

**PROGRAMMERING-PYTHON
TENTAMINA
2013-03-22**

Kontaktperson: Johan Thunberg, tel: 0729-773846

Tentamensinstruktioner

Poängsättning

Tentamina är indelad i två delar

Del 1. Innehåller uppgifter för betyg 3, totalt 20 poäng.
Du måste få minst 12 poäng för att få betyg 3.

Del 2. Innehåller uppgifter för betyg 4 och 5 totalt 10 poäng.
Du måste ha fått betyg 3 på Del 1 för att kunna få poäng på Del2. Del 2 kommer alltså inte att beaktas (eller betraktas) om betyg 3 inte har erhållits på Del 1.

För betyg 4 krävs det 6 poäng och för betyg 5, 8 poäng.

Redovisning

Tänk på att vara noggrann och strukturerad. Det är Du som skall visa vad Du kan!
I bilagan finns information du kan behöva.

Hjälpmedel

Penna och kautschuk.

Lycka till!

Del 1 (20 poäng)

1. BMI kan användas för att klassificera fetma. Det definieras som $m/(l^2)$, där m är personens vikt och l är personens längd. Betrakta programmet nedanför:

```
def hogst_BMI(person1, person2):
    bmi_person1 = person1[1]/person1[2]**2
    bmi_person2 = person2[1]/person2[2]**2

    if bmi_person1 > bmi_person2:
        print(person1[0] + " har högst BMI: " + \
              str(bmi_person1))

person1 = ["Bob", 100, 2]
person2 = ["Alice", 50, 2]

hogst_BMI(person1, person2)
```

Vad motsvarar de tre elementen i listorna person1 och person2? Vad kommer att skrivas ut på skärmen? **(2p)**

2. Vad kommer att skrivas ut på skärmen när följande kod körs?

```
lista1 = list(range(0,4))
lista2 = []
for i in range(1,len(lista1)):
    lista2.append(lista1[i] + lista1[i-1])
print(lista2)
(2p)
```

3. Skriv ett program som läser in två klockslag på formen HH:MM:SS (som strings). Sedan ska programmet addera dessa två och skriva ut det resulterande klockslaget på skärmen. T.ex om användaren anger 23:30:00 och 01:00:00, så ska 00:30:00 skrivas ut på skärmen. **(2p)**

4. Som en vidareutveckling av uppgift 3. Förbättra koden så att du undersöker om de inmatade klockslagen är på rätt format. Dvs. undersök att ":" är på rätt plats; att strängen är av rätt längd; att siffrorna anger klockslag (de två första siffrorna kan t.ex inte vara "99"). Siffrornas ASCII kod är mellan 45 och 57. **(2p)**

5. Vad skrivs ut på skärmen när följande kod körs? Fås samma utskrift om man byter ut alla förekomster av *tuple* mot *list*?

```
tuple1 = tuple(range(0,6))
tuple2 = tuple(range(5,10))
print(set(tuple1) & set(tuple2))
```

Tips: istället för &-operatorn hade man kunnat använda mängdfunktionen `intersection()`.

(2p)

6. Vad skrivs ut på skärmen när koden nedan körs? Vad gör funktionen `foo`?

```
def foo(lista):
    lista2 = []
    while len(lista) > 0:
        index = 0
        for i in range(0,len(lista)):
            if lista[i] >= lista[index]:
                index = i
        lista2.append(lista[index])
        lista.remove(lista[index])
    return lista2
```

```
print(foo([1,2,3]))
print(foo(["Bob","Alice","Alicia"]))
```

(2p)

7. Skriv en funktion som skapar en mängd med slumpade heltal mellan 1 och 100, och som sedan skriver ut alla de udda talen i mängden till en fil. **(2p)**

