Velkommen til nittende gruppetime!

Gruppe 8

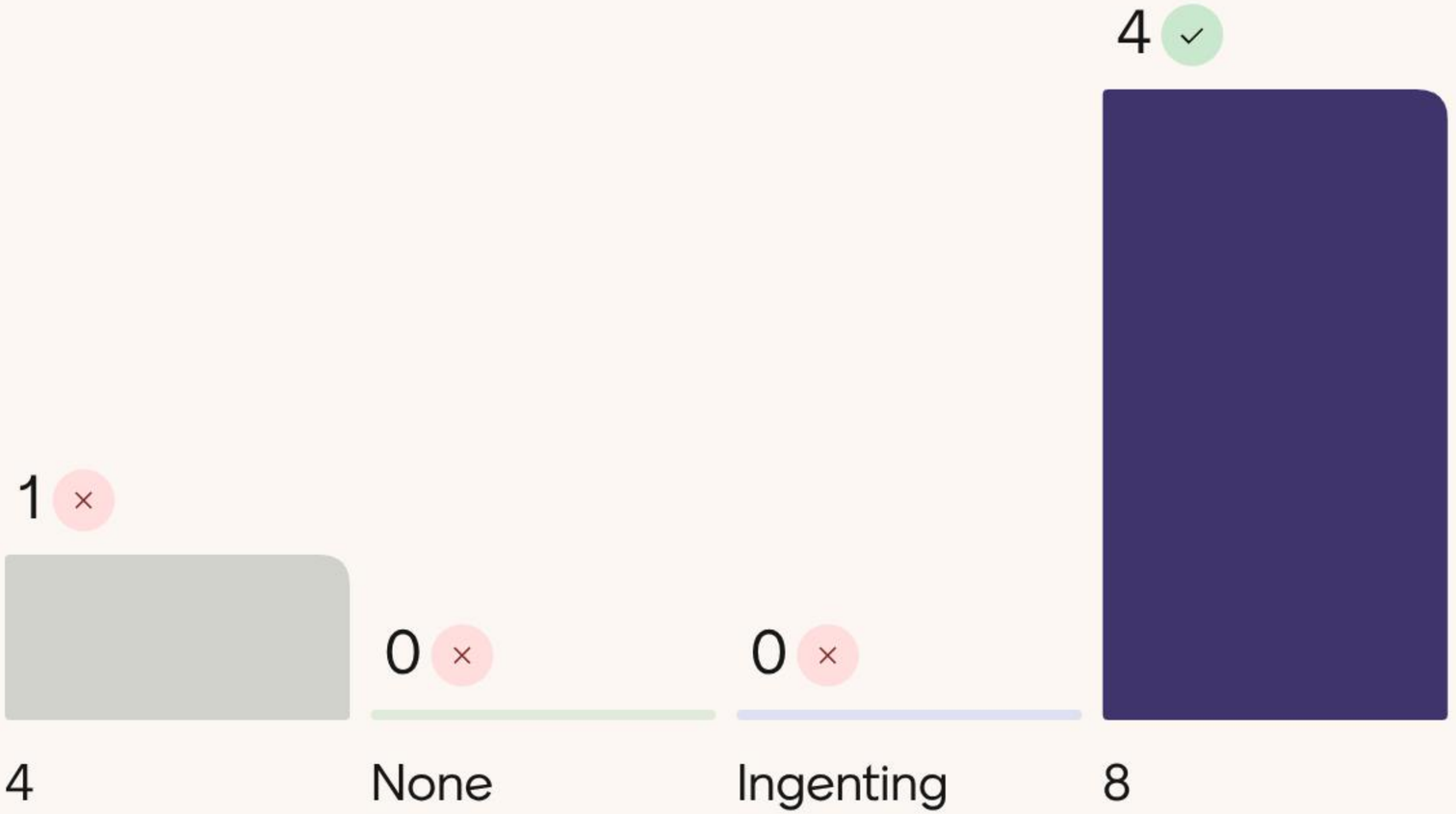
Uke 10: Datastrukturer



!!! Quiz !!!



Hva skrives ut?




```
def dobbel(x):
    return x * 2

tall = dobbel(4)
print(tall)
```

