

Laboration 3

Syfte:

- Lära dig att översätta strukturdiagram till styrstrukturer i C. Träna på att strukturera program. Tips: i Visual Studio kan man markera ett kodområde och från menyn välja *Edit - Advanced - Format Selection*, så formateras koden enligt standarden. (Din kod måste vara utan kompileringsfel).

Redovisning:

- Skriv en sammanfattning av ditt arbete. Lämna in på IT's i pdf-format. Du kan lämna in individuellt eller tillsammans med (max) en kamrat. Sammanfattningen ska innehålla punkterna:
 - ❖ Programkoden för uppgift 5.
 - ❖ Resultat (vad du lärde dig).
 - ❖ Besvara följande frågor:
 - Vilken fördel är det att deklarera konstanter med *const* i stället för med *define*?
 - Vilket är det minsta antal "varv" som en förtestad upprepning behöver utföras?
 - Vilken typ av värde kan användas som "test expression" i en switch-sats?
 - ❖ **Tips:**
 - Dokumentera ditt arbete under tiden du tar dig igenom anvisningarna.

När man ska göra olika val beroende av ett enskilda värde, arbetar man sig från ena änden av talaxeln till den andra. Om gränserna är 5 och 1000, kollar man först 5 och sedan 1000, aldrig 2 värden i samma logiska uttryck.

Uppgifter

1. Om en triangel har sidorna x , y och z får man triangelns area med Herons formel:
$$\text{arean} = \sqrt{s(s-x)(s-y)(s-z)}$$
där $s = (x+y+z)/2$ (triangelns halva omkrets).
Konstruera ett program som frågar efter en triangelns sidor och sedan räknar ut triangelns area. Programmet ska kontrollera att sidornas längder verkligen kan bilda en triangel.
2. Skriv de satser som räknar ut skatten för en fastighet med ett visst taxeringsvärde enligt nedanstående regler.
 - 2% för taxeringsvärde $\leq 450\,000$ kr
 - 4% för **delen** mellan 450 000 och 600 000 kr
 - 6% för **delen** mellan 600 000 och 750 000 kr
 - 8% för **delen** över 750 000 krAlla tal här ovan ska definieras som konstanter (använd *const*). Namnen på konstanterna ska vara beskrivande och inte vara låsta till det numeriska värdet. 2% kan t.ex. heta TAXVALUE1 och inte TWOPERCENT. Man ska kunna ändra en konstants värde utan att behöva ändra dess namn.
Testvärden:
 - 500 000 ger 11 000 i skatt
 - 800 000 ger 28 000 i skattAnvänd bara enkla villkor (inga "AND" eller "OR"). Använd ditt strukturdiagram från en tidigare övning. Prova gärna att låta JSP-editorn generera koden.

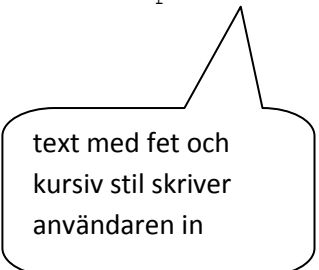
3. Talet 153 har den egenskapen att summan av kuberna på talets siffror är lika med talet:
 $1^3+5^3+3^3=1+125+27=153$
Skriv programsatser som undersöker alla tal mellan 100 och 999 och skriver ut talen med denna egenskap. Använd / och % för att få fram siffrorna i ett tal.
4. Du ska skriva ett program som låter användaren gissa vilket tal datorn "tänker på". När du startat programmet kan du försöka gissa talet. Det kan se ut ungefär så här:

```
*****
*                               *
*   Välkommen till programmet Gissa tal!!!   *
*                               *
*****

Vilket är det största tal som datorn ska "tänka" på? 10
Vilket tal tänker jag på? 3
Det var för litet, försök igen.

Vilket tal tänker jag på? 9
Det var för stort, försök igen.

Vilket tal tänker jag på? 6
Grattis! Det var rätt på 3 försök.
```



5. Skriv ett program som frågar efter ett födelseår. Programmet ska bara acceptera årtal mellan 1900 och 2013. Om användaren skriver något annat tal ska programmet be användaren att skriva in ett nytt årtal. Använd eftertestad upprepning (användaren måste ju skriva ett värde innan vi kan testa det). Alltså ungefär så här:
upprepa
 skriv ut "När är personen född? "
 läs in födelseåret
så länge som årtalet är felaktigt.

Programmet ska även kontrollera att användaren bara får 5 försök på sig. När upprepningen är klar ska programmet skriva ut om årtalet godkändes eller inte.