



TENTAMEN

Luleå tekniska universitet

Kurskod: D0009E
Kursnamn: Introduktion till programmering
Tentamensdatum: 170513
Skrivtid: 4 timmar
Tillåtna hjälpmedel: Inga

Jourhavande lärare m fullständigt telefonnr: Luleå: Fredrik Bengtsson, 0920492431	Jourhavande lärare m fullständigt telefonnr:
Jourhavande lärare m fullständigt telefonnr:	Jourhavande lärare m fullständigt telefonnr:

Betygsgränser:	
Totalt antal uppgifter och poäng:	7 uppgifter om totalt 37 poäng
Övriga upplysningar:	

Allmänna anvisningar

Kontrollera att du fått samtliga uppgifter.
Besvara endast en uppgift per lösningsblad.
Skriv tydligt.

Efter tentamen

Tentamensresultatet syns "Mina sidor" på Studentwebben.
Resultat meddelas före sista anmälningsdag till nästa tentamenstillfälle.

Uppgifter till tryckeriet för tentor campus Luleå

Projektnummer SRT: 341980	Hur många ex: 35
Hur många sidor: 4	Dubbel eller enkelsidigt: dubbel

Uppgift 1: (8p)

- a:** Kommer följande program att terminera (terminera=avsluta)?
Endast ja/nej-svar krävs.

```
lst = [1, 2, 3, 4, 5, 6]
i = 0
while len(lst)>0:
    lst = lst[1:]
    i = i+1
```

- b:** Vilken utskrift ger följande program:

```
ls = [[], [1], [6, 4, 3], [3, 4], [0]]
b=2
c=5
if len(ls)>c:
    print "Hoho!"
elif len(ls[b])<b and b>c:
    print "Maria"
else:
    print "nej"
```

- c:** Vilken utskrift ger följande program:

```
s=""
i = [1, 2, 3]
while len(i)>0:
    s = s + str(i[0])
    i = i[1:]
    print s,
```

- d:** Vad händer när följande kod exekveras (Om det blir fel, varför blir det fel?
Om det blir rätt, vilket värde får s?):

```
i=int( float( str(int("3")+float("4.3"))+"5" ) )
```

Uppgift 2: (4 p)

Någon har skrivit en funktion, `lessThan(numList, num)`, som tar en lista samt ett tal som argument och returnerar en ny lista med alla tal från den första listan som är mindre än talet i argumentet (`num`). Programmeraren kom fram till följande:

```
def lessThan(numList, num):
    i = 0
    while i<=len(numList):
        if numList[i] < num:
            lst = lst + [numList[i]]
        i = i-1
```

Programmeraren har gjort fel. Uppgiften är att identifiera och rätta felen. Det är inte tillåtet att skriva ett helt nytt program, utan felen i detta befintliga måste rättas.

Uppgift 3: (5p)

Betrakta följande program:

```
def copyOnFull(lst, maxLen):
    if len(lst) >= maxLen:
        return lst[:]
    else:
        return lst

def update(lst, e, pos):
    lst[pos] = e

lst1 = [6, 3, 8, 7]
lst2 = copyOnFull(lst1, 4)
lst3 = copyOnFull(lst1, 5)
update(lst1, 0, 2)

print 'lst1=', lst1
print 'lst2=', lst2
print 'lst3=', lst3
```

Vilka värden får `lst1`, `lst2` och `lst3` efter exekveringen? Ange också vilka listor som är alias för varandra.

Svara på formen:

```
lst1 = ...
lst2 = ...
lst3 = ...
```

där punkterna är ersatta av svaren, samt skriv `lstX=lstY` om `lstX` och `lstY` är alias (X och Y siffror).

Uppgift 4: (5p)

Skriv en funktion `rovasprak`, som tar en sträng, `st`, som argument. Funktionen returnerar en ny sträng som är texten från `st` översatt till rövarspråket. För att översätta till rövarspråket lägger man efter varje konsonant till ett "o" och därefter samma konsonant igen. Övriga tecken är oåverkade. Till vår hjälp finns en global sträng `konsonanter`, som innehåller alla konsonanter.

Körexempel:

```
>>> rovasprak("jag skriver tentamen.")
'jojagogsoskokrorivoverortotenontotamomenon.'
```

Inbygda funktioner eller metoder för strängar får inte användas. Det är tillåtet att använda funktionerna `len`, `str` och `range` samt operatoren `in`. Iteration kan ske på valfritt sätt med `for` eller `while`.

Uppgift 5: (5p)

Givet är följande följande funktioner för att läsa in heltal från fil sam skriva ut på skärmen:

```
def writeInt(name, v):  
    f=open(name, "w")  
    f.write(str(v))  
    f.close()  
  
def writeUI():  
    v = input("Enter value:")  
    writeInt("log.txt", v)
```

a, Modifiera `writeInt` så att den kastar undantaget `IOError` om filen som anges redan existerar. Undantagssystemet ska användas.

b, Modifiera `writeUI` så att "File exists." skrivs ut om filen redan finns. Undantagssystemet ska användas.

Uppgift 6: (5 p)

Vi vill göra en klass `Vector` för att representera vektorer i två dimensioner. Klassen ska ha en initierare som tar vektorns vinkel och längd som argument (två tal). Klassen ska också innehålla metoden `mul`, som multiplicerar vektorn med ett tal som tas som argument till `mul` och returnerar resultatet som en ny vektor. En vektor multipliceras med ett tal genom att längden av vektorn multipliceras med talet, och vinkeln är oförändrad. Dessutom ska metoderna `getAngle` och `getLen` finnas för att returnera vinkeln resp. längden separat.

Uppgift 7: (5p)

Skriv en rekursiv funktion, `remove`, som tar en lista av tal, `lst`, samt ett tal, `e`, som argument och returnerar en ny lista med alla tal från `lst` som inte är lika med `e`.

Exempel:

```
>>> print remove([1,5,3,3,7,3,9], 3)  
[1, 5, 7, 9]
```