

Örebro universitet  
Institutionen för teknik  
[Thomas Padron-McCarthy \(thomas.padron-mccarthy@oru.se\)](mailto:thomas.padron-mccarthy@oru.se)

# Tentamen i

## Programmering grundkurs och Programmering C

för D1 m fl, även distanskursen

fredag 28 maj 2010

Gäller som tentamen för:

DT1016 Datateknik A, Programmering grundkurs, provkod 0100  
DT1007 Datateknik A, Tillämpad datavetenskap, provkod 0410  
DT1006 Datateknik A, Programmering C, distans, provkod 0100

---

<b>Hjälpmedel:</b>	Inga hjälpmedel.
<b>Poängkrav:</b>	Maximal poäng är 35. För godkänt betyg (3 respektive G) krävs 17 poäng.
<b>Resultat och lösningar:</b>	Meddelas via e-post eller på kursens hemsida, <a href="http://www.aass.oru.se/~tpy/c/2009-2010-p2/">http://www.aass.oru.se/~tpy/c/2009-2010-p2/</a> , senast fredag 18 juni 2010.
<b>Återlämning av tentor:</b>	Efter att resultatet meddelats kan tentorna hämtas på universitetets centrala tentamensutlämning i L1506, måndag till torsdag kl 10-14.
<b>Examinator och jourhavande:</b>	Thomas Padron-McCarthy, telefon 070-73 47 013.

---

- Skriv tydligt och klart. Lösningar som inte går att läsa kan naturligtvis inte ge några poäng. Oklara och tvetydiga formuleringar kommer att misstolkas.
  - Skriv den personliga tentamenskoden på varje inlämnat blad. Skriv *inte* namn eller personnummer på bladen.
  - Skriv bara på en sida av papperet. Använd inte röd skrift.
  - Antaganden utöver de som står i uppgifterna måste anges.
  - Skriv gärna förklaringar om hur du tänkt. Även ett svar som är fel kan ge poäng, om det finns med en förklaring som visar att huvudtankarna var rätt.
- 

LYCKA TILL!

## Prioritet och associativitet hos operatorerna i C

De viktigaste operatorerna:

Prioritet	Kategori	Operator	Associativitet
Högsta	Unära postfixoperatorer	(), [], ->, ., ++, --	vänster
	Unära prefixoperatorer	!, ++, --, +, -, *, &, sizeof, ( <i>typ</i> )	höger
	Multiplikation mm	*, /, %	vänster
	Addition mm	+, -	vänster
	Jämförelser	<, <=, >=, >	vänster
	Likhetsjämförelser	==, !=	vänster
	Logiskt OCH	&&	vänster
	Logiskt ELLER		vänster
Lägsta	Tilldelning	=, +=, -=, *=, /=, %=	höger

## Uppgift 1 (1 p)

Vilka värden har följande uttryck?

a)  $1 + 2 * 3 - 4$

b)  $1 + 2 * 3 / 4$

c)  $1 + 2 / 3 * 4$

## Uppgift 2 (1 p)

**alpha**, **bravo** och **charlie** är variabler av typen **int**. Vilka värden har variablerna efter att följande kod har körts?

```
alpha = 3; bravo = 2; charlie = 1;
while (alpha > bravo) {
    if (alpha > charlie)
        charlie = charlie + 2;
    else
        alpha = alpha + 1;
        bravo = bravo + 1;
}
```

## Uppgift 3 (1 p)

Vi ska simulera ett radionät, där man skickar meddelanden mellan radioapparater.



Uppgifter om de olika radioapparaterna ska lagras i poster ("structar") av typen **struct Radio**. En radio-post innehåller radions **sändareffekt** (som mäts i watt), dess **frekvens**

(som mäts i MHz), dess **känslighet** (som mäts med ett tal mellan noll och ett), och en position som bestäms av de två koordinaterna **longitud** och **latitud** (som är tal mellan -180 och +180). Allt är flyttal.

Definiera posttypen **struct Radio**.

## Uppgift 4 (1 p)

Definiera en variabel av typen **struct Radio** och initiera den med data om en radio med sändareffekten 1.5 W, frekvensen 75.95 MHz, känsligheten 0.13, och som befinner sig i positionen latitud 59.25393 och longitud 15.24727.

## Uppgift 5 (2 p)

Vi vill kunna visa radioposternas innehåll på skärmen. Skriv en funktion som heter **visa\_radio**, som skriver ut data om ett radio. Funktionen ska ta en radiopost (eller, om du vill, en pekare till den) som parameter. Här är ett exempel på hur utskriften skulle kunna se ut:

```
Effekt: 1.5 W
Frekvens: 75.95 MHz
Känslighet: 0.13
Position: lat 59.25393, long 15.24727
```

## Uppgift 6 (2 p)

Skriv en funktion som heter **las\_radio**, och som läser in data om ett radio. Funktionen ska skriva ut lämpliga ledtexter på standardutmatningen, och läsa in data från standardinmatningen (som normalt är kopplad till tangentbordet).

