



## TENTAMEN I PROGRAMMERING DI2006 OCH DT2018

**Datum:** 2021-04-07

**Tid:** 9.00–13.00

**Ansvarig lärare:** Eric Järpe (tel: 0729-77 36 26, email: eric.jarpe@hh.se)

### Anvisningar

- Tentan fotograferas och skickas in senast klockan 13.00 via SMS till 0729-77 36 26 eller via email till eric.jarpe@hh.se. Meddela tentavärdarna när du vill börja fotografera för att lämna in. Se till att ha papper i ordning inför fotograferingen av lösningarna. **Invänta besked från ansvarig lärare att lösningen mottagits** och meddela tentavärdarna detta innan utloggning från Zoom.
- Svaren skrivs i form av programkod som ska vara i så körbart skick som möjligt.
- Tillåtna hjälpmedel är
  - dator – **endast för att ta emot email och visa pdf-dokument**
  - formelsamling (som skickas via email)
  - valfri miniräknare
  - skrivpapper
  - penna
  - suddigummi
  - (linjal)
  - frukt, fika (inte för mycket)
- Till varje uppgift finns angivet hur många poäng som maximalt utdelas för uppgiften.
- Tentamen består av två delar: **Del 1** och **Del 2**.
- Del 1 består av 22 frågor och här kan man maximalt få 30 poäng.
- Del 2 består av 4 frågor och här kan man maximalt få 20 poäng.
- För betyg 3 krävs minst 12 poäng på Del 1. (Del 2 behöver inte alls göras för att få detta betyg.)
- För betyg 4 krävs minst 12 poäng på Del 1 och 5 poäng på Del 2.
- För betyg 5 krävs minst 12 poäng på Del 1 och 10 poäng på Del 2.

*LYCKA TILL!*

# Del 1

1. Vad innebär begreppet RAM i datorsammanhang – är det en slags mikroprocessor, grafikkort, arbetsminne, programmeringsspråk, procedur för problemlösning eller något annat? (1p)

2. Ange 2 rader Pythonkod som efterfrågar användarens namn, <namn>, och därefter skriver ut 'Hej på dig <namn>!'. (1p)

3. Man vill skriva ett 5 raders program som efterfrågar en täljare och en nämnare i ett bråk och returnerar hur bråket kan skrivas som största möjliga heltal plus ett bråk med en täljare och samma nämnare som angivits som indata. Exempel:

```
Programmet efterfrågar Täljare:
Användaren svarar 17
Programmet efterfrågar Nämnare:
Användaren svarar 3
Programmet svarar Bråket 17/3 kan skrivas som 5+2/3. (2p)
```

4. Man vill göra en tankningsapp. Betrakta följande Pythonkod.

```
while a not in [1,2,3]:
    a = int(input('Ange om du vill tanka\n
                1. Bensin\n
                2. El\n
                3. Gas\n
                genom att svara med respektive nummer. '))
    *
    b = int(input('Ok, hur många liter? '))
    elif b==2:
        b = int(input('Ok, hur många minuter? '))
    elif b==3:
        b = int(input('Ok, hur många kg? '))
    **
    print('Felaktigt val, försök igen!')
```

Ange vad som måste stå på platserna \* och \*\* för att programmet ska returnera: Ok, hur många liter? om det får indata 1

Felaktigt val, försök igen! om det får någon annan indata än 1, 2 eller 3. (2p)

5. Vilken inbyggd funktion i Python kan konvertera en variabel av typ int till en variabel av typ float? (1p)

6. Nämn ett exempel på ett programmeringsspråk som är ett lågnivåspråk. (1p)

7. Vad kallas ett informellt språk som inte har syntaxregler och inte är ämnat att kompileras eller exekveras? (1p)

8. Hur anger man en kommentar i ett Pythonprogram? (1p)

9. Vad returneras av koden

```
a = True
b = False
print((a or b) and (a or not b)) ?
```

 (1p)

10. Hur skriver man för att ange roten ur 3 med 3 decimalers noggrannhet? (1p)

11. Vad kallas de element som man kan referera till i datatypen *dictionary*? (2p)

12. Vilka värden kan en Boolesk variabel anta? (1p)

13. Betrakta den rekursiva koden:

```
1. def f(x):
2.     if len(x)>3:
3.         return f([x[0]]+x[2:])
4.     else
5.         return x
6.
7. y = int(input('Ange ett tal: '))
8. print(f([y,y+1,y+3,y+2]))
```

Vid exekvering av koden ger man indatan 3 och får svaret [3,6,5]. Ange i vilken ordning raderna exekverats med hjälp av radnumren. (2p)

