TENTAMEN

Luleå tekniska universitet

Kurskod: D0009E
Kursnamn: Introduktion till programmering
Tentamensdatum: 180323
Skrivtid: 4 timmar
Tillåtna hjälpmedel: Inga

Jourhavande lärare m fullständigt telefonnr: Luleå: Jingsen Chen, 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:	

Fyll i kursvärderingen (frivilligt, men väldigt uppskattat!) och lämna in i separat hög. (kan även lämnas i Fredrik Bengtssons fack senare, om tidsbrist råder)

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:** 30
Hur många sidor: 4	**Dubbel eller enkelsidigt:** dubbel

I denna kursomgång har python 3 använts. Om denna tenta skrivs som omtenta så får man givetvis använda python 2 istället. Kodexemplen i tentan är python 3. För denna tena är enda skillnaden att `raw_input` *i python 2 heter* `input` *i python 3 och att* `print` *skrivs med parenteser (ex.* `print(a,b,c)`*).*

Uppgift 1: (8p)

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

```
ls=[5,1,9,6,4,5]
for val in ls:
    val = val - 1
    if val>10:
        break
```

b: Vilken utskrift ger följande program:

```
ls=[[6],[5],[4]]
b=3
if b<len(ls[2]) or len(ls)<b:
    print("Första!")
elif ls[1][0]>ls[0][0]:
    print("Andra!")
else:
    print("Tredje!")
```

c: Vilken utskrift ger följande program:

```
s="Anna"
c="a"
n=0
for d in s:
  if d==c:
    n = n+1
  print(n)
```

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("3"+int("3")+2))+"2"
```

Uppgift 2: (4 p)

Någon har skrivit en funktion, `minDiff(ls)`, som tar en lista som argument och returnerar den minsta differensen av två efterföljande element.
Programmeraren kom fram till följande:

```
def minDiff(ls):
  i=0
  minD = 0
  while i<len(ls):
    if (ls[i]-ls[i+1]) < minD:
      minD = ls[i]-ls[i+1]
  print(minD)
```

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 selectShortest(lstA, lstB):
    if len(lstA)>len(lstB):
        return lstB
    else:
        return lstA

def updateFirst(lst, e):
    lst[0] = e

lst1 = [4,6,8]
lst2 = selectShortest(lst1,[3,9])
updateFirst(lst2, 0)
lst3 = selectShortest(lst2,lst1)
```

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 hist, som tar en sträng som parameter och returnerar en katalog (dictionary) med ett histogram över tecknen i strängen. Ett histogram är en förteckning över hur många gånger varke tecken i strängen förekommer i strängen. I katalogen ska tecknet från strängen vara nyckel och värdet ska vara ett heltal med antalet förekomster av nyckeln.

Körexempel:

```
>>> hist("parallell")
{'p': 1, 'a': 2, 'r': 1, 'l': 4, 'e': 1}
```

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

Uppgift 5: (5p)

Givet är följande funktion som skriver en sträng till fil, samt en applikation för att hämta in filnamn och sträng från användaren:

```
def save(fName, sValue):
  f=open(fname,"w")
  f.write(sValue)
  f.close()

def app():
  value = input("value:")
  name = input("filename:")
  # python 2: value=raw_input(...
  save(name, value)
```

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

b, Modifiera `app` så att `File already exists` skrivs ut på skärmen om filen som anges redan existerar. Du får använda `save` från uppgift (a) även om du inte löst uppgift (a).

Uppgift 6: (5p)

Implementera en klass `Logger`, för att representera en log av elemment (en historik). Loggen innehåller ett antal element (från början 0) och har en maxlängd (max antal elmenent) som aldrig får överskridas. Logen ska ha en konstruktor/initierare, som tar logens maxlängd som parameter. Det ska också finnas en metod `add`, som tar ett element som parameter och lägger till i loggen. Om loggen är full (redan har sin maxlängd) så ska ett undantag av typen `ValueError` kastas/ges/genereras. En metod `get` ska returnera ett element ur loggen och tar ett tal som parameter, som representerar vilket element som ska returneras, där 0 är det senast inlagda elementet och stigande tal representerar äldre element. Implementera också en metod `combine`, som tar som parameter ytterligare en log (förutom den combine redan opererar på). `combine` gör samma sak som `add`, fast den lägger till en hel log i slutet på loggen istället för bara ett element (som `add`). Om loggen blir full ska inget göras, utan ett undantag av typen `ValueError` ska kastas/ges/genereras.

Uppgift 7: (5p)

Skriv **en** rekursiv funktion, `flatten`, som tar en lista av listor som argument och returnerar en lista där alla listor från argumentlistan är konkatenerade.
Exempel:

```
>>> flatten([[1,3,2],[4,1],[8,9,3]])
[1, 3, 2, 4, 1, 8, 9, 3]
```