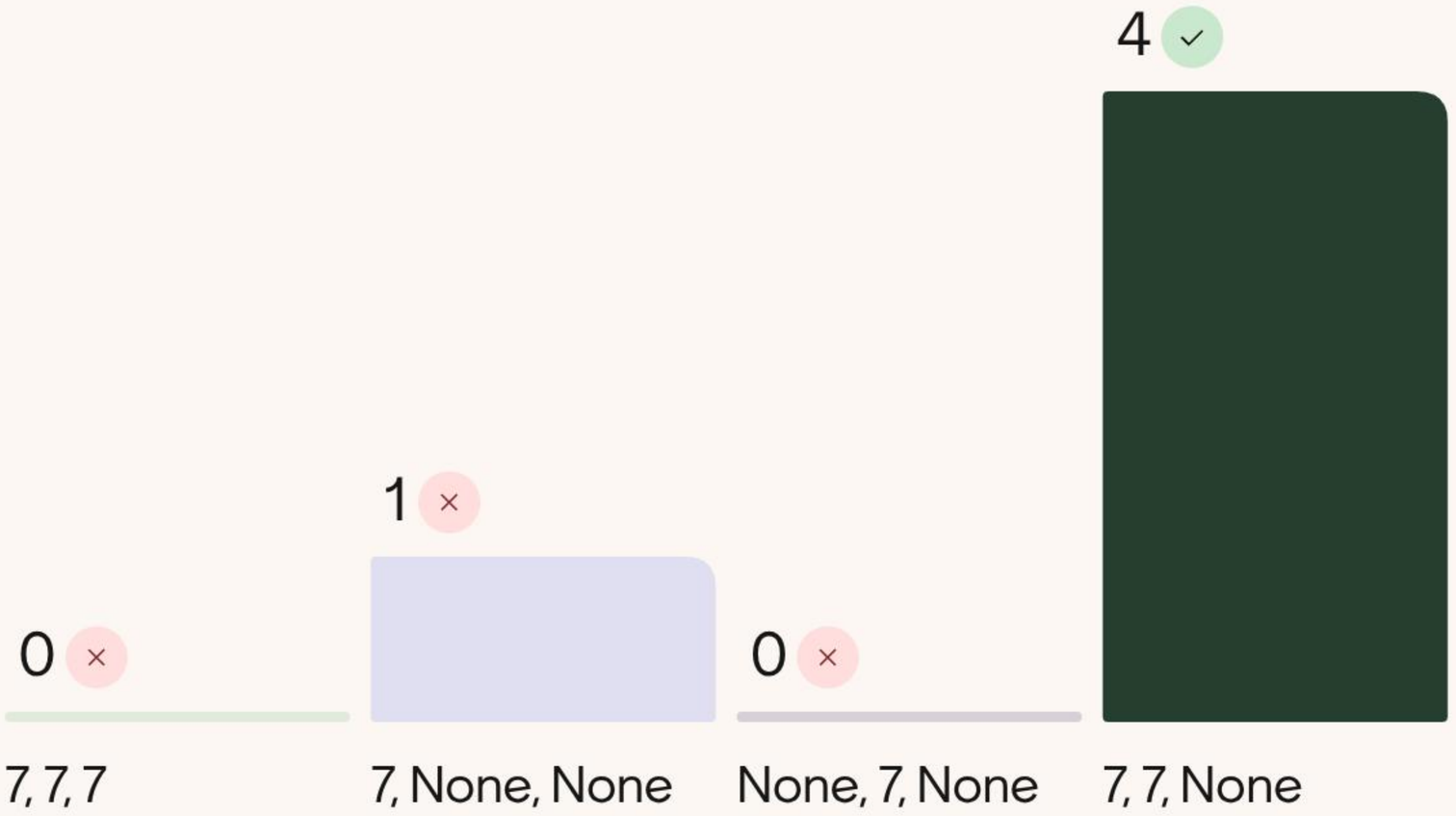
Hva ville skjedd dersom vi fjernet "return x*2" og erstattet den med "print(x*2)"?

```
def dobbel(x):
    return x * 2

tall = dobbel(4)
print(tall)
```