14. Vad kallas den typ av programmering som handlar om att skapa funktioner som är separerade från de data de hanterar? (1p)

15. Vilket felmeddelande får man om man exekverar koden

```
a = 1/0
```

 i Python? (1p)

16. Hur kan man i Python skriva för att beräkna snittet mellan mängderna  $A = \{1,2,3\}$  och  $B = \{3,4\}$ ? (2p)

17. Betrakta Pythonkoden

```
a = 'abcde'
b = ''
for x in a:
    *
```

Vad måste det stå istället för \* för att värdet på b ska bli 'bcdea'? (2p)

18. Vid utskrift av resultat *data* till filen *fil.txt* så kan man använda kommandot `f = open(fil.txt,???)` och sedan `f.write(data)`. Vad ska man här skriva istället för ??? för att inte innehållet i filen *fil.txt* ska raderas innan värdet *data* skrivs till den, utan värdet *data* ska läggas till det innehåll som redan finns i filen? (1p)

19. Hur kan man på fyra rader skriva ett program som efterfrågar positiva heltal, ett efter ett, och då man anger 0 returnerar en lista med de angivna heltalen? (2p)

20. Hur kan man skriva ett program som givet listan

```
a = ['Anna', 'Bo', 'Sven', 'Gun']
```

returnerar en lista med samma element men i slumpmässig ordning? (2p)

21. Hur kan man på en rad skriva ett kommando (dvs en rad) som rensar bort allt utom bokstäverna från en given sträng och byter alla eventuella versaler (stora bokstäver) mot gemener (små bokstäver)? Så till exempel ska det, givet strängen 'Gunnar och Alice ville besöka SCB minst 3 gånger.', returnera strängen 'gunnarochalicevillebesokascbminstgångar'. (2p)

22. Klassen Die simulerar kast med tärning:

```
import random as r
class Die:
    def __init__(self):
        self.__shows = 1
    def toss(self):
        self.__shows = r.randint(1,6)
    def get_outcome(self):
        return self.__shows
```

Detta sparas i filen `die.py`. Ange nu hur du skriver programmet `tossing_a_die.py` som kastar tärning 5 gånger genom att använda klassen `Die` och anger resultatet för varje gång. (2p)

## Del 2

### 1. Rövarspråket

Röverspråket är definierat som det man får genom att, för varje konsonant i det som ska uttryckas, putta in bokstaven "o" och sedan upprepa konsonanten. Vokaler och andra tecken (ordmellanrum och skiljetecken) lämnas oförändrade. Så till exempel "Hej!" skulle på rövarspråket bli "Hohejoj!". Skriv ett *rekursiv* funktion som tar ett ord eller en hel mening som indata och returnerar översättningen till rövarspråket. (5p)

### 2. Diskret matematik

Kvoten vid heltalsdivision av talet  $a$  med talet  $n$  kan i Python beräknas med

```
a // n
```

och resten vid heltalsdivision av  $a$  med  $n$  beräknas med

```
a % n
```

För att beräkna resten av  $a \cdot b$  vid heltalsdivision med  $n$  kan man skriva

```
(a*b) % n
```

men också

```
a = a % n
b = b % n
(a*b) % n
```

för att kunna hantera större tal  $a$  och  $b$ .

- (a) Skriv nu programmet `diskr.py` som definierar funktionen `de`. Denna funktion ska ta 3 tal:  $a$ ,  $b$  och  $n$  och returnera resten vid heltalsdivision av  $a^b$  med  $n$ . Programmet ska vara skrivet så att det klarar att hantera maximalt stora tal  $a$  och  $b$ . (2p)
- (b) Skriv funktionen `d1` som också kan vara del av programmet `diskr.py`. Denna funktion ska ta 3 tal:  $a$ ,  $c$  och  $n$  och returnera det minsta positiva heltalet  $b$  sådant att  $a^b$  ger resten  $c$  vid heltalsdivision med  $n$ . Programmet ska vara skrivet så att det klarar att hantera maximalt stora tal  $a$  och  $c$ . (4p)

### 3. Skärmsläckare

Skriv ett program som med hjälp av modulen `tkinter` ritar punkter i ett grafiskt fönster som har samma storlek som en skärm med upplösningen  $1080 \times 1920$ . Varje ny punkt (en fylld svart cirkel med diameter 8 enheter) ska vara på avståndet 10 enheter ifrån föregående punkt. Detta kan åstadkommas genom att om föregående punkt har  $x$ -koordinat  $x$  och  $y$ -koordinat  $y$  så ska nästa punkt ha  $x$ -koordinat  $= x + 10 \cos v$  och  $y$ -koordinat  $= y + 10 \sin v$  där  $v$  är vinkeln för nästa punkt i förhållande till föregående. Denna vinkel ska genereras rekursivt enligt  $v = v + \text{random.uniform}(-1,1)$ . Punkterna får inte försvinna ”utanför fönstret” – om de är på väg mot kanten av ritytan måste de ”studsa tillbaka”. Efter att varje ny punkt ritats kan man ange `.update()` till fönstret där punkterna ritas så att det uppdateras. (5p)