Du får själv välja om du vill att funktionshuvudet ska se ut så här:

```
struct Radio las_radio()
```

eller så här:

```
void las_radio(struct Radio *p)
```

I den här och alla andra uppgifter på tentan gäller:  
Normalt är felhantering en stor del av ett program. Vad ska till exempel hända om användaren skriver **Kalle** när hon egentligen borde mata in ett tal? Här behövs dock ingen felhantering, om så inte särskilt efterfrågas i uppgiften.

## Uppgift 7 (2 p)

Jorden är rund, så om man vill räkna ut avståndet mellan två punkter på jordytan som man angett med deras latitud och longitud, räcker det inte med enkel plangeometri och Pythagoras sats. Men om man håller sig i närheten av Örebro är det här en bra approximation:

$$\text{avstånd} = \sqrt{(57.03 \cdot (\text{long}_1 - \text{long}_2))^2 + (111.40 \cdot (\text{lat}_1 - \text{lat}_2))^2}$$

**lat<sub>1</sub>** och **long<sub>1</sub>** är den första punktens position, och **lat<sub>2</sub>** och **long<sub>2</sub>** den andras. Avståndet anges i kilometer.

Skriv en funktion som heter **avstand**, som tar två radioposter som argument, och som returnerar avståndet (i kilometer) mellan de två radioapparaterna. Vi antar från och med nu att alla radioapparater finns i närheten av Örebro.

## Uppgift 8 (1 p)

Vi förenklar lite, och antar att två radioapparater kan kommunicera med varandra om de är mindre än fem kilometer från varandra. Skriv en funktion som heter **kan\_kommunicera**, som tar två radioposter som argument, och som returnerar ett sant eller falskt värde beroende på om radioapparaterna kan kommunicera eller inte. Anropa funktionen **avstand** för att räkna ut avståndet.

I den här och alla andra uppgifter på tentan gäller:  
Om du ska använda något från en tidigare uppgift eller deluppgift, till exempel anropa en funktion som skrevs i den tidigare uppgiften, så behöver du inte skriva samma kod igen. Du får också göra uppgiften även om du inte gjort den tidigare uppgiften.

## Uppgift 9 (3 p)

Skriv en main-funktion som har två lokala variabler av typen **struct Radio**, och som läser in data om två radioapparater till dessa variabler med hjälp av funktionen **las\_radio**. Därefter ska programmet tala om ifall de två radioapparaterna kan kommunicera eller inte. (Anropa funktionen **kan\_kommunicera**.)

## Uppgift 10 (5 p)

Vi vill lagra data om radioapparater på en fil. Skriv ett C-program som upprepat läser in radioapparater med hjälp av funktionen `las_radio`, och sparar alla inlästa radioapparater på en fil. Om filen inte kan öppnas, ska ett felmeddelande skrivas ut och programmet avslutas. Välj själv hur inmatningen ska avslutas.

## Uppgift 11 (5 p)

Skriv ett C-program som först läser in data om en radioapparat med hjälp av funktionen `las_radio`, och sen talar om hur många av radioapparaterna på filen från uppgiften ovan som den radioapparaten kan kommunicera med. Om filen inte kan öppnas, ska ett felmeddelande skrivas ut och programmet avslutas.

## Uppgift 12 (5 p)

Skriv ett C-program som upprepat läser in två temperaturer (som anges i grader och kan ha decimaler), och kontrollerar att dessa inte skiljer sig åt med mer än 2.5 grader. Är differensen mellan temperaturerna större ska ett felmeddelande skrivas ut innehållande den aktuella differensen. Annars skriver programmet att temperaturerna är ok. Programmet ska avslutas om båda temperaturerna är noll.

Ett körexempel, med användarens inmatning understruken:

```
Ge två temperaturer: 3.4 2.9
Temperaturerna OK!
Ge två temperaturer: 16.4 20.2
Differensen är 3.8 grader! För hög!
Ge två temperaturer: 0.0 0.0
```

## Uppgift 13 (6 p)

Skriv ett program som läser in personnummer på formen **ÅÅÅÅMMDD-SSSS**, med ett personnummer per rad, ända tills användaren matar in en tom rad. Alla personer är födda efter år 1900, och ingen är född efter år 2010. Därefter ska programmet ange vilket år som de flesta av de inmatade personerna fyller år. Exempel på hur en programkörning skulle kunna se ut, med användarens inmatning understruken:

```
Ange ett personnummer (avsluta med tom rad): 19631211-1658
Ange ett personnummer (avsluta med tom rad): 19451019-2270
Ange ett personnummer (avsluta med tom rad): 19631214-3360
Ange ett personnummer (avsluta med tom rad): 19851001-2273
Ange ett personnummer (avsluta med tom rad):
Flest personer var födda 1963.
```

---