Hva skrives ut?




```python
def beregn_sum(a, b):
    return a + b

def skriv_sum(a, b):
    print(a + b)

x = beregn_sum(3, 4)
y = skriv_sum(3, 4)

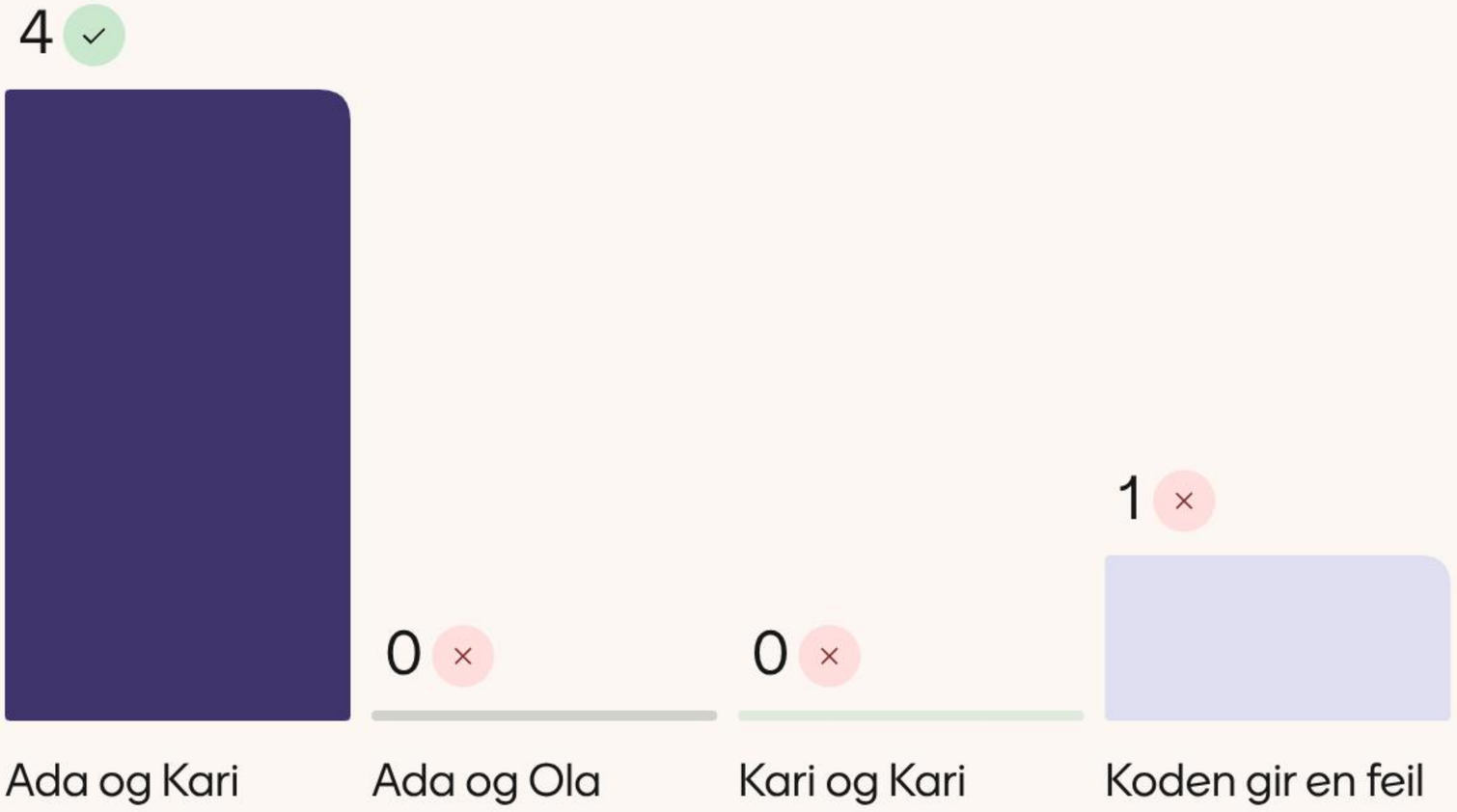
print(x)
print(y)
```

Funksjon og prosedyre

- **Funksjon**
  - Beregner eller henter en verdi.
  - Bruker `return` for å sende verdien tilbake.
  - Kan brukes videre i uttrykk og beregninger.
- **Prosedyre**
  - Utfører en handling, for eksempel å skrive ut informasjon eller endre en datastruktur.
  - Trenger ikke å returnere en verdi.
  - Returnerer `None` implisitt hvis ingen `return` -verdi er angitt.
- ***Diskuter:*** *Bør en metode som legger til en person i en liste, returnere en verdi eller bare utføre handlingen?*
- ***Fin huskeregel****: Står det "skriv en prosedyre som ..." skal den som regel ikke returnere noe, mens "skriv en funksjon som ..." betyr at den SKAL returnere noe.*

0

Hva blir a etter følgende kode kjøres?




```python
class Person:
    def __init__(self, navn):
        self.navn = navn

p1 = Person("Ada")
p2 = Person("Ola")

p2.navn = "Kari"

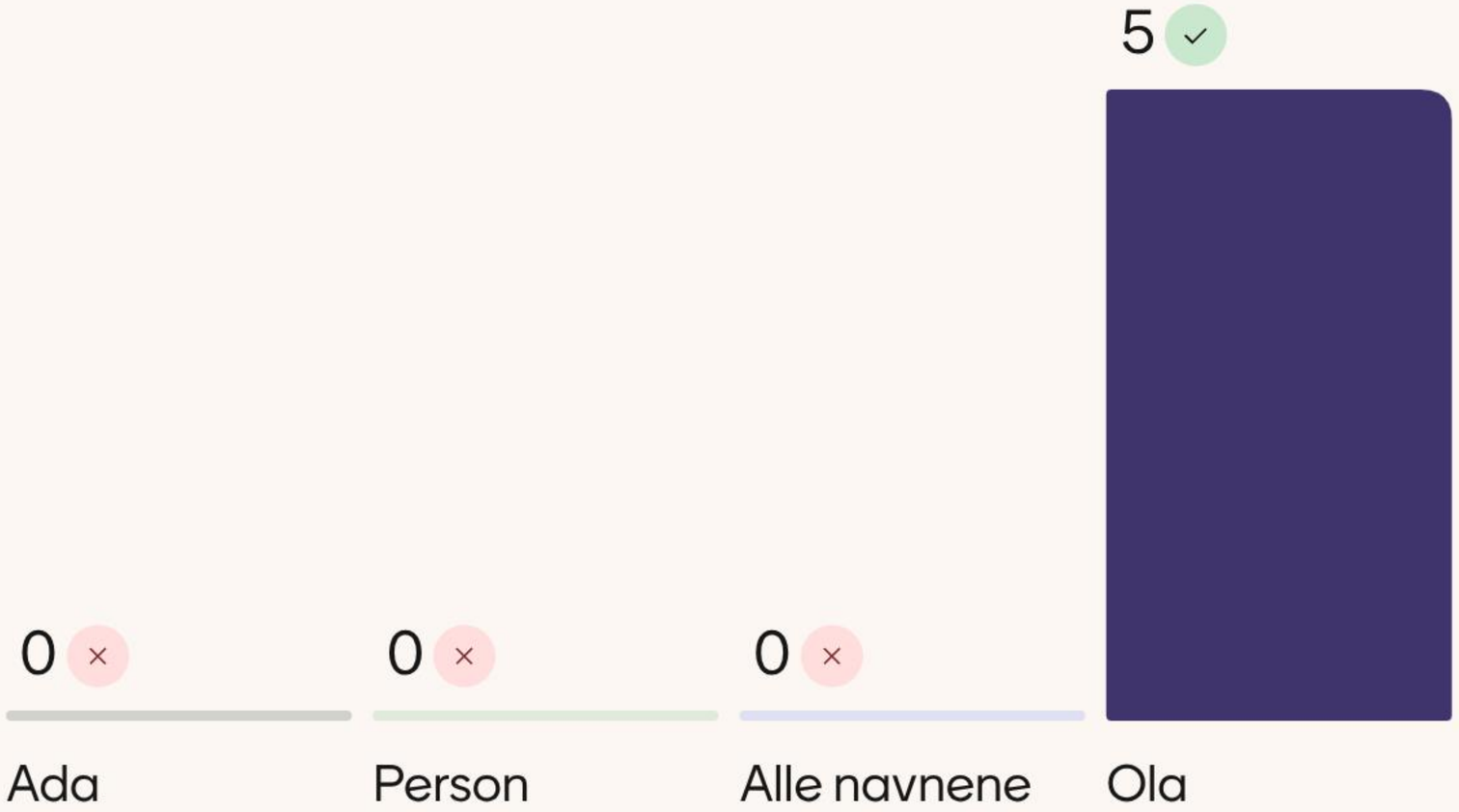
print(p1.navn)
print(p2.navn)
```

