



TENTAMEN I PROGRAMMERING DI2006 OCH DT2018

Datum: 2021-06-02

Tid: 15.00–19.00

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

Anvisningar

- Tentan fotograferas och skickas in senast klockan 19.00 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 20 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. Vilken datatyp har en variabel med värdet "123"? (1p)

2. Vilka värden kan en boolesk variabel anta? (1p)

3. Vad kallas den typen av diagram som används för att designa en klass vid objektorienterad programmering? (Den 3-bokstaviga förkortningen räcker som svar.) (1p)

4. Hur kan koden

```
if score>100:
    next_level = True
else:
    next_level = False
if next_level:
    print("You are go for next level!")
```

skrivs på bara två rader? (1p)

5. Nämn en typ av sats kan användas för att skriva en dual beslutsstruktur? (1p)

6. För att beräkna produkten

$$\frac{1}{2} \cdot \frac{3}{2} \cdot \frac{3}{4} \cdot \frac{5}{4} \cdot \frac{5}{6} \cdot \frac{7}{6} \cdots \frac{101}{102} \cdot \frac{103}{102}$$

skrivs koden

```
p = 1
for i in range(101):
    p = p*((i%2==0)*i/(i+1)+(i%2==1)*(i+1)/i)
```

(a) Denna kod ger dock felmeddelandet `ZeroDivisionError: division by zero.` Varför då? (1p)

(b) Hur kan man ändra koden så att den beräknar produkten helt korrekt? (2p)

7. Om koden

```
a = list('scoobydoo')
b = 'abcdef'
answ = ''
for i in b[:(-j)]:
```

Vad måste det stå istället för `*` för att värdet på `answ` ska bli `'odybo'` om `j=1` och värdet på `answ` ska bli `'ody'` om `j=3` utan att använda `if`-sats? (2p)

8. Vad brukar man med en trebokstavig förkortning kalla mikroprocessorn i en dator? (1p)

9. Vad kallas kod som specificerar de data-attribut och metoder som gäller för en viss typ av objekt? (1p)

10. Man vill bilda en mängd, **A**, med 11 slumpvisa tal mellan 0 och 101 och en annan, **B**, med 22 slumpvisa tal mellan 2 och 202. Hur kan man
- (a) i Python skriva för att bilda mängderna **A** och **B**? (2p)
 - (b) på en enda rad i Python bilda en mängd, **C**, med de element som både ingår i **A** och i **B**? (1p)
11. Ange en Pythonkod som efterfrågar 2 bilmärken och sedan svarar med att skriva ut dessa bilmärken åtskilda av en TAB. (1p)
12. Vad kallas det talsystem som bara använder siffrorna 0 och 1? (1p)
13. Betrakta den rekursiva koden
- ```
def f(a):
 if len(a)<2:
 return a
 elif len(a)<4:
 a.reverse()
 return(a)
 else:
 return(a[:(-2)]+f(a[2:]))
```
- Vad returneras av `f(list('abcdef'))`? (2p)
14. Ange Pythonkod som efterfrågar ett positivt heltal **n** och svarar genom att skriva **n** stycken tecken slumpmässigt valda bland **a, b, c, d, e, f, g** till filen `some_chars.txt`. (2p)
15. Vad är en *inkapsling* (eng. *encapsulation*)? (1p)
16. Skriv ett litet program som frågar om man vill resa till Ödeshög med tåg. Om man svarar **N** så ska programmet returnera meddelandet **Nähä**. men om man svarat något annat än **N** så ska programmet fråga om man vill resa i första eller andra klass där svaret ska anges med motsvarande siffra. Om man svarat **1** så ska programmet svara **Då kostar biljetten 789:-, tack!** men om man svarat något annat än **1** så ska programmet svara **Då kostar biljetten 123:-, tack!**. (2p)
17. Skriv ett litet program som simulerar ett tärningskast. Då programmet körs ska det svara **Tärningen visar:** och därefter resultatet av tärningskastet som ska vara ett slumpmässigt heltal mellan 1 och 6. (1p)
18. Låt `lista1` vara en lista med 7 heltal. Hur kan man på en rad generera listan `lista2` som innehåller kvadraterna på talen i `lista1`? (2p)
19. Låt `s` vara strängen "Det var en gång en liten liten gumma". Hur kan man nu mha Regular Expressions, på två rader, bilda en lista av alla **e** och **n** som finns i strängen `s`? (2p)

20. I Python kan man ha både lokala och globala variabler. Om man skrivit

```
v = 'gurka'
def f():
 v = 'banan'
```

och därefter kör programmet `f`

```
f()
```

vad kommer då variabeln `v` ha för värde?

