

DA2004 Programmering för matematiker, 2019-04-17

- **Skriv tydligt.** Svårlästa svar riskerar 0 poäng.
 - Skriv bara på en sida av varje papper!
 - Tentan har flervalsfrågor där minst ett svarsalternativ är korrekt. Om man svarar fel eller inte har exakt antal rätta alternativ får man noll poäng på frågan.
 - Man måste bli godkänd på del A (5 rätt på 10 frågor) för att del B ska rättas.
 - **Hjälpmedel:** Ett A4 med så mycket information du vill. Du får skriva på båda sidorna.
 - **Betygsgränser:** E: 10, D: 12, C: 14, B: 16, A: 18, av maximala 20.
-

Del A: flervalsfrågor

Var snäll samla svaren på del A på ett svarpapper.

1. Vad är syftet med instruktionen `import`?

- Man läser in data från fil.
- Man hanterar särfall (eng: *exceptions*).
- Man avbryter sitt program.
- Man gör kod från en modul tillgängligt för ditt program.
- Inget av ovanstående.

2. Hur många punkter skrivs ut av följande kodsnuitt?

- 1
- 3
- 5
- 14
- 15
- 72

```
for x in 'seven':
    for y in 'two':
        print('.')
```

3. Vilket resultat skrivs ut av koden till höger?

- 3
- 4
- 6
- 7
- Ingenting, det blir ett särfall.

```
def compose(f, g):
    return lambda x: f(g(x))

f = compose(lambda x: 1 + x, lambda x: 2*x)
print(f(3))
```

4. Vad blir det för utskrift av koden till höger?

- 3
- 10
- 17
- 19
- Ingenting, det blir ett särfall.

```
x=17
def f(x, y):
    return x + y
y=-7
print(f(1,2))
```

5. Vad är syftet med `break`?
- Att skapa ett särfall.
 - Att stänga fil för läsning/skrivning.
 - Att avsluta en loop i förtid.
 - Att orsaka radbrytning efter en `print`-sats.
 - Inget, instruktionen finns inte i Python.
6. Följande konstruktioner användas för att skapa iteration över listor och liknande strukturer.
- `map`
 - `for`
 - `while`
 - `try-except`
 - `from-import`
7. Vilket eller vilka påståenden är sanna?
- Funktioner kan innehålla lokala funktionsdefinitioner.
 - Funktioner kan returnera tre värden i en `return`-sats.
 - Funktioner kan returnera värden tidigt med `except`.
 - Flera funktioner i samma fil kan ha samma identifierare.
 - Inget av ovanstående.
8. Vad är resultatet av satsen `print([2*x for x in range(3)])`?
- `[0, 1, 2]`
 - `[0, 2, 4]`
 - `[0, [0, 1], [0, 1, 2]]`
 - `[0, [0, 2], [0, 2, 4]]`
 - Ett särfall
9. Vilket eller vilka av följande skrivsätt är korrekta strängar i Python?
- `"Hubba"`
 - `'Hubba'`
 - `[Hubba]`
 - `(Hubba)`
 - `{Hubba}`
 - Inget av ovanstående.
10. Vilken eller vilka boolska variabeltilldelningar gör att uttrycket
- ```
(x or (y and z)) and (y or (x and z)) and (z or (x and y))
```
- evaluerar till `True`?
- `x=False, y=False, z=False`
  - `x=False, y=False, z=True`
  - `x=False, y=True, z=False`
  - `x=False, y=True, z=True`
  - `x=True, y=False, z=False`

## Del B: kodfrågor

11. Ge ett exempel på en sträng  $s$ , minst fyra tecken lång, som anropet `fun(s)` returnerar `True` för, och beskriv generellt vad funktionen beräknar.

