Informasjon

Prøveeksamen i IN1000 høsten 2020

Alle hjelpemidler er tillatt (lærebok, nettressurser, notater osv.)

Det er *ikke tillatt* å samarbeide eller kommunisere med andre under eksamen om oppgavene.

Man kan trekkes ut til samtale for å kontrollere eierskap til sin besvarelse, jf. https://www.mn.uio.no/om/hms/koronavirus/kontrollsamtale/. Samtalen har ikke innvirkning på sensuren/karakteren, men kan lede til at instituttet oppretter fuskesak. Les mer om hva som regnes som fusk på UiOs nettsider: https://www.uio.no/om/regelverk/studier/studier-eksamener/fuskesaker/

Dette oppgavesettet inneholder oppgaver som gir totalt 120 poeng. Eksamen vil inneholde oppgaver som gir totalt 100 poeng.

1(a) Oppgave 1a)

```
tall1 = 1+2+3
tall2 = 2
tall1 =
```

Skriv et uttrykk i svarfeltet på høyresiden av siste tilordning som gir **tall1** verdien **11**. Uttrykket skal inkludere **tall1**, **tall2** og tallet **1,** men kun én gang hver**.**

Maks poeng: 1

1(b) Oppgave 1b)

```
tall = 4*2
t1 = "a"
if tall>9:
    t1 = t1 + "a"
print(          )
```

Fyll ut det som mangler for at denne koden skal printe ut strengen **ba** .
Det du fyller ut skal inkludere bruk av variabelen **t1**.

Maks poeng: 1

1(c) Oppgave 1c)

Fyll ut en logisk (boolsk) operator i svarfeltet som gjør at denne koden skriver ut strengen **ja** .

```
tall1 = 2*3
tall2 = 2**3
if (tall1>7)          (tall2>7):
    print("ja")
else:
    print("nei")
```

Maks poeng: 1

1(d) Oppgave 1d)

Hvilke verdier må ligge i variabelen **liste** ved starten av denne koden for at hele koden skal kjøre (for at **liste** skal ha lengde 3 og for at **tall** skal ha verdien 12 i siste linje)?

Verdiene i liste skal være positive heltall større enn 0.

Skriv svaret som man skriver en python-liste, f.eks. **[1,1,1]** men med verdier i **liste** som gjør at koden kjører riktig.

```
#liste = ??
tall = 0
for i in liste:
    tall+=1
    tall=tall*i
assert len(liste)==3
assert tall==12
```

Maks poeng: 3

1(e) Oppgave 1e)

Fyll inn tallet som mangler for at tallet **30** skal skrives ut:

```
a=3
for i in range (0, [          ] ) :
   a = a*2
   for b in [1,2,3]:
      a += b
print(a)
```

Maks poeng: 1

1(f) Oppgave 1f)

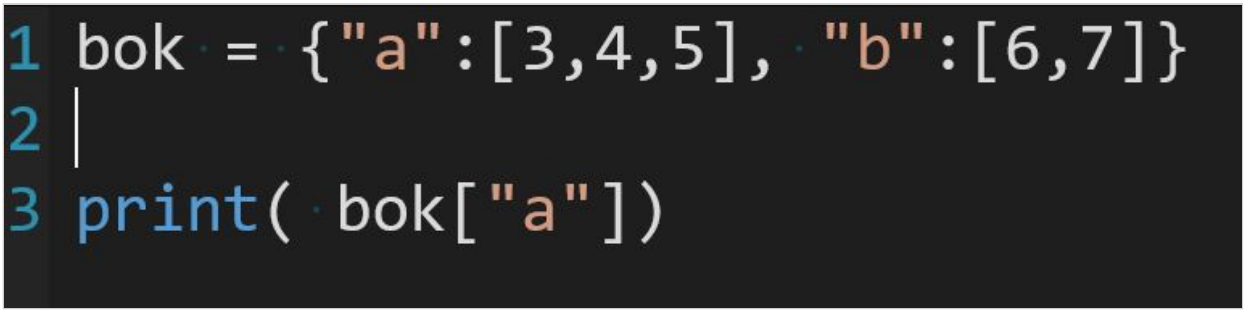
Noe mangler i uttrykket i linje 7 for at koden vil skrive ut tallet **2** til terminalen. Skriv det komplette uttrykket i svarfeltet.

```
tall = 0
operasjoner = "idd.d..did"
for operasjon in operasjoner:
    if operasjon == "d":
        tall = tall*2
    elif operasjon == "i":
        tall = tall                 # linje 7
    else:
        tall = 0
print(tall)
```

