

**PROGRAMMERING-PYTHON  
TENTAMINA  
2019-08-16**

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 10 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 Del 2. 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. Skriv ett program där användaren först anger en summa pengar i Euro. Sedan skrivs det ut på skärmen hur många femtiolappar och enkronor detta motsvarar i svenska kronor. Vi antar att en Euro motsvarar elva kronor. **(2p)**
2. Om en boll släpps från höjden  $h$  så kan man approximativt beräkna tiden  $T$  det tar innan den träffar marken enligt formeln

$$T = \sqrt{\frac{h}{5}} .$$

Enheten på  $T$  är sekunder och enheten på  $h$  är meter.

Skriv ett program där användaren anger en höjd över marken samt en gissad tid som det tar för en boll att falla till marken från höjden. Om det tar längre eller kortare tid för bollen att falla än den gissade tiden (enligt formeln ovan) ska detta skrivas ut på skärmen. **(2p)**

3. Vad kommer att skrivas ut på skärmen när följande program körs?

```
c = 0
for a in range(1,3):
    for b in range(1,a):
        c = a + b
print(c)
(2p)
```

4. Vad gör funktionen `foo` nedan? Vad skrivs ut på skärmen när hela programmet körs?

```
def foo(string):
    if not string.isdigit():
        return False
    if string[0] == "0":
        return False
    if string == string[::-1]:
        return True
    else:
        return False

print(foo("121"))
```

**(2p)**

5. Vad gör funktionen *bar* nedan? Vad skrivs ut på skärmen när hela programmet körs? Har du ett kortfattat förslag på förenkling av programmet?

```
def bar(lista1, lista2):
    lista3 = []
    index1 = 0
    index2 = 0
    while index1 < len(lista1):
        if not lista1[index1] in lista2:
            lista3.append(lista1[index1])
            index1 += 1
    while index2 < len(lista2):
        if not lista2[index2] in lista1:
            lista3.append(lista2[index2])
            index2 += 1
    return lista3

print(bar([1,2,3],[3,2,1]))
print(bar([1,3,5],[2,4,6]))
```

**(2p)**

6. Funktionen *encrypt* nedan har till uppgift att kryptera en textsträng innehållande tecknen a-z. Förklara kortfattat hur funktionen fungerar. Skapa, genom att ändra ett enda tecken i koden, en funktion *decrypt* som dekrypterar det som *encrypt* har krypterat.

```
def encrypt(string):
    string2 = ""
    l = len(string)
    for i in string:
        string2 += chr(97 + (ord(i) - 97) % 25)
        l -= 1
    return string2
```

**(2p)**

7. Skriv ett program som först skapar en mängd slumpmässiga heltal och sedan skriver ut alla de jämna talen i mängden till en fil. **(2p)**

8. Skriv en funktion som skapar email-adresser enligt följande specifikation: användaren skriver in sitt förnamn och efternamn på formen "Fornamn Efternamn". Sedan skapas en email-adress på formen "foreft@hh.se", vilket skrivs ut på skärmen. Dvs. de tre första bokstäverna på förnamnet efterföljs av de tre första bokstäverna i efternamnet varefter "@hh.se" följer. (2p)

9. I denna uppgift ska du skapa (virtuellt) en låda som innehåller en mängd bollar. Du skapar en klass Box som representerar lådan med en variabel nr\_bollar som representerar hur många bollar lådan innehåller. Vid skapandet av en instans av Box så ska man kunna ange hur många bollar som finns i lådan. Klassen Box ska innehålla en funktion som returnerar antalet bollar, samt innehålla en funktion för att plocka ur en boll ur lådan.

När du har skapat klassen box, skriv ett program där du skapar en låda (instans av klassen Box) som innehåller 10 bollar. Skriv därefter en while-loop där en boll plockas bort ur lådan i varje iteration tills lådan är tom dvs. det inte finns några bollar kvar.

10. Givet personuppgifterna nedan, skapa en dictionary som representerar en telefonbok för personerna.

<b>Namn</b>	<b>Telefonnummer</b>
Lisa	1122334455
Kalle	1223344551
Johan	2233445511
Walid	2334455112

## Del 2 (10 poäng)

Du ska skapa en klass Kortlek. Klassen ska uppfylla följande specifikation.

- Som standard ska alla 52 spelkort som finns i en vanlig kortlek finnas representerade när en instans av klassen skapas.
  - Man kan dock välja att endast vissa kort ska ingå i kortleken när den skapas.
- Kortet ska ligga i en hög som ska gå att blanda.
  - Det ska gå att ta bort det översta kortet på högen (varvid kortleken inte innehåller det kortet längre).
- Det ska gå att skriva ut vilka kort som finns i kortleken.
- Det ska gå att jämföra två kort och se vilket som har högst värde, t.ex., har en knekt högre värde än en 7:a oavsett färg.
- Det ska gå och jämföra två kort och se om de har samma färg.
- Det ska gå att dela ut kort från kortleken. Funktionen som hanterar detta ska få som input hur många spelare som vill ha kort och hur många kort varje spelare ska ha. Som output returneras de utdelade korten i en lämplig datatyp. Kortet som har delats ut är nu borttagna från kortleken.
- Det ska gå att stoppa in ett kort i kortleken, när detta görs så kontrolleras att kortet inte redan finns i kortleken.

När väl klassen Kortlek har skapats. Simulera ett spel där fyra datorspelare får varsitt kort från en kortlek. Den spelare som har fått högst kort har vunnit. Kortet samt vinnaren skrivs ut på skärmen.

## 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( <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.