



TENTAMEN I PROGRAMMERING DI2006

Datum: 2022-01-11

Tid: 15.00–19.00

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

Anvisningar

- Tillåtna hjälpmedel är
 - formelsamling (som är häftad till tentamenstexten)
 - miniräknare TI-30Xa (Texas Instruments)
 - skrivpapper
 - penna
 - suddigummi
 - linjal
 - frukt, fika
- Till varje uppgift finns angivet hur många poäng som maximalt utdelas för uppgiften.
- Tentamen består av två delar: **Del 1** och **Del 2**.
- Samtliga frågor i Del 1 ska besvaras i den svarstalong som är bifogad med tentamenstexten.
- Frågorna i Del 2 ska besvaras på vanligt separat rutat papper.
- Då programkod anges som svar ska den vara i så körbart skick som möjligt.
- Del 1 består av 22 frågor och här kan man maximalt få 30 poäng.
- Del 2 består av 3 frågor och här kan man maximalt få 20 poäng.
- För betyg 3 krävs minst 15 poäng på Del 1. (Del 2 behöver inte alls göras för att få detta betyg.)
- För betyg 4 krävs minst 15 poäng på Del 1 och 5 poäng på Del 2.
- För betyg 5 krävs minst 15 poäng på Del 1 och 10 poäng på Del 2.

LYCKA TILL!

Svartsalong till Del 1

Svara med ett kryss i *ett* av alternativen (a)–(g) för var och en av frågorna 1–14.
Besvara frågorna 15–22 med text på de angivna raderna.

Fråga	Svarsalternativ						
	(a)	(b)	(c)	(d)	(e)	(f)	(g)
1.							
2.							
3.							
4.							
5.							
6.							
7.							
8.							
9.							
10.							
11.							
12.							
13.							
14.							

15. _____ 16. _____

17. _____

18. _____

19. _____

20. _____ 21. _____

22. _____

Namn/anonymitetskod: _____

Del 1

1. Programmeringsspråket Python är ett exempel på (1p)
- (a) maskinkod
 - (b) pseudokod
 - (c) födesschema
 - (d) naturligt språk
 - (e) högnivåspråk
 - (f) relationellt språk
 - (g) Inget av de ovanstående alternativen
-

2. Vad kallas ett program som utan mellansteg exekverar instruktioner skrivna i ett programmeringsspråk? (1p)
- (a) En kompilator
 - (b) Ett orakel
 - (c) En translator
 - (d) En examinerator
 - (e) En dator
 - (f) En interpretator
 - (g) Inget av de ovanstående alternativen
-

3. Vilken version av Python handlar kursen Programmering DI2006 främst om? (1p)
- (a) Python 1
 - (b) Python 2
 - (c) Python 3
 - (d) Python 4
 - (e) Python 5
 - (f) Python 6
 - (g) Inget av de ovanstående alternativen
-

4. Om man vill att ett program ska ange ett numeriskt värde, `a`, med 7 decimalers noggrannhet skriver man (1p)
- (a) `print(format(a, '.7f'))`
 - (b) `output(a, format='7.')`
 - (c) `write(a:decimals=7)`
 - (d) `write(digits(a,7)`
 - (e) `print(a+{'digits:7'})`
 - (f) `echo(dec:7,num:a)`
 - (g) Inget av de ovanstående alternativen
-

5. Om man i Python skriver koden

```
a=1
b=0
if a>0:
    b=a+b
    a=b-a
elif a>=1:
    b=a-b
    a=b+a
else:
    a=a-b
    b=b+a
```

så får

(1p)

- (a) a värdet 1 och b värdet 0
 - (b) a värdet 0 och b värdet 1
 - (c) a värdet 2 och b värdet 1
 - (d) a värdet 1 och b värdet 2
 - (e) a värdet 0 och b värdet 2
 - (f) a värdet 2 och b värdet 0
 - (g) Inget av de ovanstående alternativen
-

6. Vad kallas det när man definierar en funktion genom att någon eller några av dess värden ska beräknas med hjälp av funktionen själv? (1p)

- (a) Självreferens
 - (b) Induktion
 - (c) Rekurrens
 - (d) Rekursion
 - (e) Loop
 - (f) Feedback
 - (g) Inget av de ovanstående alternativen
-

