

Regler

- Student får lämna salen tidigast en timme efter tentans start.
- Vid toalettbesök eller rökpaus ska pauslista utanför salen fyllas i.
- All form av kontakt mellan studenter under tentans gång är strängt förbjuden.
- Böcker och anteckningssidor kan komma att granskas av tentavakt i samband med tentans start samt under tentans gång.
- Frågor om specifika uppgifter eller om tentan i stort ska ställas via tentasystemet.
- Systemfrågor kan ställas till assistent i sal.
- Ingen uppgift rättas efter tentatidens slut.
- Ingen kompletteringsmöjlighet ges de sista tio minuterna.
- Så länge en uppgift inte har betyget U kan den kompletteras till godkänt.
- Inga elektroniska hjälpmedel får medtas. Mobiltelefon ska vara avstängd och ligga i jacka eller väska.
- Inga ytterkläder eller väskor får förvaras vid skrivplatsen.

Antal uppgifter	5
Antal sidor (exklusive denna)	5
Hjälpmedel	En pythonbok (tex Learning Python 4 ed.) En A4-sida med egna anteckningar

VÄND!

Betygssättning

Tentan innehåller fem uppgifter. För godkänt betyg krävs minst två avklarade uppgifter. För gränser för högre betyg, se i tabell 1.

Tid (timmar)	Antal uppgifter	Betyg
3	3	5
4	4	5
4	3	4

Tabell 1: Betygssättning

Bonus från labserien

Bonus från labserien gäller endast vid förstagångstentamen och går därför inte att använda vid detta tillfälle.

Information

Inloggning

Logga in på tentakontot med användaruppgifter som står på tavlan.

Följ menyvalen så långt det går tills du ska mata in ett engångslösenord. Tag fram ditt LiU-kort och visa det för tentavakten för att få detta lösenord.

När du är inloggad är det viktigt att du startar tentaklienten genom att högerklicka på bakgrunden och välja “tentaklient” i menyn.

Avslutning

Tryck på knappen märkt “exit” i menyn längst nere på skärmen och välj ok. Vänta ett tag och tryck sedan på knappen “Avsluta tentamen” när det är möjligt. När detta är gjort är det omöjligt att logga in igen.

Uppgift 1

IPv6-adresser är i grunden skrivna som åtta 16-bitars hexadecimala tal separerade med kolon, dvs åtta grupper av fyra hexadecimala siffror (0-9 samt a-f). För att förenkla skrivandet kan inledande nollor i varje grupp ignoreras samt upprepade grupper med värdet noll kan ersättas med två kolon i rad. Den sista regeln kan dock endast utnyttjas en gång i adressen då det annars skulle leda till tvetydligheter. Nedan kommer samma adress formaterad på tre olika sätt enligt reglerna ovan:

- 2001:0dba:0000:0000:0000:0000:3257:9652
- 2001:dba:0:0:0:0:3257:9652
- 2001:dba::3257:9652

Din uppgift är att skapa en funktion `pad` som tar emot en sträng som motsvarar en IPv6-adress formaterad på något av de giltiga sätten och returnerar en sträng på standardformat, dvs åtta grupper om fyra tecken.

Körexempel 1

```
>>> pad('::1')
'0000:0000:0000:0000:0000:0000:0000:0001'
>>> pad('abcd::1:0:0:a')
'abcd:0000:0000:0000:0001:0000:0000:000a'
>>> pad('2001:0dba:0000:0000:0000:0000:3257:9652')
'2001:0dba:0000:0000:0000:0000:3257:9652'
```

Uppgift 2

Din farbor James bor i USA. Då James vet att du är matlagningsintresserad har han skickat sina favoritrecept till dig (det finns några i mappen `given_files/recipes`). Då du pratar ofta med din farbor har du inga problem med språket, men enheterna är något värre, de är ju i amerikanska mått! Därför ska du här göra ett program som översätter ingredienslistan för alla recept som anges på kommandoraden när ditt program startas. För varje fil som anges ska en ny fil med samma namn samt filändelsen `.sv` skapas där resultatet sparas. Vid omvandlingen ska värdena i tabell 2, 3 och 4 användas. Vikter ska avrundas till hela gram och volymmått ska anges med så stora enheter som möjligt, t.ex ska en volym motsvarande 1208ml anges som "1l, 2dl, 1tsk, 3krm". Vikt ska alltid anges i gram (g), volymer ska anges som minst i hela kryddmått.