(1p)

## Del 2

### 1. Reseparlör

Inför sin resa till Spanien gör Ulrik en liten parlör dvs en liten svensk-spansk ordbok. Ulrik har efter ett tags användande av sitt program, `translate.py`, genererat en fil, `parlor.txt`, där svenska ord står i en första kolumn och respektive spanska ord står intill sin svenska motsvarighet i en andra kolumn. Varje gång han kommer på ett svenskt ord skriver han det till programmet `translate.py`. Om ordet finns med bland de svenska ord som tidigare sparats (i första kolumnen i filen `parlor.txt`) händer inget men om det inte finns med så efterfrågas även den spanska motsvarigheten och båda sparas till första respektive andra kolumnen i filen `parlor.txt`. Sedan skriver Ulrik även programmet `parlor.py` som då man kör det först läser in filen `parlor.txt` som en dictionary. Sedan efterfrågas ett svenskt ord och programmet svarar med den spanska motsvarigheten. Därefter frågar programmet användaren om man vill slå upp ett nytt ord. Om man svarar J så börjar proceduren om på nytt (ett nytt svenskt ord efterfrågas och motsvarande spanskt ord returneras osv) men om svaret blir något annat än J terminerar programmet. Skriv nu programmet

- (a) `translate.py` som ska läsa från och skriva till filen `parlor.txt` enligt beskrivningen ovan. (4p)
- (b) `parlor.py` som ska läsa in `parlor.txt` som datatypen `dictionary` och efterfråga ett svenskt ord och sedan leverera motsvarande spansk översättning om ordet finns i parlören. (3p)

### 2. Hur säkert blir transpositionskryptot?

Det sägs att en text blir hyfsat läsbar trots att man slumpmässigt flyttar om bokstäverna inom varje ord, bara man har kvar första och sista bokstaven i orden. Exempelvis kan man kanske se att

```
Sgunjom seetntduns lglckya dag,
ltåom oss faöjdrs i uennmogds vår!
```

egentligen är texten

```
Sjungom studentens lyckliga dag,
låtom oss fröjdas i ungdomens vår!
```

fast de inre bokstäverna i varje ord permuterats. Skriv ett program, `transposition.py`, som läser in en text från filen `text.txt` och returnerar en motsvarande text där de inre bokstäverna i orden permuterats enligt exemplet ovan. Ord mellanrum, radbrytningar och skiljetecken ska dock vara kvar precis som i originaltexten. (7p)

### 3. *Kanonkul*

Om en projektil skjuts iväg i ett gravitationsfält beskrivs projektilens bana av en  $x^2$ -kurva. Du ska i denna uppgift göra en GUI som beskriver om en kanonkula träffar sitt mål.

Det grafiska fönstret ska vara så stort som möjligt (t.ex.  $1000 \times 1920$  pixlar om upplösningen är  $1080 \times 1920$ ). Längst ned i bild ska kanonen och målet finnas. Kanonen ska vara en ofylld cirkel med diameter 20 pixlar längst till vänster och målet ska vara en fylld cirkel med diameter 20 pixlar på en placering horisontellt slumpmässig mellan 500 – 1800 pixlar från vänsterkanten.

Sedan ska projektilen avfyras. För att göra detta efterfrågas användaren om vilken vinkel som kanonen ska ställas in med. Denna vinkel ska vara mellan 5–85 grader. Sedan ska vinkeln transformeras till radianer: om variaben  $v$  är vinkeln uttryckt i grader så ska detta transformeras till  $\frac{\pi}{180}v$ . Därefter ska projektilens bana ritas i det grafiska fönstret. Detta ska göras i steg om 10 pixlar i horisontellt led (dvs längs  $x$ -axeln). Så banan ska ritas med ett streck från punkten med  $x = 10$  till punkten med  $x = 20$ , nästa streck från punkten med  $x = 30$  till punkten med  $x = 40$ , osv. Motsvarande  $y$ -koordinater ska beräknas mha funktionen `arc` som definieras i början av programmet. Lämpligen börjar programkoden enligt:

```
import tkinter
import math
import random
def arc(x,v):
 return(1000+0.001*x*(x-4000*math.tan(v)/2)/(1+(v-math.pi/4)**2))
```

där funktionen `arc` tar  $x$ -koordinaten `x` och vinkeln uttryckt i radianer `v` som indata och returnerar värdet på  $y$ -koordinaten `y` för bågen som beskriver punkten  $(x, y)$  i projektilens bana. Det är givetvis bara den del av projektilbanan som hamnar i det grafiska fönstret som ska ritas.

