



# TENTAMEN

Luleå tekniska universitet

Kurskod: D0009E
Kursnamn: Introduktion till programmering
Tentamensdatum: 171222
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: Skellefteå: Patrik Holmlund, 0910585392	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.**

## 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 anmälningdag 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

### Uppgift 1: (8p)

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

```
ls=[5,1,9,6,4,5]
while len(ls)>0:
    ls[0] = ls[0] - 1
```

- b:** Vilken utskrift ger följande program:

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

- c:** Vilken utskrift ger följande program:

```
s="luJdoG"
sn=""
for c in s:
    sn = c + sn
print sn
```

- 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("3.5")+"0.1")
```

### Uppgift 2: (4 p)

Någon har skrivit en funktion, `prodEven(ls)`, som tar en lista som argument och returnerar produkten av alla tal på jämna index från listan. Programmeraren kom fram till följande:

```
def prodEven(ls):
    i=0
    prod = 0
    while i<len(ls):
        prod = prod * ls[2*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 selectAndModify(lst, e):
    if e in lst:
        return lst
    else:
        return lst+[e]

lst1 = [1,2,4]
lst2 = selectAndModify(lst1, len(lst1))
lst1[0] = 3
lst3 = selectAndModify(lst2, len(lst2))
lst2[2] = 0
```

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 `revDel`, som tar en sträng `s`, som argument och returnerar en ny sträng som *motsvarar* att man först tagit bort alla dubletter av samma tecken och endast behållit den *sista* förekomsten samt sedan reverserat strängen (alla tecken i omvänd ordning).

Körexempel:

```
>>> revDel("ulouJ doG")
'God Jul'
```

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

### Uppgift 5: (5p)

Givet är följande funktion som läser in heltal från varje rad i en fil, adderar dessa och returnerar summan:

```
def fileSumInt():
    sum = 0
    f= open("test")
    while True:
        line = f.readline()
        if line == "":
            f.close()
            return sum
        sum = sum + int(line)
```

Givet är också följande användargränssnitt till funktionen:

```
def fileSumUI():
    print fileSumInt_(fileName)
```

**a,**

- Modifiera `fileSumInt` så att den kastar/ger undantaget `ReadError` om filen som läses från ("test") inte existerar. Undantaget `ReadError` kan antas existera.
- Modifiera också funktionen så att summeringen slutar och summan returneras vid första raden som inte innehåller ett giltigt heltal. Undantagssystemet ska användas.
- Modifiera även så att filnamnen för filen parametrar till `fileSumInt`.

**b,** Modifiera `fileSumUI` så att filnamn att läsa från efterfrågas från användaren. Modifiera också så att "read error" skrivs ut om filen inte existerar.

Undantagssystemet ska användas och den nya, modifierade `fileSumInt` ska användas för all filhantering. Det är tillåtet att använda `fileSumInt` som om uppgift (a) är löst, även om du inte löst uppgift (a).

### Uppgift 6: (5 p)

Vi vill göra en klass `limitInt` för att representera heltal,  $n$ , större än 0 men mindre än  $m$ ,  $0 < n < m$ . Denna klass ska ha en initierare som tar  $n$  och  $m$  som beskrivits. Om  $n \geq m$  ska undantaget `ValueError` genereras vid skapande av objektet. Det ska också finnas två funktioner `getVal` och `getLimit`, som returnerar  $n$  resp.  $m$ .

Vidare ska en metod för addition, `add`, implementeras. Den ska använda talet som objektet den anropas på representerar samt ytterligare ett tal (ett annat objekt av samma klass som tas som argument) och returnera resultatet av additionen av  $n$  från resp. objekt. Resultatet ska vara ett objekt av samma klass och med samma  $m$ . Om resultatet är större än eller lika med  $m$  ska undantaget `ValueError` genereras (samma beteende som initieraren). Om  $m$  skiljer sig åt mellan de två adderade objekten ska samma undantag också genereras.

### Uppgift 7: (5p)

Skriv en rekursiv funktion, `revDelRec`, som fungerar exakt som lösningen till uppgift 4.

Exempel:

```
>>> revDelRec("ulouJ doG")
'God Jul'
```