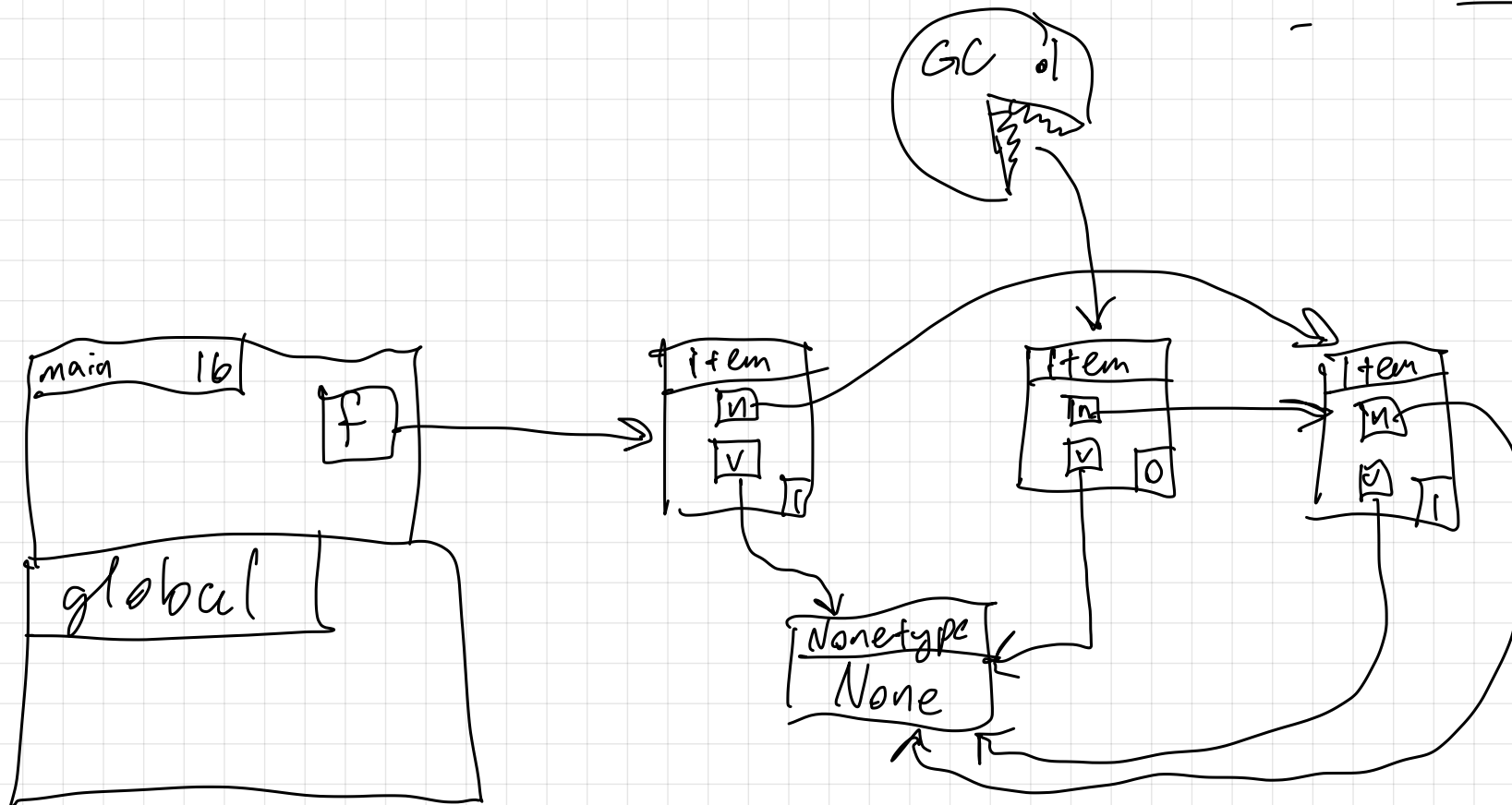
```



9. Från exempelenta:

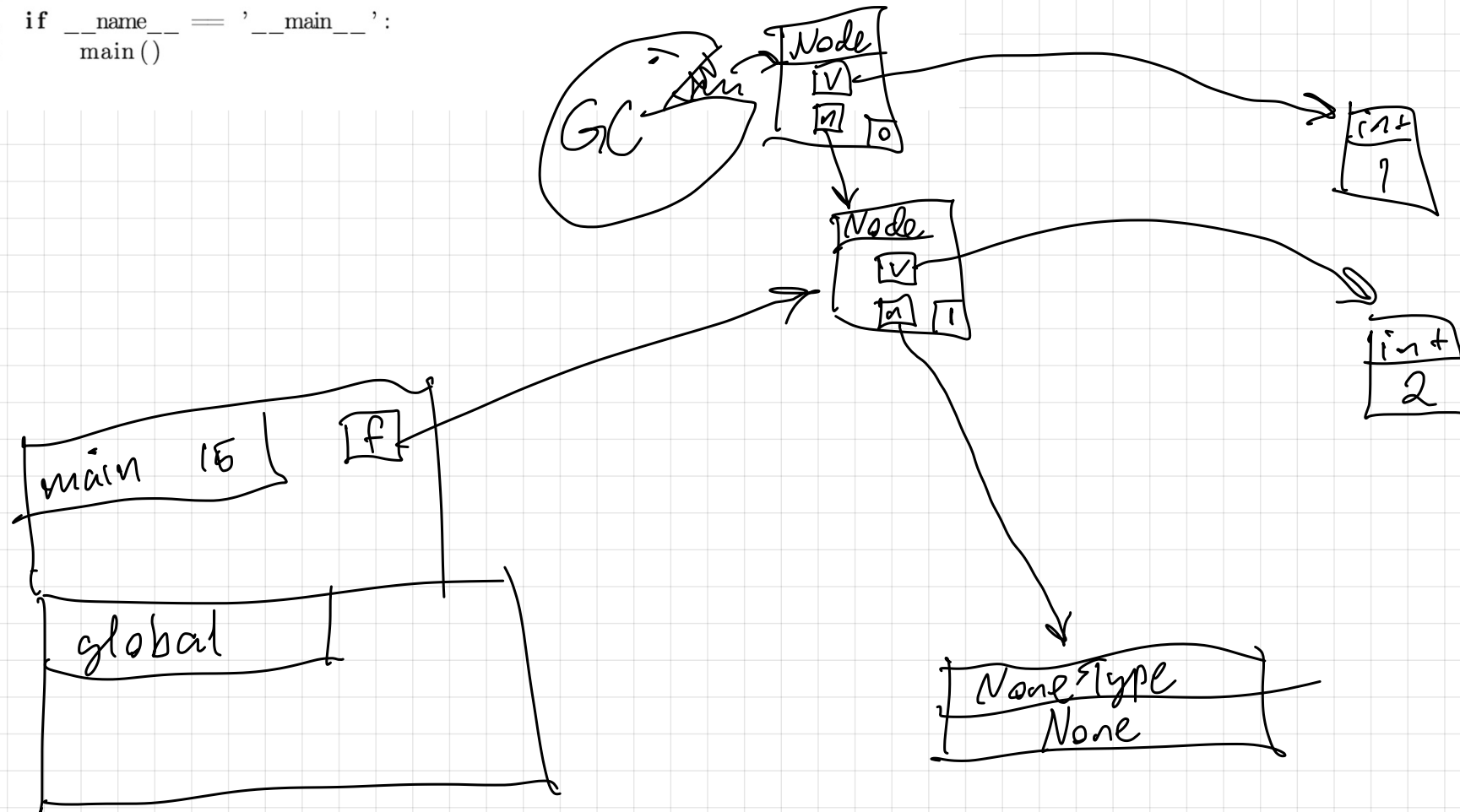
```
1. class Item:
2.     """A singly linked list."""
3.     def __init__(self, value=None, next=None):
4.         self.value = value
5.         self.next = next
6.
7.
8. def main():
9.     first = Item()
10.    first.next = Item()
11.    first.next.next = Item()
12.    first.next = first.next.next
13.    # Rita hur minnet ser ut här
14.
15.