7. Vad kallas den datatyp som är sekventiell och vars element är par av objekt kallade *key* och *value*? (1p)

- (a) Function
 - (b) Dictionary
 - (c) Library
 - (d) Int
 - (e) String
 - (f) List
 - (g) Inget av de ovanstående alternativen
-

8. Förkortningen GUI står för (1p)
- (a) Graphical User Interface
 - (b) Gravity Universe Infinite
 - (c) Golems Usually Included
 - (d) Generally Under Inspection
 - (e) Gradual Understanding of Implementation
 - (f) Gamma Unmasking Ideology
 - (g) Inget av de ovanstående alternativen
-
9. Vad den datatyp som kan ha funktionella egenskaper i sin definition? (1p)
- (a) Booleska värden
 - (b) Tupel
 - (c) Hidden
 - (d) Lista
 - (e) Funktion
 - (f) Klass
 - (g) Inget av de ovanstående alternativen
-
10. För att generera ett slumpmässigt tal från och med 10 till och med 100 kan man importera biblioteket random och därifrån ange (1p)
- (a) `random(10,100)`
 - (b) `random(10,101)`
 - (c) `randrange(10,100)`
 - (d) `randrange(10,101)`
 - (e) `randint(10,100)`
 - (f) `randint(10,101)`
 - (g) Inget av de ovanstående alternativen
-
11. Objektorienterad programmering är fokuserad kring att (1p)
- (a) kryptera funktioner
 - (b) bilda procedurer
 - (c) skapa objekt
 - (d) skapa instanser
 - (e) manipulera listor och strängar
 - (f) eliminera klasser
 - (g) Inget av de ovanstående alternativen
-

12. Vad kallas den obligatoriska första metoden `__init__` vid definition av en klass?
(1p)

- (a) Initiering
 - (b) Pionjär
 - (c) Konstruktör
 - (d) Destruktör
 - (e) Genesis-segment
 - (f) Head block
 - (g) Inget av de ovanstående alternativen
-

13. Vad kallas den del av en dator som där alla kommandon på lägsta nivå fördelas?
(1p)

- (a) Nätaggregat
 - (b) RAM-minne
 - (c) Grafkort
 - (d) Moderkort
 - (e) Mikroprocessor
 - (f) Nätverkskort
 - (g) Inget av de ovanstående alternativen
-

14. Funktionen

```
def f(t,s):
    if t==():
        return s
    else:
        s=s+t[-1]
        del t[-1]
        return f(t,s)
```

`f((5,7,1,3),0)`

ger utskriften

(1p)

- (a) 16
 - (b) 5
 - (c) 5713
 - (d) -1
 - (e) `TypeError`
 - (f) `AttributeError`
 - (g) Inget av de ovanstående alternativen
-

15. Om man exekverar koden (2p)

```
a='abrakadabra'
b='abcd'
answ=''
for i in range(len(b)):
    if b[i] in a:
        answ=answ+a[i+1]

print(answ)
vad skrivs då ut?
```

16. Betrakta koden (2p)

```
a=int(input('Ange ett heltal: '))

f=open('siffror.txt','w')
while a>0:
    if a%3==1:
        f.write(str(a**2))
    a=a-1

f.close()
s=0
f=open('siffror.txt')
temp=list(f.read())
while len(temp)>0:
    ???
    temp=temp[:-1]

print(s)
```

Vad ska stå på raden `???` om man får utskriften 22 då man angivit 10 som värde på `a` men får utskriften 38 då man angivit 13 som värde på `a`?

17. Hur kan man skriva koden (2p)

```
a=[]
i=0
while i<5:
    a.append(int(i*3.7))
    i+=1

på en rad?
```

18. Betrakta koden (2p)

```
def rev(s):
    if len(s)==0:
        return ''
    else:
        ???
```

Ange vad som ska stå på raden `???` så att funktionen `rev` blir en *rekursiv* funktion som tar strängen `s` som argument och returnerar värdet av samma sträng fast baklänges. Till exempel ska `rev('hej')` returnera värdet `'jeh'`.