Skriv ditt svar her:

Maks poeng: 1

1(g) Oppgave 1g)



Det mangler en programsetning i linje 2 i vedlagte kode for at følgende skrives ut på terminalen: **[3, 4, 5, 8]**

Programsetningen skal inneholde et metodekall på **bok["a"]** .
Det skal altså ikke skal gjøres en tilordning til **bok["a"]** .

Skriv hele setningen som mangler i linje 2:

Maks poeng: 2

1(h) Oppgave 1h)

```
stater = {"AL":"Alabama", "AK":"Alaska", "AZ":"Arizona", "AR":"Arkansas", "CA":"California", \
         "CO":"Colorado", "CT":"Connecticut", "DE":"Delaware", "FL":"Florida", "GA":"Georgia", \
         "HI":"Hawaii", "ID":"Idaho", "IL":"Illinois", "IN":"Indiana", \
        "IA":"Iowa", "KS":"Kansas", "KY":"Kentucky", "LA":"Louisiana", \
        "ME":"Maine", "MD":"Maryland", "MA":"Massachusetts", "MI":"Michigan", \
        "MN":"Minnesota", "MS":"Mississippi", "MO":"Missouri", "MT":"Montana", \
        "NE":"Nebraska", "NV":"Nevada", "NH":"New Hampshire", "NJ":"New Jersey", \
        "NM":"New Mexico", "NY":"New York", "NC":"North Carolina", "ND":"North Dakota", \
        "OH":"Ohio", "OK":"Oklahoma", "OR":"Oregon", "PA":"Pennsylvania", \
        "RI":"Rhode Island", "SC":"South Carolina", "SD":"South Dakota", "TN":"Tennessee", \
        "TX":"Texas", "UT":"Utah", "VT":"Vermont", "VA":"Virginia", \
        "WA":"Washington", "WV":"West Virginia", "WI":"Wisconsin", "WY":"Wyoming", \
        "AS":"American Samoa", "DC":"District of Columbia", "FM":"Federated States of
Micronesia", "GU":"Guam", \
        "MH":"Marshall Islands", "MP":"Northern Mariana Islands", "PW":"Palau", "PR":"Puerto
Rico", \
         "VI":"Virgin Islands" \
         }

# statkoder er ei liste med nøklene i ordboka stater i den samme rekkefølgen som ovenfor.
# statkoder = ["AL", "AK", .... , "VI"]

maks = 0
ny = ""
for k in statkoder:
    navn = stater[k].split()
    if len(navn) > maks:
        maks = len(navn)
        ny = k
print(ny, stater[k])
```

Hva skrives ut her?

Maks poeng: 3

2(a) Oppgave 2a)

```
1 def funk(a, b):
2     
3
4 c = funk(2+3, 2) * 4
5 print(c)
```

Hva er det enkleste uttykket der parametrene a og b inngår, som kan returneres fra funksjonen i linje 3 for at vedlagte kode vil skrive ut tallet **40**? Ta med nøkkelordet **return**, for eksempel **return xxx**
(men erstatt **xxx** med korrekt uttrykk)

Skriv ditt svar her:

Maks poeng: 2

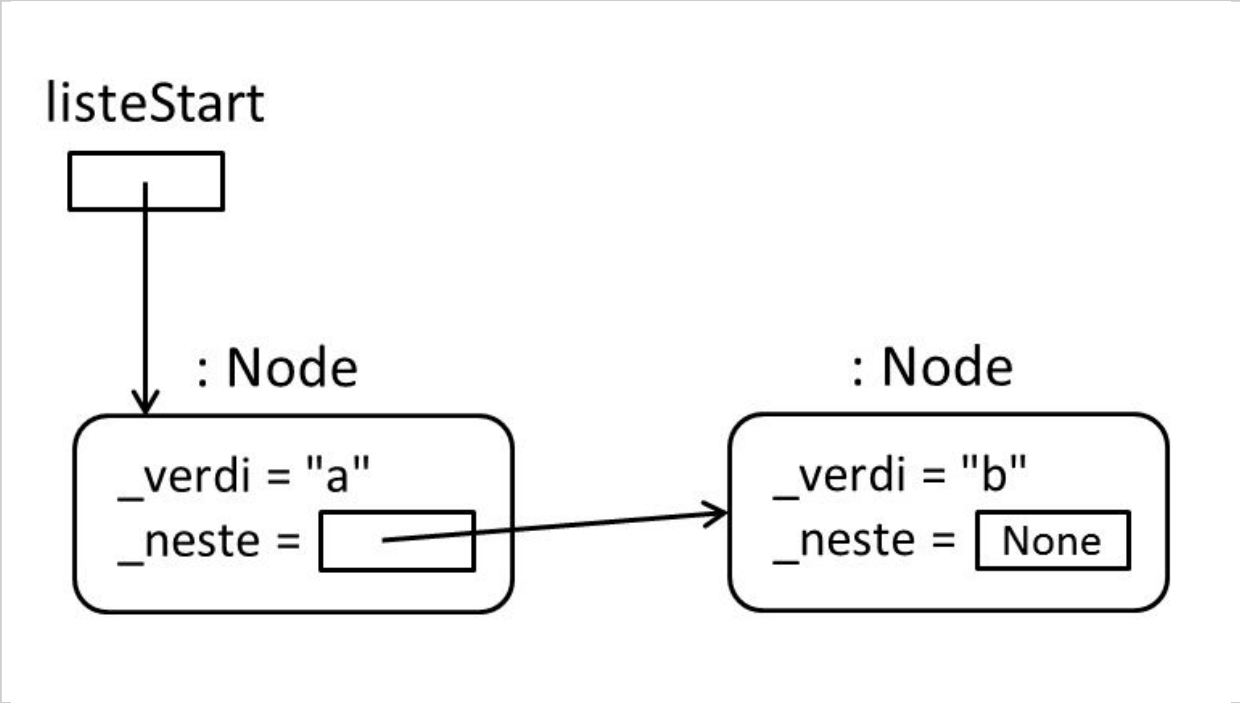
2(b) Oppgave 2b)

Hvilken ordning av føgende programsetninger vil gi et kjørende program som skriver ut 16 (og kun det)? Bare noen av programsetningene under skal brukes, og du kan bruke samme programsetning flere ganger i programmet for å oppnå ønsket funksjonalitet. Klasse- og metodedefinisjoner skal beholde rekkefølgen under.

```
a) class MyClass:
b)     def __init__(self, value):
c)     def change(self):
d)     def conclude(self):
e)         self._number = value
f)         self._value = value
g)         self._number = self._number * 2
h)         return self._number
i) number = 3
j) myObj.change()
k) print(myObj.conclude())
l)  myObj = MyClass(number+1)
m) myObj = MyClass(number)
```

Maks poeng: 4

2(c) Oppgave 2c)



Gitt klassen **Node** og et hovedprogram som vist nedenfor, oppgi programsetningen som mangler i metoden **settInn** for at et kall på **hovedprogram** vil opprette 2 objekter med verdiene "a" og "b" i en struktur som vist i figuren.

I figuren er en variabel (lokale og instansvariabler) vist som en firkantet boks, piler angir objektreferanser, og objektene er vist som rektangler med avrundede hjørner.
Variabelen **listeStart** refererer til objektet med verdi "a".

```
class Node :
    def __init__ (self, verdi) :
        self._verdi = verdi
        self._neste = None

    def settInn (self, ny) :


def hovedprogram():
    listeStart = Node("a")
    ny = Node ("b")
    listeStart.settInn(ny)
```

Skriv ditt svar her (kun den ene setningen som mangler i metoden **settinn**

Maks poeng: 5