### 4. Receptgenerator

Här ska genereras olika varianter på temat *Pasta med tomatsås!*

- (a) Först ska du bilda en klass `Pasta` och en annan klass `Tomat`. Ett objekt av klassen `Pasta` ska, via funktionen `.choose()` för denna klass, slumpmässigt anta värdet `Spaghetti`, `Fusilli` eller `Tagliatelle`. Ett objekt av klassen `Tomat` ska, via funktionen `.choose()` för denna klass, anta värdet av en lista. Listan ska bestå av 3–5 ingredienser slumpmässigt valda bland `Lök`, `Vitlök`, `Syltlök`, `Tabasco`, `Basilika`, `Chilsås` och `Jalapeno`. Observera att inte bara ingredienserna ska vara slumpvis valda utan också *antalet* ingredienser ska vara slumpmässigt. (3p)
- (b) Programmet `receptgenerator.py` ska sedan, genom att använda klasserna `Pasta` och `Tomat`, ge ett förslag på ett recept bestående av värdet av ett objekt av klassen `Pasta` och värdet av ett objekt av klassen `Tomat`. Dessutom ska ingrediensen `Krossade tomater` alltid finnas med i receptet. (2p)

## Apendix till tentamina i Programmering DI2006

### Python: sammanfattning

Kod	Förklaring
<pre>name = "Kalle" age = 10 pi = 3.14 savings= BankAccount()</pre>	<p>En variabel av <b>typen string</b> En variabel av <b>typen int</b> En variabel av <b>typen float</b> <b>En variabel av typen BankAccount</b></p>
<pre>print(y) print("Hej")</pre>	<p><b>Skriver ut</b> värdet av variabeln y. Skriver ut "Hej"</p>
<pre>x = input("Namn: ")</pre>	<p><b>Skriver ut:</b> "Namn: " och <b>pausar tills användaren skriver in</b> något och trycker på Enter. <b>Lagrar sedan</b> texten användaren skrev in, i variabeln x.</p>
<pre>x = 15 if x &lt; 10:     s = "x &lt; 10"     print(s) elif x &lt; 20:     s = "x &lt; 20"     print(s) elif x == 20:     print("x = 20") else:     print("else:")     print("x &gt; 20")</pre>	<p>Villkorssatser.</p> <p><b>if</b> testar om ett villkor är uppfyllt.</p> <p><b>elif</b> hänger ihop med if och testas bara om testet i if misslyckas.</p> <p><b>else</b> hänger också ihop med if och testas bara om det i if och eventuella elif misslyckas.</p>
<pre>age == 100</pre>	<p>Exempel på villkor man kan testa. <b>Kom ihåg: == INTE =</b> när man vill testa ett villkor!</p>
<pre>x &lt; 100 and y &lt; 100</pre>	<p>Exempel på villkor man kan testa. Testar om både variabeln x <b>och</b> variabeln y är mindre än 100.</p>

<pre>for tal in range(100):     print(tal)</pre>	<p><b>for loop</b></p> <p>För varje tal från 0 till och med 99: Skriv ut talet.</p>
<pre>for num in [1, 6, 2, 5]:     print(num)</pre>	<p>För varje nummer i listan [1, 6, 2, 5]: Skriv ut numret.</p>
<pre>var = 10 while var &lt; 100:     print(var)     var = var + 1</pre>	<p><b>while loop</b></p> <p>Medans villkoret var &lt; 100 är uppfyllt, gör följande om och om igen:</p> <ul style="list-style-type: none"> <li>- Skriv ut värdet av var</li> <li>- Öka var med ett .</li> </ul>
<pre>def funktionsnamn(a, b):     r = a*b + 10     r = r/10     r = r - 10     return r</pre>	<p><b>Funktionsdefinition.</b> Denna kod körs först då funktionen anropas. a och b kallas för <b>parametrar</b> och är variabler utan värde som kan användas inuti funktionen.</p>
<pre>funktionsnamn(1, 2)  x = funktionsnamn(2, 5)</pre>	<p><b>Anrop</b> av funktionen <i>funktionsnamn</i> kör koden i funktionsdefinitionen.</p> <p><b>Argumenten</b> 1 och 2 är värden som lagras i parametrarna i funktionsdefinitionen</p>
<pre>fil=open("filnamn", "r")  for ord in fil:     //gör något  text=fil.read() line=fil.readline()</pre>	<p>Öppna en fil och läs den ord för ord. w- skriv , a-append</p> <p>läser en fil till variabel text</p> <p>läser en rad från fil till variabel line</p>
<pre>lista=[]  for var in lista</pre>	<p>Skapa tom lista</p> <p>Iterera genom lista</p>

