KANDIDAT

294

PRØVE

INF100 0 Innføring i programmering

Emnekode	INF100
Vurderingsform	Skriftlig eksamen
Starttid	27.11.2023 15:00
Sluttid	27.11.2023 19:00
Sensurfrist	--
PDF opprettet	31.05.2024 13:16

Informasjon

Oppgave	Oppgavetype
i	Informasjon eller ressurser
i	Informasjon eller ressurser
i	Informasjon eller ressurser

Automatisk rettet

Oppgave	Oppgavetype
1(a)	Nedtrekk
1(b)	Fyll inn tekst
1(c)	Fyll inn tekst
1(d)	Fyll inn tall
1(e)	Fyll inn tekst
1(f)	Fyll inn tekst
1(g)	Fyll inn tekst
1(h)	Fyll inn tekst
1(i)	Fyll inn tekst
1(j)	Fyll inn tekst
1(k)	Fyll inn tekst
1(l)	Fyll inn tekst
1(m)	Fyll inn tekst

Forklaring

Oppgave	Oppgavetype
2(a)	Langsvar
2(b)	Langsvar

Kodeskriving

Oppgave	Oppgavetype
3(a)	Programmering
3(b)	Programmering
3(c)	Programmering
3(d)	Programmering

1(a)

```
a = [-1, 3.14, 'foo']
b = 'bar'
c = 2
d = '[woz]'
e = {
  'foo': 2,
  0: [0],
  1: 'foo'
}
```

Anta at kodesnutten over har blitt kjørt. Hvilken datatype (klasse) får uttrykkene under hvis de evalueres?

2 + 2	int (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
a	list (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
d	str (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
a[c]	ingen, det krasjer (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
a[2][2]	str (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
'wax' in d	bool (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
e[0]	list (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
e[-1]	ingen, det krasjer (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)

len(e) / 2	float (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)
c == (-2) * a[0]	bool (bool, dict, float, int, list, str, NoneType, ingen, det krasjer)

Maks poeng: 5

1(b)

```
a = [1, -1, 0, 2, 0, 4]
```

Gitt at koden over er kjørt.
Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

print(a[1])	-1
print(a[2 + 2])	0
print(a[3] + 1)	3
print(a[a[-1]-1])	2
print(a[a[a[0]]])	4

Maks poeng: 5

1(c)

```
d = {
    'foo': 1,
    'bar': 2,
    0: 'foo',
    1: 0,
    2: 'woz',
    -1: -2
}
```

Gitt at koden over er kjørt.
Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

print(d['foo'])	1
print(d[2][0])	"w"
print(d[d[0]])	1
print('bar'[2])	Error

Maks poeng: 4

1(d)

```
x = 10
y = 5
x = x + y
y = x - y
x += 1
print(x + y)
```

Hva skriver dette programmet ut?

26

Maks poeng: 2

1(e)

```
a = [1, 3, 0, 4]
r = []
for i in range(len(a)):
    r.append(i)
    r.append(a[i])
print(f'{r[4]} {r[5]}')
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

"2 0"

Maks poeng: 2

1(f)

```
a = [6, 4, 3, -2, 4]
x = 10
for e in a:
    if e < x:
        x -= e
print(x)
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

3

Maks poeng: 2

1(g)

```
def bar():
    x = 20
    return x
x = 10
y = x
y = bar()
print(x + y)
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

30

Maks poeng: 2

1(h)

```
def baz(x):
    x += 2
    print(x, end='')
x = 5
baz(x)
print(baz(x))
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

77None

Maks poeng: 2

1(i)

```
def quz(x, a):
    for i in a:
        x -= 1
    return x
p = 10
q = [10, 10, 10]
print(quz(p, q))
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

7

Maks poeng: 2

1(j)

```
def alpha(p, q):
    r = p + q
    s = beta(r + 1, q) + beta(p + 1, r)
    return s
def beta(t, u):
    v = t - u
    return v - 1
print(alpha(5, 2))
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