19. Komplettera koden

(2p)

```
try:
    p=int(input('Ange ett heltal: '))
    q=int(input('Ange ett till dock ej lika med 0: '))
    print('Vid heltalsdivision av '+ 111 +' med '+ 222 +' blir kvoten '\
        + 333 +' och resten '+ 444 +'.')
555
    print('Nämnamaren '+str(q)+' ger division med 0 vilket inte är tillåtet.')
666
    print('Detta är ett tal men inget heltal.')
777
    print('Något blev fel...')
```

genom att ange vad som ska stå på platserna **111**, **222**, **333**, **444**, **555**, **666** och **777** så att den skriver ut kvoten och resten vid heltalsdivision och de angivna felmeddelandena för respektive fel.

20. Betrakta koden

(2p)

```
import random as r
a=[]
b=a
a.append(["Diabas", "Granit"] [r.randint(0,1)])
b.append(["Tulpan", "Pion"] [r.randint(0,1)])
print(???)
```

Vad ska stå på raden **???** om man vill ha reda på innehållet i listan b utan att använda bokstaven b?

21. Bertil deltar i en biljardturnering tillsammans med 2 andra: Gunnar och Göran. Deras poäng sparas i en dator och vid dagens slut ska deras namn och poängresultat som är sparade i en dictionary: dict={"Bertil":27, "Gunnar":39, "Göran":32} skrivas ut. Hur kan man på två rader åstadkomma detta? (2p)

22. Hur kan man på en rad skapa en lista som består av heltalen från och med 1 till och med 100? (2p)

Del 2

1. *Sten-sax-påse*

Detta är ett spel som kan spelas av två personer, A och B. Då man spelar spelet *Sten-sax-påse* gäller att vid ett antal omgångar välja ett av alternativen sten, sax eller påse. Både A och B måste göra sitt val exakt samtidigt så att inte den ena vet i förväg vad den andre ska välja. Den enkla regeln är: sax vinner över påse, påse vinner över sten och sten vinner över sax.

- (a) Implementera en klass `StenSaxPase` som spelar detta spel. Användaren (A) ombedes göra sitt val (`sten`, `sax` eller `påse`) genom att skriva in det. Då A tryckt `<ENTER>` meddelar datorn (B) sitt val och om det innebär att A eller B vunnit. Om A skrivit något annat än ett av de tre alternativen ska A meddelas att valet var ogiltigt och ombedas att ånyo göra sitt val. (3p)
- (b) Implementera en ny klass `StenSaxPase2` som efter varje omgång frågar om A vill spela en omgång till. Då A förr eller senare väljer att sluta ska A meddelas hur slutställningen blev: hur många gånger A vann och hur många gånger B vann och därmed vem som vunnit totalt. (2p)

2. *Simulering*

För att beräkna förväntade kvantiteter i komplicerade teoretiska modeller kan det bli väldigt svårt (och till och med omöjligt) att enbart använda matematiska resonemang. Då kan en alternativ väg till lösning vara att simulera resultatet med hjälp av en slumpvalsgenerator (PRNG).

Antag att man vill få reda på masscentrum i sannolikhetsfördelningen för det tal man får om man beräknar

$$\frac{1}{(1 + X_1)^{X_1}} + \frac{1}{(2 + X_2)^{X_2}} + \frac{1}{(3 + X_3)^{X_3}} + \dots$$

där X_1, X_2, X_3, \dots är slumpantal som oberoende av varandra är 2, 3 eller 4 med samma sannolikhet, dvs det är 33% chans att $X_1 = 2$, 33% chans att $X_1 = 3$ och 33% chans att $X_1 = 4$. Sedan är det 33% chans att $X_2 = 2$, och så vidare. Detta masscentrum är faktiskt omöjligt att beräkna med exakthet.