## Skapa klasser och objekt

### class BankAccount:

```
def __init__(self, iname, ibalance):
    self.__name=iname
    self.__balance=ibalance

def deposit(self, amount):
    balance=balance-amount
```

```
savigs= BankAccount("kalle", 100) // skapa objekt
```

### Användbara funktioner:

```
int(a) // konverterar a från String till int
str(a) // konverterar a från int till String
ord(a) // konverterar a från char till int
chr(a) // konverterar a från int till char
```

```
random.randint(a,b) // slumpar tal mellan a och b
```

text.split(delimitator) Dela en sträng i en lista av ord. Argumenten delimitator används som separator. Metoden returnerar en lista.

### Andra String metoder:

**Table 8-1** Some string testing methods

Method	Description
isalnum()	Returns true if the string contains only alphabetic letters or digits and is at least one character in length. Returns false otherwise.
isalpha()	Returns true if the string contains only alphabetic letters and is at least one character in length. Returns false otherwise.
isdigit()	Returns true if the string contains only numeric digits and is at least one character in length. Returns false otherwise.
islower()	Returns true if all of the alphabetic letters in the string are lowercase, and the string contains at least one alphabetic letter. Returns false otherwise.
isspace()	Returns true if the string contains only whitespace characters and is at least one character in length. Returns false otherwise. (Whitespace characters are spaces, newlines (\n), and tabs (\t).
isupper()	Returns true if all of the alphabetic letters in the string are uppercase, and the string contains at least one alphabetic letter. Returns false otherwise.

**Table 8-3** Search and replace methods

Method	Description
endswith(substring)	The <i>substring</i> argument is a string. The method returns true if the string ends with <i>substring</i> .
find(substring)	The <i>substring</i> argument is a string. The method returns the lowest index in the string where <i>substring</i> is found. If <i>substring</i> is not found, the method returns -1.
replace(old, new)	The <i>old</i> and <i>new</i> arguments are both strings. The method returns a copy of the string with all instances of <i>old</i> replaced by <i>new</i> .
startswith(substring)	The <i>substring</i> argument is a string. The method returns true if the string starts with <i>substring</i> .



## List metoder

**Table 7-1** A few of the list methods

Method	Description
<code>append(item)</code>	Adds <i>item</i> to the end of the list.
<code>index(item)</code>	Returns the index of the first element whose value is equal to <i>item</i> . A <code>ValueError</code> exception is raised if <i>item</i> is not found in the list.
<code>insert(index, item)</code>	Inserts <i>item</i> into the list at the specified <i>index</i> . When an item is inserted into a list, the list is expanded in size to accommodate the new item. The item that was previously at the specified index, and all the items after it, are shifted by one position toward the end of the list. No exceptions will occur if you specify an invalid index. If you specify an index beyond the end of the list, the item will be added to the end of the list. If you use a negative index that specifies an invalid position, the item will be inserted at the beginning of the list.
<code>sort()</code>	Sorts the items in the list so they appear in ascending order (from the lowest value to the highest value).
<code>remove(item)</code>	Removes the first occurrence of <i>item</i> from the list. A <code>ValueError</code> exception is raised if <i>item</i> is not found in the list.
<code>reverse()</code>	Reverses the order of the items in the list.

## Dictionary metoder:

**Table 9-1** Some of the dictionary methods

Method	Description
<code>clear</code>	Clears the contents of a dictionary.
<code>get</code>	Gets the value associated with a specified key. If the key is not found, the method does not raise an exception. Instead, it returns a default value.
<code>items</code>	Returns all the keys in a dictionary and their associated values as a sequence of tuples.
<code>keys</code>	Returns all the keys in a dictionary as a sequence of tuples.
<code>pop</code>	Returns the value associated with a specified key and removes that key-value pair from the dictionary. If the key is not found, the method returns a default value.
<code>popitem</code>	Returns a randomly selected key-value pair as a tuple from the dictionary and removes that key-value pair from the dictionary.
<code>values</code>	Returns all the values in the dictionary as a sequence of tuples.