Sedan ska det avgöras om projektilen träffat sitt mål – dvs om banans slutpunkt är inom avståndet 20 pixlar från målpunkten. Om projektilen inte träffat målet ska meddelandet **Bom!** skrivas ut och användare ska ånyo ange en vinkel för att avfyra en ny projektil osv. Om projektilen däremot träffat målet ska meddelandet **TRÄFF!!!** skrivas ut och programmet ska terminera. (6p)

## 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></p> <p>En variabel av <b>typen int</b></p> <p>En variabel av <b>typen float</b></p> <p><b>En variabel av typen BankAccount</b></p>                                                                                            |
| <pre>print(y) print("Hej")</pre>                                                                                                                                                                    | <p><b>Skriver ut</b> värdet av variabeln y.</p> <p>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.</p> <p><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.</p> <p><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.</p> <p>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.                                                                                                   |

## GUI metoder

**Table 13-2** Some of the optional arguments to the `create_line` method

| Argument                  | Description                                                                                                                                                                                                                                                                                                                                                                                          |
|---------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>arrow=value</code>  | By default, lines do not have arrowheads, but this argument causes the line to have an arrowhead at one or both ends. Specify <code>arrow=tk.FIRST</code> to draw an arrowhead at the beginning of the line, <code>arrow=tk.LAST</code> to draw an arrowhead at the end of the line, or <code>arrow=tk.BOTH</code> to draw arrowheads at both ends of the line.                                      |
| <code>dash=value</code>   | This argument causes the line to be a dashed line. The value is a tuple, consisting of integers, that specifies a pattern. The first integer specifies the number of pixels to draw, the second integer specifies the number of pixels to skip, and so forth. For example, the argument <code>dash=(5, 2)</code> will draw 5 pixels, skip 2 pixels, and repeat until the end of the line is reached. |
| <code>fill=value</code>   | Specifies the color of the line. The argument's value is the name of a color, as a string. There are numerous predefined color names that you can use, and Appendix D shows the complete list. Some of the more common colors are 'red', 'green', 'blue', 'yellow', and 'cyan'. (If you omit the <code>fill</code> argument, the default color is black.)                                            |
| <code>smooth=value</code> | By default, the <code>smooth</code> argument is set to <code>False</code> , which makes the method draw straight lines connecting the specified points. If you specify <code>smooth=True</code> , the lines are drawn as curved splines.                                                                                                                                                             |
| <code>width=value</code>  | Specifies the width of the line, in pixels. For example, the argument <code>width=5</code> causes the line to be 5 pixels wide. By default, lines are 1 pixel wide.                                                                                                                                                                                                                                  |

**Table 13-4** Some of the optional arguments to the `create_oval` method

| Argument                   | Description                                                                                                                                                                                                                                                                                                                                                                                                         |
|----------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>dash=value</code>    | This argument causes the outline of the oval to be a dashed line. The value is a tuple, consisting of integers, that specifies a pattern. The first integer specifies the number of pixels to draw, the second integer specifies the number of pixels to skip, and so forth. For example, the argument <code>dash=(5, 2)</code> will draw 5 pixels, skip 2 pixels, and repeat until the end of the line is reached. |
| <code>fill=value</code>    | Specifies a color to fill the oval with which. The argument's value is the name of a color, as a string. There are numerous predefined color names that you can use, and Appendix D shows the complete list. Some of the more common colors are 'red', 'green', 'blue', 'yellow', and 'cyan'. (If you omit the <code>fill</code> argument, the oval will not be filled.)                                            |
| <code>outline=value</code> | Specifies the color of the oval's outline. The argument's value is the name of a color, as a string. There are numerous predefined color names that you can use, and Appendix D shows the complete list. Some of the more common colors are 'red', 'green', 'blue', 'yellow', and 'cyan'. (If you omit the <code>outline</code> argument, the default color is black.)                                              |
| <code>width=value</code>   | Specifies the width of the oval's outline, in pixels. For example, the argument <code>width=5</code> causes the line to be 5 pixels wide. By default, lines are 1 pixel wide.                                                                                                                                                                                                                                       |