



TENTAMEN

Luleå tekniska universitet

Kurskod: D0009E
Kursnamn: Introduktion till programmering
Tentamensdatum: 180523
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: 25
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 tenta 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.

```
s="xyz"
i = len(s)-1
while i>0:
    print(s[i])
```

- b:** Vilken utskrift ger följande program:

```
s="abc"
n=3
u=[]
if len(u)>n and s=="abc":
    print("abc")
elif len(s)==n or len(u)<0:
    print("string")
else:
    print("list")
```

- c:** Vilken utskrift ger följande program:

```
ls=[1,1,1]
i=0
while i<len(ls):
    ls[i] = ls[i] + i
    print(ls)
    i=i+1
```

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

```
s="3"+str(2.1+float("3.0")+6)
```

Uppgift 2: (4 p)

Någon har skrivit en funktion, countChar(s, c), som tar två strängar som argument och returnerar antalet förekomster av c i s, där c kan antas endast bestå av ett enda tecken.

Programmeraren kom fram till följande:

```
def countChar(s, c):
    num = 1
    while i<=len(s):
        if c==s[i]:
            num = num + 1
    return num
```

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
    return lst

lst1 = [5,3,9]
lst2 = updateFirst(lst1, 0)
lst1 = selectShortest(lst2[1:], lst1)
lst3 = selectShortest(lst1, lst2)
lst3[1] = 7
```

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 `cleanSpaces`, som tar en sträng som argument och byter ut alla förekomster av fler än ett, på varandra följande, mellanslag, mot ett mellanslag.

Körexempel:

```
>>> cleanSpaces("  Vi  rensar onödiga      mellanslag.")
' Vi rensar onödiga mellanslag.'
```

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 funktion som läser in tal från en fil till en lista och returnerar listan, samt en applikation för att hämta in filnamn från användaren:

```
def loadList(fName):
    lst = []
    f=open(fName, 'r')
    while True:
        lst.append(int(readline(f)))
    f.close()
    return lst

def loadUI():
    fname=input("Filename: ")
    # python 2: fname=raw_input("Filename: ")
    lst = loadList_(fname)
    print(lst)
```

a, Modifiera `loadList` enligt följande:

- Lägg till en extra parameter, `defaultList`. Denna lista ska returneras om filen som anges i parametern `fName` ej existerar. Undantagssystemet ska användas.
- Om någon rad i filen inte innehåller ett giltigt heltal ska undantaget `loadError` kastas/genereras.

b, Modifiera `loadUI` enligt följande:

- Texten `load error` ska skrivas ut på skärmen om `loadList` ger undantaget `loadError`. Du får använda `loadList` från uppgift (a) även om du inte löst uppgift (a).
- listan `[1,2,3,4,5]` ska skickas som argument till parametern `defaultList` i den modifierade `loadList`.

Uppgift 6: (5p)

Implementera en klass `Circle`, för att representera cirklar i planet. `log` av element (en historik). Loggen innehåller ett antal element (från början 0) och har en maxlängd (max antal element) som aldrig får överskridas. Loggen ska ha en konstruktor/initierare, som tar loggens 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, `logStar`, som tar ett heltal, `num`, som argument och returnerar $\log^*(\text{num})$. Funktionen \log^* definieras som antalet applikationer av 2-logaritmen som krävs för att talet ska bli mindre än eller lika med 1. T.ex, blir $\log^*(16)=3$ därför att $\log_2(\log_2(\log_2(16)))=1$ och på samma sätt är $\log^*(6)=3$ för att $\log_2(\log_2(\log_2(6)))=0,4543...<1$. 2-logaritmen i python heter `log2`.
Exempel:

```
>>> logStar(16)
3
>>> logStar(6)
3
```