*Tips:* index  $-1$  för en sträng (dvs, `s[-1]`) motsvarar sista tecknet i strängen. Man kan se det som ett kortare sätt att skriva `s[len(s)-1]`.

```
def fun(s):
 if s == '':
 return True
 elif s[0] == s[-1]:
 return fun(s[1 : -1])
 else:
 return False
```

12. Definiera funktionen `dropwhile` som tar två parametrar: ett predikat (en funktion som returnerar `True` eller `False`) som tar en parameter, och en lista. Funktionen ska returnera en kopia av listan, men utan de inledande element som gör predikatet sant. Till exempel, om vi har predikatet `even` som returnerar `True` för jämna tal och annars `False`,

```
>>> dropwhile(even, [0,2,4,1,2,3,4])
[1, 2, 3, 4]
```

och för ett enkelt lambda-uttryck:

```
>>> dropwhile(lambda x: x, [True, True, False, True])
[False, True]
```

13. På förra tentan fick man kod liknande den nedan, och uppgiften var att identifiera när det kunde uppstå fel. Problemet är ju att man frågar användaren efter indata och denna kanske inte alls är heltal, som det är avsett.

I denna uppgift ska du rätta till koden och göra den mer användarvänlig. Om man matar in ett korrekt heltal ska multiplikationstabell skrivas ut, precis som avsett, men om man skriver in något som inte går att tolka som ett heltal ska man få meddelandet "Bad input, try again or hit return to quit." Om man inte skriver in något alls, utan bara slår retur, då ska funktionen returnera `None`.

```
def mul_tabell():
 s = input('Vilken multiplikationstabell? ')
 for i in range(1,13):
 print(s, '*', i, '=', i*int(s))
```

14. Funktionen `polynomial_degree` tar en lista som antas representera ett polynom, med konstanttermen först och därefter termer av ökande grad, och returnerar polynomets gradtal. Antag att vi är oroliga att tomma listan kan dyka upp, representerande noll-polynomets vars gradtal inte är definierat. Att låta tomma listan representera ett polynom betraktas i vårt program som ett fel. Använd en teknik från defensiv programmering till att avbryta funktionen om funktionen anropas med tomma listan.

```
def polynomial_degree(p):
 return len(p) - 1
```

15. På förra tentan skulle man skriva en funktion `count_nucleotides(s)` som räknar antalet A, C, G, och T i strängen  $s$ , som antas representera DNA. Man fick anta att  $s$  bara innehåller tillåtna bokstäver, som versaler. Exempel på användning, inifrån Spyder:

```
In [1]: count_nucleotides('')
Out [1]: {'A': 0, 'C': 0, 'G': 0, 'T': 0}

In [2]: count_nucleotides('AACCAATT')
Out [2]: {'A': 4, 'C': 1, 'G': 0, 'T': 2}
```

Det fanns flera lösningar som liknade den nedan, men den lösningen ger inte helt rätt svar. Vad är problemet?

```
def count_nucleotides(s):
 counts = dict()
 for nucleotide in s:
 if nucleotide in counts:
 counts[nucleotide] += 1
 else:
 counts[nucleotide] = 1
 return counts
```

- 16.** Du har en opålitlig kollega som skrivit funktionen `rootfinder`. Den tar en funktion  $f(x)$  som parameter och returnerar sedan ett par  $(x, y)$ , som antingen är en rot till  $f$ , dvs  $f(x) = 0$ , och `None`, eller så är både  $x$  och  $y$  rötter.

Denna funktion är bara nästan vad du vill ha. Skriv en funktion `rootfinder_fixer` som tar en funktion som parameter, anropar `rootfinder` för att hitta minst en lösning, och därefter returnerar den rot som har störst absolutbelopp.

- 17.** Skriv en klass `Point` för att lagra punkter i planet. Dess konstruktor ska ta en  $x$ - och en  $y$ -koordinat.
- 18.** Visa hur man instantierar två `Point`-objekt till variablerna `p1` och `p2`.
- 19.** Skriv en metod `compare` till `Point` som jämför sitt objekt med ett annat `Point`-objekt.
- 20.** Visa hur man kan använda metoden `compare` med variablerna `p1` och `p2` från fråga 18.