Hva skrives ut?

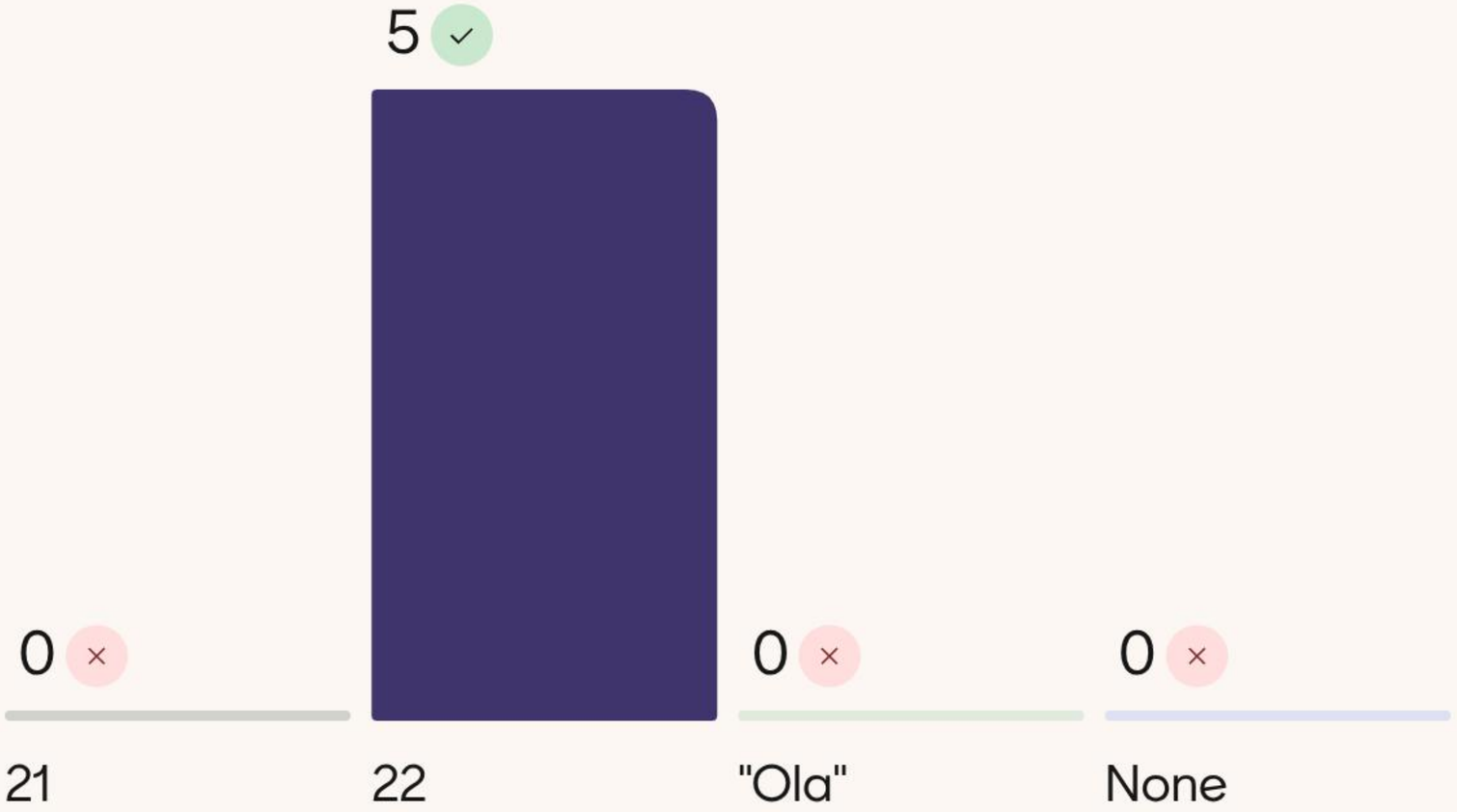
```python
class Person:
    def __init__(self, navn):
        self.navn = navn

personer = [
    Person("Ada"),
    Person("Ola"),
    Person("Kari")
]

print(personer[1].navn)
```



