



TENTAMEN

Luleå tekniska universitet

Kurskod: D0009E
Kursnamn: Introduktion till programmering
Tentamensdatum: 170324
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:

Om du skriver tentan för första gången: Fyll i och lämna in kursvärdering i en separat hög! (Värderingen kan också lämnas i mitt fack i A-huset efteråt.)

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 anmälningssdag 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:
    print lst[0]
    i = i+1
```

- b:** Vilken utskrift ger följande program:

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

- c:** Vilken utskrift ger följande program:

```
s="abcd"
i = 0
while i<3 and len(s)>2:
    s = s[i:]
    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?):

```
s=str(int( 2+float("3"+"2") ) + 4)
```

Uppgift 2: (4 p)

Någon har skrivit en funktion, `sum(lst)`, som tar en lista som argument och returnerar summan av tal, fast där jämna tal dividerats med två innan summering. Vi antar att listan `lst` innehåller heltal. Programmeraren kom fram till följande:

```
def sum(lst):
    i = 1
    sum = 0
    while i<=len(lst):
        if lst[i]/2==0:
            sum = sum + lst[i]/2
        else:
            sum = sum + lst[i]
```

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 newIfShort(lst, l):
    if len(lst)<1:
        return []
    else:
        return lst

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

lst1 = [6,3,8,7]
lst2 = newIfShort(lst1,lst1[1])
update(lst2, 0, 2)
lst3 = newIfShort(lst2,lst1[2])
```

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 eraseSubst, som tar två strängar, st och subst, som argument. Funktionen returnerar en sträng där alla förekomster av subst tagits bort från st. Det är tillåtet att anta att strängen subst alltid innehåller tre tecken.

Körexempel:

```
>>> eraseSubst("Muppar pruppar upp.", "upp")
'Mar prar .'
```

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 readInt(name):  
    f=open(name, "r")  
    v = int(f.readline())  
    f.close()  
    return v  
  
def printInt():  
    print readInt("testfil.txt")
```

a, Modifiera `readInt` så att den kastar undantaget `IOError` om det inlästa talet inte är ett heltal. Undantagssystemet ska användas.

b, Modifiera `printInt` så att "File error" skrivs ut om filen ej finns eller om talet i filen ej är ett heltal. Undantagssystemet ska användas.

Uppgift 6: (5 p)

Vi vill göra en klass `Complex` för att representera komplexa tal. Klassen ska ha en initierare som tar real- och imaginärdel som argument (två tal). Klassen ska också innehålla metoden `add`, som adderar talet som `add` anropas på samt ytterligare ett tal (en till parameter). Metoden ska returnera ett nytt komplext tal, som är summan av de två talen (real- och imaginärdel summeras separat hos komplexa tal). Dessutom ska metoderna `getReal` och `getImag` finnas för att returnera real- resp. imaginärdelen separat.

Uppgift 7: (5p)

Skriv en rekursiv funktion, `strangeSum`, som tar en lista av tal samt argument och returnerar summan av talen fast där alla jämna tal dividerats med två innan summering.

Exempel:

```
>>> strangeSum([1,2,3,4,5])  
12
```