16. main()
```

Class Node:
pass



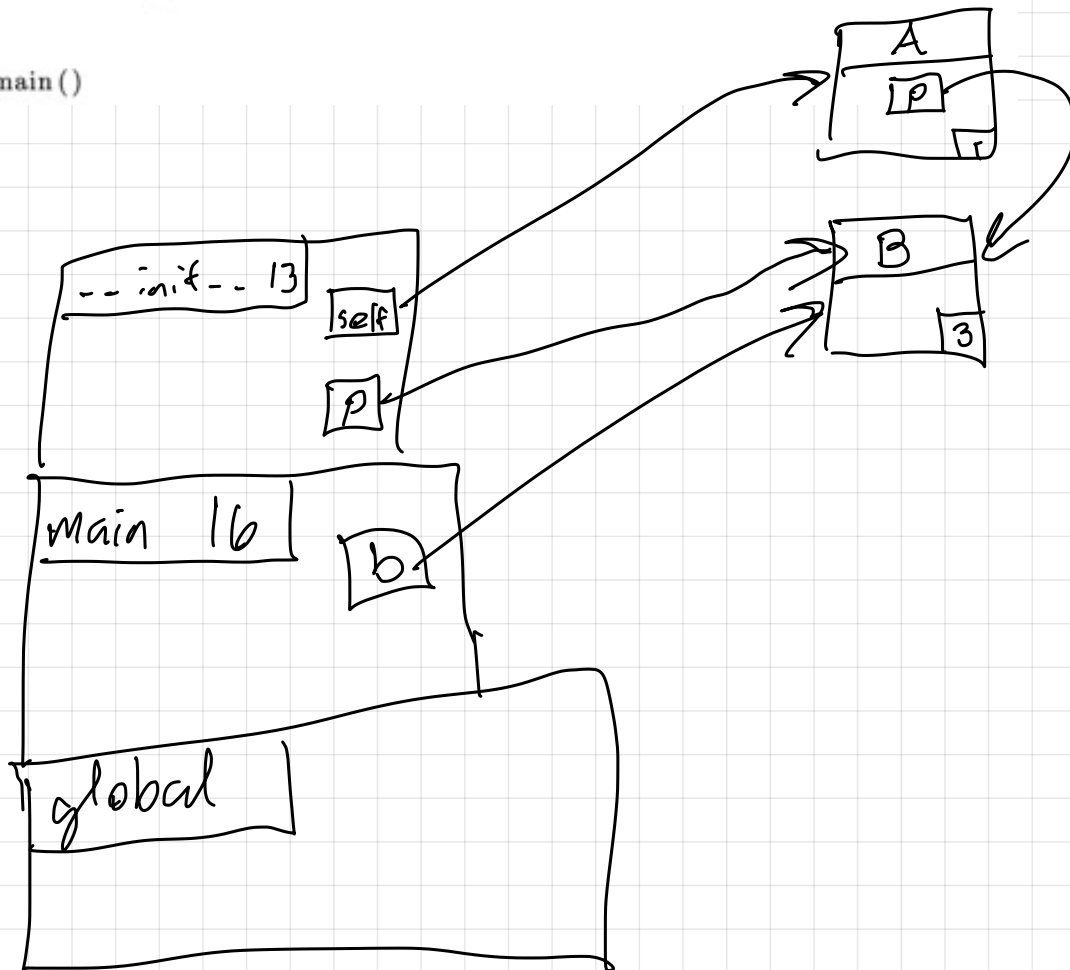
30. Rita ett låd- och pildiagram över minnet då programkörningen når den kommenterade raden.

```
1 class Node:
2     def __init__(self, v, n=None):
3         self.v = v
4         self.n = n
5
6
7 def main():
8     first = Node(2)
9     first = Node(1, first)
10    first = first.n
11    # Den kommenterade raden.
12
13
14 if __name__ == '__main__':
15     main()
```



28. Rita ett l d- och pildiagram  ver minnet d  programk rningen n r den kommenterade raden.

```
1 class A:
2     def __init__(self, parent):
3         self.parent = parent
4         # Den kommenterade raden.
5
6
7 class B:
8     pass
9
10
11 def main():
12     b = B()
13     a = A(b)
14
15
16 main()
```



Vad tar GC inte? (Vad behöver inte referensräkning?)

- Funktioner i stacken
 - Integers i intervallet $[-5, 255]$
 - Booleans: True och False
 - NoneType: None
 - Pekare (pilar)
- } (Finns alltid i minnet)
-

Vad tar GC (Inte komplett lista)?

- Listor
- Egenskapade objekt
- Strings
- tuples
- Dictionaries

Ska det garbage collectas?