Hva skrives ut?




```
personer = {
    "Ada": 23,
    "Ola": 21,
    "Kari": 25
}

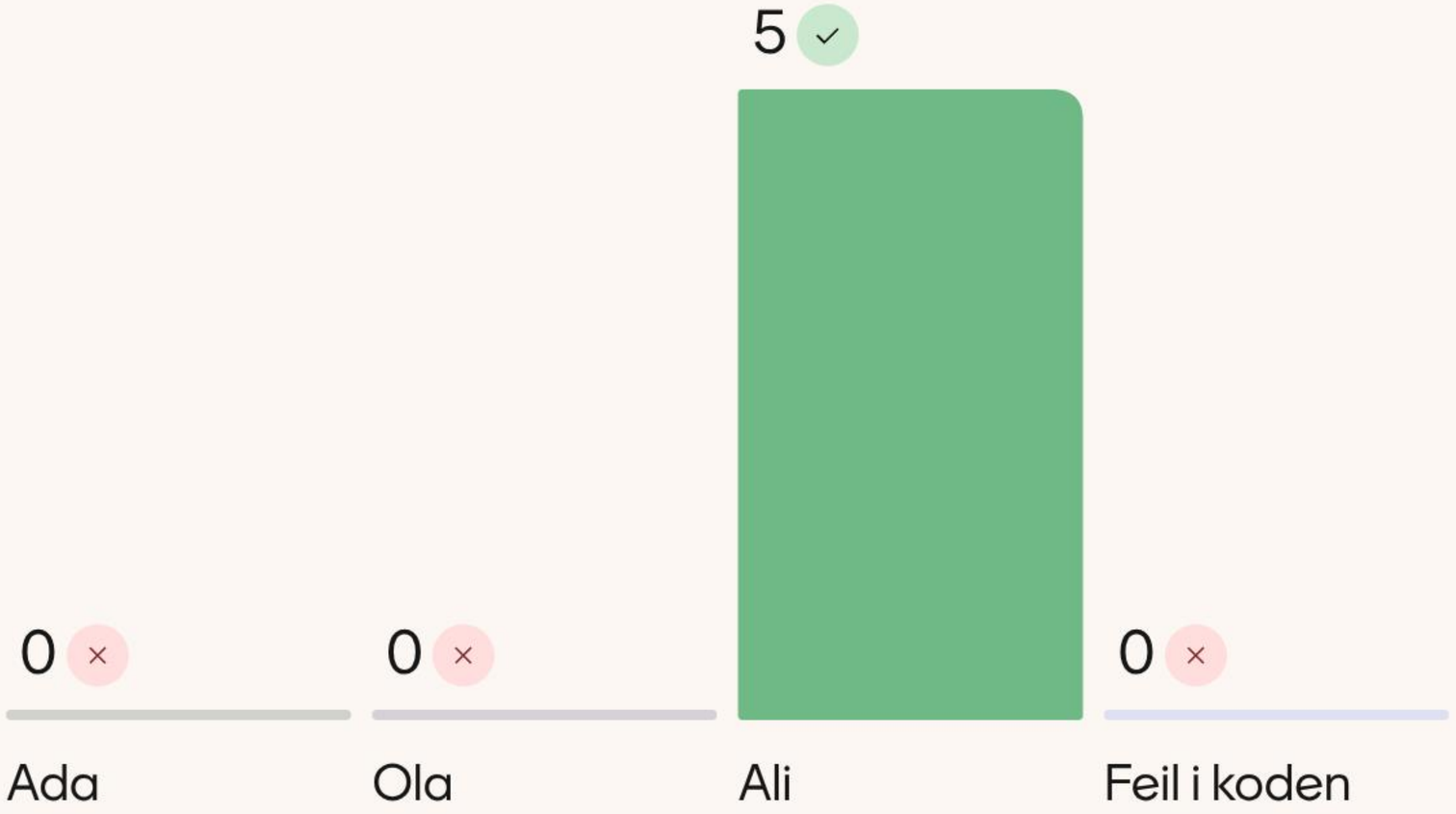
personer["Ola"] = 22
print(personer["Ola"])
```

Hva skrives ut?