Du kan anta att filerna som ska översättas alltid ligger i den mapp där programmet körs samt att resultatfilerna ska sparas i samma katalog.

Enhet	Förkortning	Volym (ml)
teaspoon	tsp	4.93
tablespoon	tbsp	14.79
fluid ounce	fl.oz	29.57
cup	C	236.59
pint	pt	473.18
gallon	gal	3785.41

Tabell 2: Amerikanska volymenheter

Enhet	Förkortning	Volym (ml)
kryddmått	krm	1
tesked	tsk	5
matsked	msk	15
deciliter	dl	100
liter	l	1000

Tabell 3: Svenska volymenheter

Enhet	Förkortning	Vikt (g)
ounce	oz	28.35
pound	lb	453.6

Tabell 4: Amerikanska viktenheter

Du kan anta att alla indatafiler följer följande format (se de givna filerna för exempel):

```

Titel / beskrivning (en rad)
===== (25 st ==tecken)
Ingredienser, en eller flera rader på följande format:
<mått> <enhet> <beskrivning>
=====
Receptbeskrivning (lämnas oförändrad)
```

Uppgift 3

När man var yngre var det häftigt att skapa egna språk. Ett vanligt låtsasspråk är rövarspråket som uppfanns av Astrid Lindgren för Kalle Blomkvist där varje konsonant dubblas och ett o sätts in mellan dem. Vokaler och andra tecken lämnas oförändrade. Exempelvis blir det svenska ordet “apa” apopa på rövarspråket.

Din uppgift här är att skapa ett program som ber användaren mata in en fras (en rad) på rövarspråket och sedan ska programmet skriva ut frasen på svenska.

Körexempel 1

Mata in en fras på rövarspråket: **hohejoj jojagog hohtoteror Anondoderorsos!**

Frasen på svenska: hej jag heter Anders!

Körexempel 2

Mata in en fras på rövarspråket: **Anonnona kokokokaror kokafoffofe**

Frasen på svenska: Anna kokar kaffe

Uppgift 4

Talet 999 är lite speciellt. Om man tar ett slumpmässigt valt tal, tex 4761 och multiplicerar detta tal med 999 får man så klart en produkt, i detta fall 4756239. Delar man upp produkten i tresiffriga tal bakifrån får man i detta fall talen 239, 756 och 4. Tar man summan av de tresiffriga talen kommer man, enligt en välkänd populärvetenskaplig tidsskrift, alltid att få summan 999, oavsett ursprungstal! Gör ett program som slumpar ut fem tal i intervallet $[1, 10000]$ och för varje tal skriver ut produkten när man multiplicerar talen med 999 samt kontrollerar att summan faktiskt blir 999 enligt tabellen i körexemplet (där så klart ursprungstalet ska vara slumpat). Observera att det finns tal i intervallet som inte ger summan 999, tex ursprungstalet 1001 som ger summan 1998.

Körexempel 1

Slumptal	Produkt	Summering
36	35964	$964+35=999$
190	189810	$810+189=999$
29	28971	$971+28=999$
1045	1043955	$955+043+1=999$
4806	4801194	$194+801+4=999$

Uppgift 5

Du ska skapa ett program som samsorterar en eller flera filer innehållandes heltal. Varje fil innehåller en sorterad mängd positiva heltal och filernas namn ges på kommandoraden. Resultatet ska sedan skrivas ut i terminalfönstret. I mappen `given_files` finns det datafiler med namn på formen `data*.txt` som du kan testa programmet med.

Krav: Du får inte använda dig av den inbyggda funktionen `sorted` utan ska göra en samsortering, dvs använd dig av att talen i filerna alltid är i stigande ordning.

Körexempel 1

```
zaza2: ./merge.py data10.txt data5.txt
Merging "data10.txt" and "data5.txt"...
Done, results:
1 4 10 16 24 104 112 153 285 523 892 1023 1121 2343 2543
```