-4

Maks poeng: 2

1(k)

```
def echo(x):
    if x < 0:
        return 0
    if x >= 10:
        x = 10
    elif x % 2 == 0:
        x += 1
    if x < 5:
        x += 2
    else:
        x -= 1
    return x + 1
```

Gitt at funksjonen over er definert.
Hva skrives ut i de følgende setningene? (hvis programmet krasjer, skriv kun 'Error')

print(echo(0))	4
print(echo(5))	5
print(echo(10))	10
print(echo(echo(2)))	7

Maks poeng: 4

1(l)

Gitt at minnet har tilstanden vist over, hva blir skrevet ut etter setningen **print(a[1][1] + b[1][1])**?
(hvis programmet krasjer, skriv kun 'Error')

"deltafoxtrot"

Maks poeng: 2

1(m)

```
x = 999_8_22222_1_333_000_7777_66_44444444
b = 10
a = [0] * b
for i in range(b):
    t = x
    while t > 0:
        r = t % b
        if i == r:
            a[i] += 1
        t //= b
print(a[6])
```

Hva skriver dette programmet ut? (hvis programmet krasjer, skriv kun 'Error')

2

PS: Programmet vil ikke krasje på første linje (det er lov å inkludere understreker «_» når man oppgir tall i Python for å gjøre det lettere å lese tallet).

Maks poeng: 4

2(a) Tidemann holder på med lab6 (Snake), men har gjort noe feil i steget der han skal tegne et rutenett. Når test-programmet hans *view_test.py* kjører, vises programmet til venstre, selv om han egentlig skulle ønske at det så ut som programmet til høyre. Det kommer ingen feilmeldinger i terminalen.

Hva har Tidemann gjort feil? Les koden hans til venstre, og forklar:

- hva han har gjort feil,
- hvorfor feilen fører til oppførselen som vises, og
- hva han kan gjøre for å rette feilene.

Skriv ditt svar her

Første feilen jeg ser han har gjort er i get_color(value). De rutene som er ment å være "lightgray" og har verdien null blir først grå når ifsetningen if value== 0: color = "ligthgray". Men på linjen under så overskriver if value>= 0: color = "orange". så for å fikse det må tideman skive if value > 0: color = "orange".

Så er det en feil på linje 11. Der står det for col in range(rows) som gjør at brettet blir 6x6 istedenfor 6x8 som ønsket. Dette må bli skrevet om til for col in range(cols).

på linje 22 er det enda en feil. Denne gjør at if debug_mode bare blir skjekket hver gang row itererer og ikke når både row og col itererer. Dette gjør at du bare får debugmode på den sise rekken. for å fikse denne feilen så må Tideman putte if debug_mode etter hakk inn så den kommer under begge for loppene og ikke bare den ene.

Ord: 160

Maks poeng: 10

2(b) Othilie har lest seg opp om mengder og oppslagsverk, og har lagt merke til at de er *muterbare*. Men hva innbærer egentlig det? Hvilke hensyn må Othilie ta med muterbare verdier, som kanskje ikke ville vært nødvendig ellers?

Gi en forklaring til Othilie med illustrerende eksempler. Vi forventer ca 3-4 avsnitt, ikke mer enn 600 ord.

Skriv ditt svar her

At noe kan muteres kan tenkes på som at det kan endres, når vi ser på datatyper som int, str og bool så endrer vi de egentlig aldri, når skriver eks x=1 og senere x=2 så har vi laget et helt nytt objekt. at setts, dicts og lister er muterbare betyr at vi kan ender verdier uten å lage et helt nytt objekt. La oss si at vi har et set a = {1,2,3} så kan vi nå for å legge til fire bare skrive a.add(4). hadde vi skreve a = {1,2,3,4} så hadde vi endt opp med et helt bytt objekt. Det samme går for dicts. Har vi et dict b = {a:0,b:1,c:2} så kan vi legge til en ny key og element med b[d] = 3. eller så kan vi endre en av verdiene vi allerede har med b[a] = 4. nå vil det samme oppslagsverket se slik ut: {a:4,b:1,c:2,d:3}.