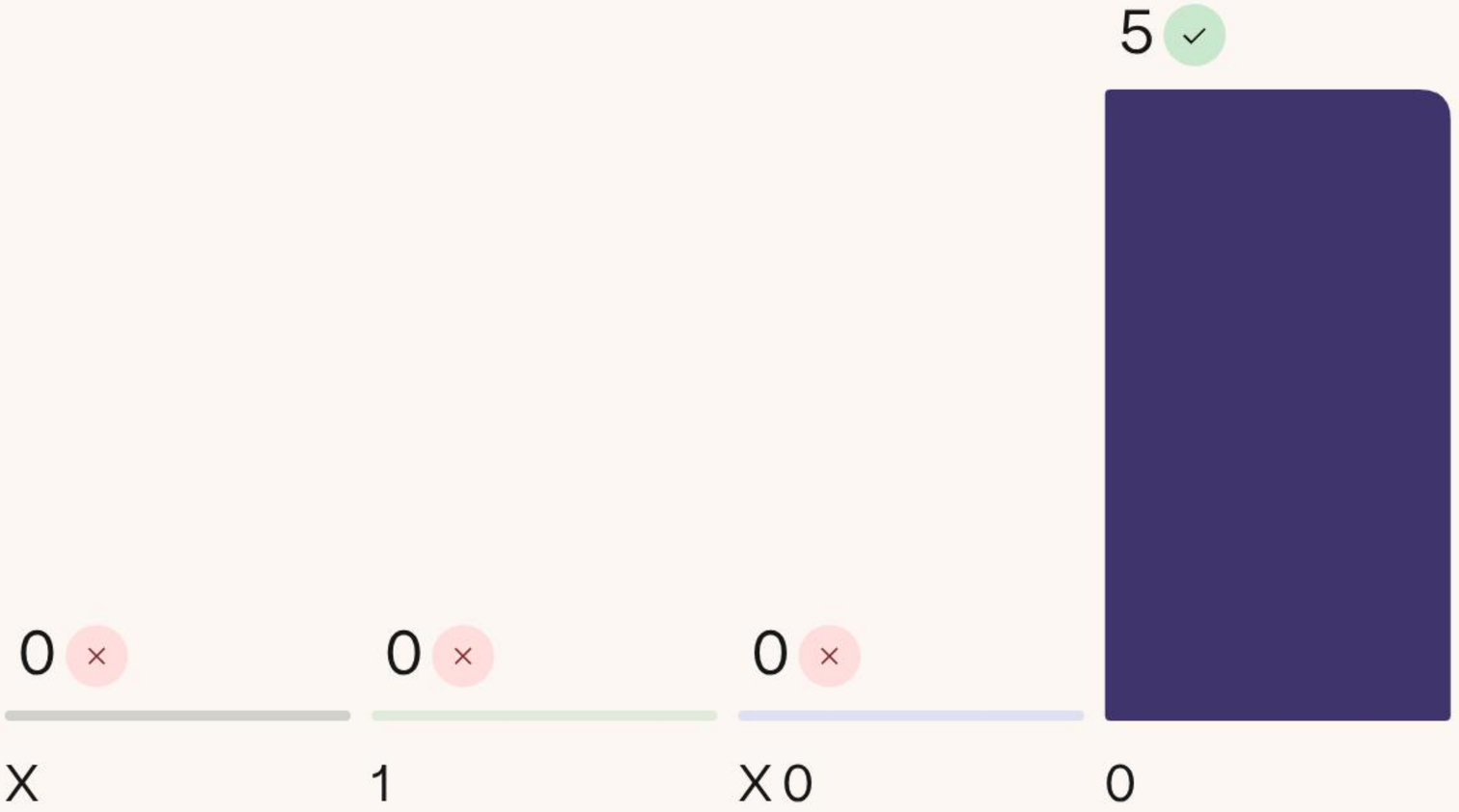
```python
grupper = {
    "gruppe1": ["Ada", "Ola"],
    "gruppe2": ["Kari", "Per"]
}

grupper["gruppe1"].append("Ali")

print(grupper["gruppe1"][2])
```



Hva skrives ut?




```
brett = [
    ["X", "0", "X"],
    ["0", "X", "0"],
    ["0", "0", "X"]
]

print(brett[1][2])
```

Ordbøker, objekter og referanser

- **Ordbøker**
  - Kobler nøkler til verdier.
  - Verdiene kan være tall, strenger, lister eller objekter.
- **Objekter i samlinger**
  - Lister og ordbøker kan inneholde objekter opprettet fra klasser.
  - Vi kan hente attributter og kalle metoder på objektene.
  - Eksempel: `personer[0].navn`
- **Referanser**
  - Variabler og samlinger kan referere til objekter.
  - Flere variabler kan referere til det samme objektet.