- (a) Skriv ett program som med hjälp av en *rekursiv* funktion, `braksumma`, genererar slumpantal `x1`, `x2`, `x3`, ... som oberoende av varandra är värdena 2, 3, 4, 5 med samma sannolikhet. Detta görs genom att importera ett lämpligt bibliotek och därifrån använda en lämplig funktion. Den rekursiva funktionen ska beräkna summan $\mathbf{s} = \frac{1}{(1+x_1)^{x_1}} + \frac{1}{(2+x_2)^{x_2}} + \frac{1}{(3+x_3)^{x_3}} + \dots$ där antalet termer i summan är så stort att avståndet mellan de två sista termerna i summan är mindre än 1‰ (en tusendel). Funktionen ska returnera värdet av summan: `s`. (3p)
- (b) Skriv en annan funktion, `masscentrum`, som använder sig av funktionen i (a) och med hjälp av den beräknar värdet av 10 000 sådana summor: `s1`, `s2`, `s3`, ..., `s10000`. Därefter ska medelvärdet av summorna beräknas: $(\mathbf{s1} + \mathbf{s2} + \mathbf{s3} + \dots + \mathbf{s10000}) / 10000$. Returnera detta svar med 5 decimalers noggrannhet som funktionsvärdet. (2p)

3. Bokstavsblandning

Det sägs att ett meddelande förblir hyfsat läsbart trots att man slumpmässigt kastar om bokstäverna inom varje ord så länge man behåller första och sista bokstaven i varje ord. Till exempel skulle meddelandet

packa pappas kappsäck

kunna bli

pakca ppaaps kscpaäpk

- (a) Skriv ett program med en funktion som läser filen `text.txt` och returnerar dem i en lista av orden som element. Orden ska bara innehålla bokstäverna a-ö, inga specialtecken som ordmellanrum, `\n` (newline), `\t` (tab), `.` (punkt), `,` (komma) osv). Alla stora bokstäver ska omvandlas till motsvarande små. (2p)
- (b) Skriv i nämnda program en annan funktion som tar ett ord i form av en sträng som argument och behåller första och sista bokstav i ordet men slumpmässigt kastar om de inre bokstäverna. Funktionen ska returnera det nya ordet med omkastade inre bokstäver. (4p)
- (c) Skriv ytterligare en funktion i samma program som tar en text i form av en lista av strängar som argument och skriver den till filen `ny_text.txt`. (2p)
- (d) Skriv huvuddelen av programmet som använder funktionerna i (a), (b) och (c) för att läsa in en text från filen `text.txt`, kasta om de inre bokstäverna i varje ord och skriva resultatet i filen `ny_text.txt`. (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 typen string En variabel av typen int En variabel av typen float En variabel av typen BankAccount</p>
<pre>print(y) print("Hej")</pre>	<p>Skriver ut värdet av variabeln y. Skriver ut "Hej"</p>
<pre>x = input("Namn: ")</pre>	<p>Skriver ut: "Namn: " och pausar tills användaren skriver in något och trycker på Enter. Lagrar sedan texten användaren skrev in, i variabeln x.</p>
<pre>x = 15 if x < 10: s = "x < 10" print(s) elif x < 20: s = "x < 20" print(s) elif x == 20: print("x = 20") else: print("else:") print("x > 20")</pre>	<p>Villkorssatser.</p> <p>if testar om ett villkor är uppfyllt.</p> <p>elif hänger ihop med if och testas bara om testet i if misslyckas.</p> <p>else 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. Kom ihåg: == INTE = när man vill testa ett villkor!</p>
<pre>x < 100 and y < 100</pre>	<p>Exempel på villkor man kan testa. Testar om både variabeln x och variabeln y är mindre än 100.</p>

<pre>for tal in range(100): print(tal)</pre>	<p>for loop</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 < 100: print(var) var = var + 1</pre>	<p>while loop</p> <p>Medans villkoret var < 100 är uppfyllt, gör följande om och om igen:</p> <ul style="list-style-type: none"> - Skriv ut värdet av var - Öka var med ett .
<pre>def funktionsnamn(a, b): r = a*b + 10 r = r/10 r = r - 10 return r</pre>	<p>Funktionsdefinition. Denna kod körs först då funktionen anropas. a och b kallas för parametrar och är variabler utan värde som kan användas inuti funktionen.</p>
<pre>funktionsnamn(1, 2) x = funktionsnamn(2, 5)</pre>	<p>Anrop av funktionen <i>funktionsnamn</i> kör koden i funktionsdefinitionen.</p> <p>Argumenten 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.