Det som kan bli litt skummelt når vi har muterbare objekter er at hvis vi skriver feks set_a = {1,2,3} og set_b = set_a. så kommer de nå til å peke på samme objekt. så la oss si du vil gjøre noe med set_b og så sammeligne den med set_a senere. set_b.add(4) og print(set_b.difference(set_a)). nå skulle man kanskje trodd at utskriften ble 4, men med muterbare objekter som sets så blir utskriften None. Det er siden de peker på samme objekt så når du skriver set_b.add(4) så blir objekter som bårde set_a og set_b peker på lagt til 4. så print(set_a) blir også {1,2,3,4}#set har ingen rekkefølge/er ikke indexsert så ikke alltid 1,2,3,4.

La oss ta et eksempel på et program som skal sammenligne hvor mye aksjene dine er verdt nå mot hvor mye de forventes å være verdt om n år. {aksje:markedsverdi i kr}

```
beholding = {APPL:20, KOG:50, LSG:40}
årlig_vekst = 0.1
fremtidig_beholdning = beholdning
for i in range(n):
    for key in behodning.keys():
        fremtidig_behodning[key] *= årlig_vekst
```

prøver vi nå sammenligne beholdning og fremtidig_beholdning så kunne man trodd at vi får en forskjell. men siden fremtidig_beholdning = beholdning så peker begge på samme objekt som gjør at orginalverdiene som vi hadde i beholdning er borte. så differansen mellom de to blir 0.

Ord: 364

Maks poeng: 10

3(a)

Skriv en kodesnutt slik at variabler og minnets tilstand for variablene a, b og c blir som vist over.

Skriv ditt svar her

Maks poeng: 4

3(b) Anta at **months_of_temps** er en variabel som peker på en to-dimensjonal liste av flyttall. De indre listene inneholder flyttall som representerer gjennomsnittstemperaturen for hver dag i en måned; mens den ytterste listen inneholder flere ulike måneder. Forskjellige måneder kan ha ulikt antall dager.

Skriv en funksjon **average_temp** med en parameter **months_of_temps** som beskrevet over. La funksjonen returnere gjennomsnittstemperaturen for alle dagene, uansett måned.

Eksempel på months_of_temps:

Hvis funksjonen du skriver kalles med eksempelet vist over som argument, skal returverdien bli 3.0: i eksempelet er det 11 ulike måleverdier (dager med temperaturmålinger), og summen av alle måleverdiene er 33.0.

Hint:

- tell hvor mange måleverdier som finnes,
- regn ut den totale summen av alle måleverdiene, og
- returner den totale summen delt på antall måleverdier.

Sensor vil vektlegge:

- Forståelse av løkker. Ikke bland indekser med verdier.
- Forståelse av funksjoner.
- Helhetlig algoritmisk forståelse.

Skriv ditt svar her

Maks poeng: 8

3(c) Oppgavetekst i PDF til venstre.
Skriv ditt svar her

Maks poeng: 20

3(d) Anta at den store Kvidoria har skrevet en ny episk roman i tekstfilen *story.txt*. Du jobber for forlaget, som er opptatt av et levende og variert språk. Redaktøren ber deg derfor om å lage et program som finner de 100 mest vanlige ordene i filen, samt hvor mange forekomster av disse ordene filen inneholder.

Hvilke hensyn må du ta for at løsningen skal fungere godt for redaktøren i praksis? *(Du skal ikke besvare dette som et forklaringsspørsmål, men la i stedet ditt svar på dette spørsmålet avgjøre hvilken funksjonalitet du inkluderer i programmet ditt.)*

Alle meningsfylte løsninger gir uttelling, men sensor vil vektlegge:

- anvendelighet for redaktøren,
- fornuftig valg av datastrukturerer, og
- hvor oversiktelig/lesbar koden er.

Skriv ditt svar her

Maks poeng: 10