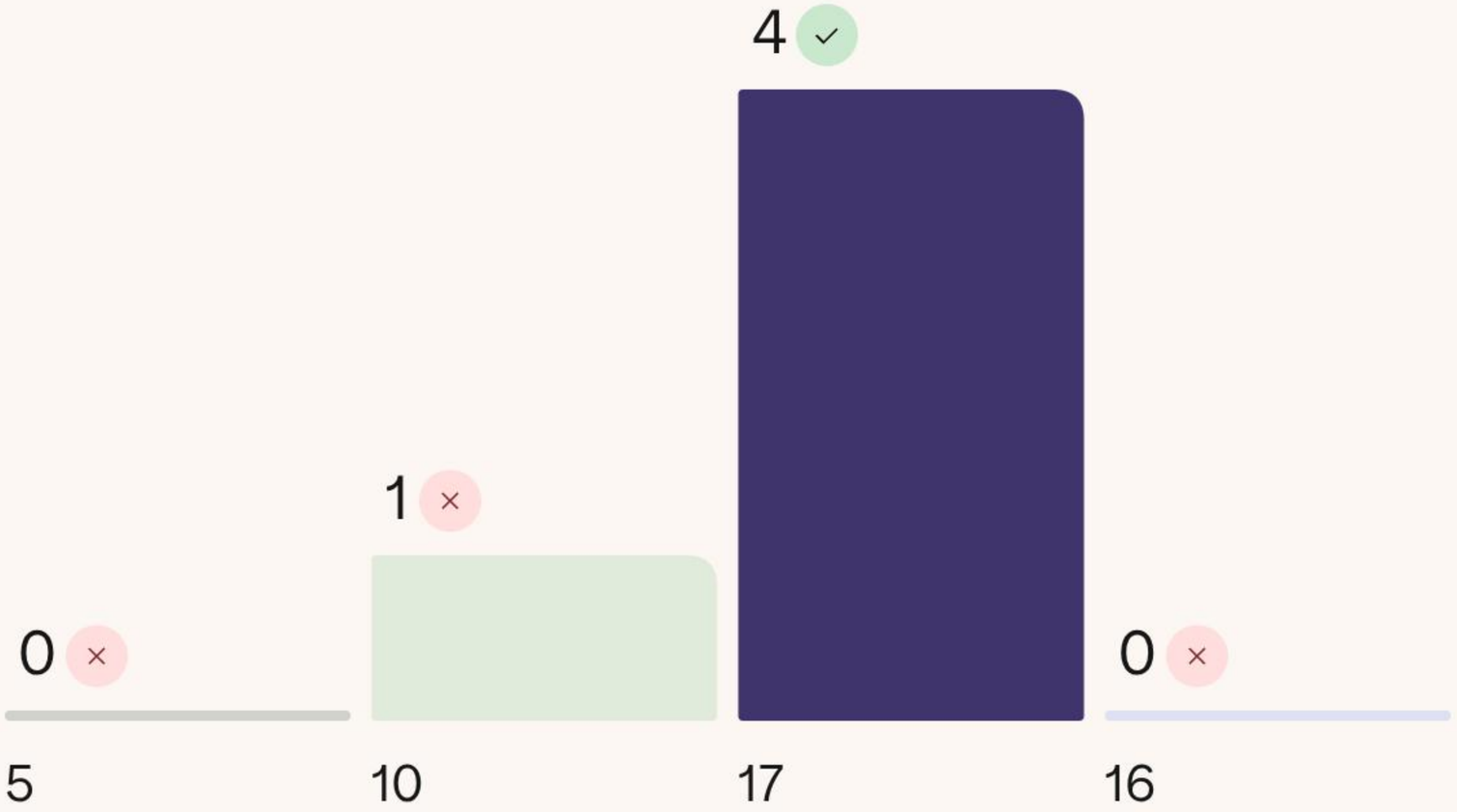
```python
class Person:
    def __init__(self, navn):
        self.navn = navn

grupper = {
    "gruppe1": [Person("Ada"), Person("Ola")],
    "gruppe2": [Person("Kari")]
}

print(grupper["gruppe1"][1].navn)
```

0

Hva blir a etter følgende kode kjøres?




```
a = 0

for b in [2,4,1]:
    a = 2*a + b
```

Hvilket tall skrives ut?

