

Tentamen i Introduktion till programmering

Kurskod:	D0009E
Skrivtid:	09:00-13:00 (4 timmar)
Totalt antal uppgifter:	7
Totalt antal poäng:	38
Tentamensdatum:	2013-12-21
Jourhavande lärare:	Fredrik Bengtsson (luleå, endast telefon), tel. 0920-492431, 0738166670
	Daniel Granlund (skellefteå, endast telefon), tel. 0910-585352
Tillåtna hjälpmedel:	Ordbok (språkordbok för svenska och ett annat språk – ej programmering)
Tentamen avser exempel implementerade i programmeringsspråket som använts i kursen, Python, och lösningarna förväntas i samma programmeringsspråk.	

Lycka till!

Uppgift 1: (8p)

a: Kommer följande program att terminera (terminera=avsluta)?

```
i = 3
while (i>0):
    i=i+2
    if i==6:
        break
```

b: Vilken utskrift ger följande program:

```
j=4
i=10
b=j<=i
if b and j-i<0:
    print 15
else:
    if b:
        print 10
    else:
        print 0
```

c: Vilken utskrift ger följande program:

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

d: Vad händer när följande kod exekveras (om det blir fel, varför blir det fel?):

```
s="3"
s=float(str(3+1)+s)+5.0
```

Uppgift 2: (5 p)

Någon har skrivit en funktion, `areUnique(a, b, c)`, som är tänkt att returnera `True` om `a`, `b` och `c` är olika, dvs. inga variabler får ha samma värde. Annars returneras `False`. Programmeraren kom fram till följande:

```
def areUnique(a, b, c):
    if a==b:
        return False
    elif a==c:
        return False
    else:
        return b!=c
```

Programmeraren är fundersam om han/hon gjort rätt. Uppgiften är att identifiera och rätta ev. fel genom att skriva en rättad version av programmet som fungerar i samtliga fall alternativt bevisa att versionen ovan är rätt.

Det kan antas att `a`, `b` och `c` är av typen heltal.

Uppgift 3: (5p)

Vad blir utskriften av följande program?

```
def forkIfFull(lst, size):
    if len(lst)>=size:
        lst[len(lst):len(lst)]=["end"] # append
        return []
    else:
        return lst

l1=["a"]
l2=forkIfFull(l1,2)
l2[0:0]=["b"]
l3=forkIfFull(l2,2)
l3[0:0]=["c"]
print l1,l2,l3
```

Uppgift 4: (5p)

Skriv en funktion `spaceSplit(s)` som tar en sträng som parameter och returnerar en lista innehållande alla ord som förekommer i strängen. Dvs. varje ord som förekommer i strängen ska sparas som en sträng i en lista som sedan returneras.

Det kan antas att orden i strängen avdelas med ett enkelt mellanslagstecken. Mellanslagstecknet ska ej inkluderas i strängen som sparas i listan. Vidare kan antas att inga mellanslagstecken förekommer i början eller slutet av strängen.

Körexempel:

```
>>> spaceSplit("Ordlista")
["Ordlista"]
>>> spaceSplit("Ord lista")
["Ord", "lista"]
>>> spaceSplit("Hej på dig")
["Hej", "på", "dig"]
```

Inbygda funktioner eller metoder för strängoperationer får inte användas. Det är tillåtet att använda funktionerna `len` och `range`. Iteration kan ske på valfritt sätt med `for` eller `while`. `s` kan ha godtycklig längd.

Uppgift 5: (5p)

Betrakta följande kodsnitt:

```
f=open(fname, 'r')
name    = f.readline()
salary  = int(f.readline())
time    = int(f.readline())
print name[:-1]+": "+str(float(salary)/time)
f.close()
```

Identifiera lämpliga parametrar och kapsla in koden som en egen funktion vid namn `readDb`. Modifiera funktionen och lägg till kod så att det, om filen inte existerar, skapas en tom fil med samma filnamn varpå funktionen avslutar. Om filen inte kan läsas på grund av fel datatyp i filen så ska ett felmeddelande skrivas ut: "filen kan ej läsas". Till detta ska undantagshanteringssystemet användas. Felhantering utöver det som nämns i uppgiften behöver inte implementeras.

Uppgift 6: (5 p)

Vi vill göra en klass `Article` som är tänkt att användas för att representera en artikel i ett varulager. Aktuellt lagersaldo för artikeln ska kunna anges vid instansiering av objektet. Vidare ska fyra metoder implementeras; `update(s)` för att uppdatera aktuellt lagersaldo, `get()` för att returnera aktuellt saldo, `lock()` för att låsa artikeln samt `unlock()` för att låsa upp artikeln. Observera att när artikeln är låst så ska ej lagersaldot kunna uppdateras utan bara avläsas. Artikeln ska vara olåst vid instansiering av objektet. Implementera klassen `Article` enligt ovan.

Uppgift 7: (5p)

Skriv en rekursiv funktion, `listCon`, som tar en lista av strängar, `ls`, som argument och returnerar en sträng som är sammanslagningen av samtliga strängar från listan, fast i omvänd ordning (sista strängen från listan först i sammanslagningen) och med ett mellanslag adderat efter varje sträng från listan. Exempel:

```
>>> listCon(["string", "a", "is", "This"])
'This is a string '
```