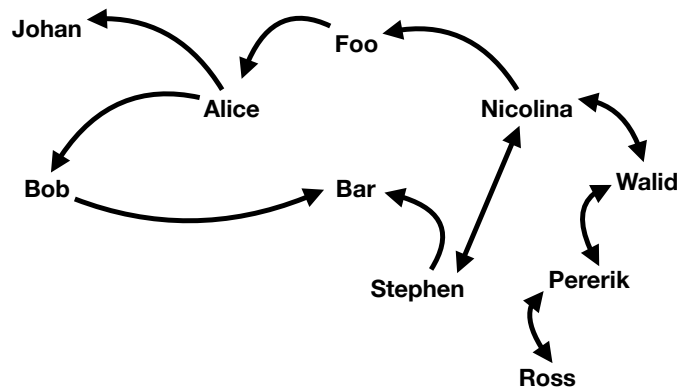
8. Inte över 10! Du ska skriva ett program där du utmanar datorn i ett spel. Det fungerar på följande sätt. Om vartannat så väljer du ett tal och datorn ett tal. Dessa summeras hela tiden i en variabel `summa`. Den som först väljer ett tal så att `summa` överstiger 10 har förlorat. Datorn väljer hela tiden ett slumpmässigt flyttal mellan 0 och 1. Du som användare måste välja ett flyttal mellan 0.5 eller 1

eller välja att avstå från att välja ett tal (i vilket fall du väljer talet 0 i praktiken). **(2p)**

9. I denna uppgift ska du (virtuellt) kasta tärning. Du börjar med att skapa en klass Dice som innehåller en variabel för sidan som är uppåt på tärningen (en tärning har sex sidor: 1,2,3,4,5,6). Klassen ska innehålla en funktion rulla() (rullar tärningen) samt en funktion get_sida_upp() (returnerar sidan på tärningen som är uppåt).

När du har är färdig med klassen så ska du skriva ett program som skapar tre tärningar (instanser av Dice) och kastar dessa tills du får tre sexor (detta görs genom att anropa funktionen rulla()). **(2p)**

10. Sociala nätverk, såsom det nedan, används för att representera människors sociala interaktioner. En pil från en person till en annan person innebär att den första personen känner den andra. Enligt bilden nedan så känner Alice (till) Johan, men Johan känner inte (till) Alice. Walid och Pererik känner båda varandra.



Skapa en Dictionary som representerar det sociala nätverket ovan. Du behöver inte skapa hela denna utan det räcker med tre nyckel-värde par.

(2p)

Del 2 (10 poäng)

Du ska skapa en klass Varuautomat som simulerar en riktig varuautomat (såsom kaffemaskin eller godisautomat).
Specifikationen är:

- Man kan välja att ett antal drycker såsom kaffe, te, och läsk.
- Varje vara finnas av ett visst antal.
- Varje vara ska ha ett pris.
- Det finns en kassa med pengar.
- Man ska kunna välja vara och se pris.
- Om en vara är slut så ska man inte kunna välja den varan.
- Man kan endast betala med pengar (ej kort).
 - Om man betalar för mycket så ska man betala mer eller avbryta köpet.
 - Om man har betalat för mycket och det inte går att få växel tillbaka från kassan, så ska köpet avbrytas och pengarna fås tillbaka.
 - Man kan avbryta köpet efter betalning och få pengarna tillbaka.
 - Om man har betalat tillräckligt kan man beställa varan.
- Vid en lyckad beställning ska varan levereras, kassan ska uppdateras, antalet av den valda varan ska minskas med 1.

Introducera en tillståndsvariabel för varuautomaten. Denna representerar vad varuautomaten gör för tillfället såsom:

- väntar på val av vara,
- väntar på betalning,
- lyckad betalning,
- väntar på beställning,

Denna variabel uppdateras efter varje steg av en beställning och håller reda på var man är i beställningen.

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 är 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(<i>substring</i>)	The <i>substring</i> argument is a string. The method returns true if the string ends with <i>substring</i> .
find(<i>substring</i>)	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(<i>old</i> , <i>new</i>)	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(<i>substring</i>)	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 item. A <code>ValueError</code> exception is raised if item 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 item 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.