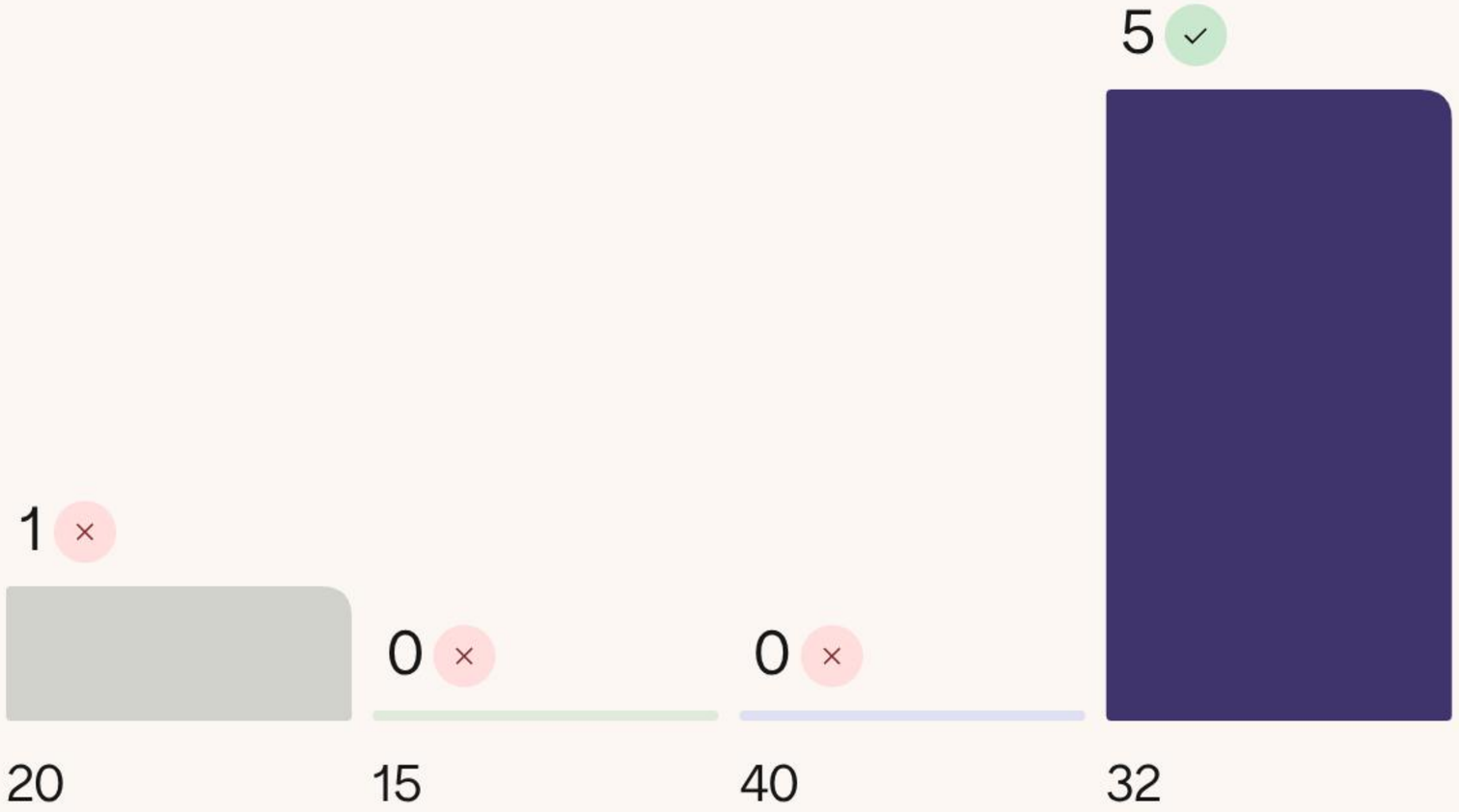
```
class Tall:
  def __init__(self, a):
    self._a = a

  def m1(self, c):
    self._a = self._a + c

  def m2(self):
    self._a = self._a * 2

  def m3(self):
    return self._a + 10

t1 = Tall(5)
t2 = Tall(2)
t1.m2()
t2.m1( t1.m3() )
print( t2.m3() )
```



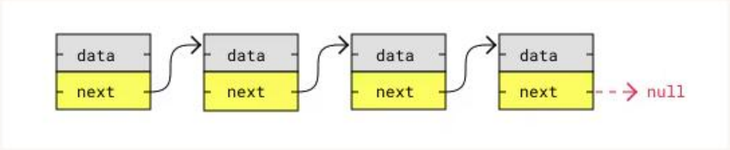
Lenkelister

- En lenkeliste består av noder som refererer videre til neste node.
- Hver node har typisk:
  - En verdi, for eksempel navnet på en holdeplass.
  - En referanse til neste node.
  - `None` som neste-referanse når noden er den siste.
- **Viktige begreper:**
  - **Første node**: starten på lenkelista.
  - **Neste-referanse**: koblingen mellom to noder.
  - **Traversering**: å gå gjennom lista fra node til node.
- En lenkeliste er ikke det samme som en vanlig Python-liste.
- Jernbanetorget → Stortinget → Nationaltheatret → `None`

0

Lenkelister (2)

- Hvordan ser lenkelista ut i minnet?
- Hva skjer hvis vi mister referansen til første node?
- Hva skjer hvis vi setter neste-referansen i første node til `None`?
- Hvorfor må vi passe på referansene når vi fjerner en node?



0

Bondesjakk: Eksamensoppgave!

- Nå skal dere jobbe med bondesjakk (tre på rad).
- Målet er å løse oppgaven ved å bruke det dere kan om:
  - Klasser og objekter
  - Instansvariabler og metoder
  - Lister og flerdimensjonale strukturer
  - Referanser mellom objekter
  - Testing og feilsøking
- ***Tips:*** *Tegn datastrukturen hvis dere er usikre på hvilke objekter som refererer til hva.*

  Oppgaven finner dere her: https://www.uio.no/studier/emner/matnat/ifi/IN1000/h26/grupper/gruppe-8/Ressurser/10_bondesjakk.pdf

0

Ferdig med bondesjakk-oppgaven?

- Hvis du synes oppgaven er enkel, tenk på mulige utvidelser.
- Du kan også jobbe med:
  - Trix eller innleveringen
  - Oppgaver fra tidligere gruppetimer
  - Eksamensoppgaver fra tidligere år
- Hvis alt føles enkelt, prøv et større prosjekt som bruker flere deler av IN1000-pensum.
- Velg gjerne noe du er interessert i, og bygg det på en ryddig måte slik at du kan utvide det senere.

0