Örebro universitet
Akademin för naturvetenskap och teknik
Thomas Padron-McCarthy (Thomas.Padron-McCarthy@oru.se)

Tentamen i

Programmering grundkurs och Programmering C

för D1 m fl, även distanskursen

lördag 7 mars 2009 kl 8:15 - 13:15

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

Hjälpmedel:	Inga hjälpmedel.
Poängkrav:	Maximal poäng är 40. För godkänt betyg (3 respektive G) krävs 20 poäng.
Resultat och lösningar:	Meddelas via e-post eller på kursens hemsida, http://www.aass.oru.se/~tpy/c/2008-2009-p2/, senast lördag 28 mars 2009.
Återlämning av tentor:	Efter att resultatet meddelats kan tentorna hämtas på institutionen. Man kan också få sin rättade tenta hemskickad.
Examinator och jourhavande:	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, (*typ*)	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)

i är en heltalsvariabel. **x** och **y** är flyttalsvariabler. Ange värdet på i, x och y när följande kod har körts.

```
x = 0;
y = x;
i = 10;
while (x < i) {
    i = i - 2;
    x = x + 1;
}
```

Uppgift 3 (1 p)

a och **b** är heltalsvariabler. **p1** och **p2** är variabler av typen **pekare till heltal**. Ange värdet på a och b när följande kod har körts.

```
a = 1;
b = 2;
p1 = &a;
p2 = p1;
*p2 = 3;
p2 = &b;
*p2 = *p2 + *p1;
```

Uppgift 4 (1 p)

Skriv en funktion som heter **max**, som tar två flyttal av typen **double** som argument, och returnerar det största av de två talen.

Uppgift 5 (1 p)

Skriv en main-funktion som anropar funktionen **max** från uppgiften ovan, och skriver ut resultatet.

main-funktion måste skicka med lämpliga argument i anropet, så att den anropade funktionen kan returnera ett meningsfullt svar. Välj själv var värdena som skickas med ska komma ifrån: inmatning från användaren, konstanter, eller kanske något annat.

> 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 6 (3 p)

Skriv en funktion som heter **array_min_positive**, som tar en array ("vektor") med flyttal av typen **double** som argument, tillsammans med ett heltalsargument som anger antalet tal i arrayen, och returnerar det minsta positiva talet i arrayen.

Exempel: Av de fem talen -1000, -1, 0, 3 och 7 är det 3 som är det minsta positiva talet.

Vi behöver inte bry oss om att hantera specialfallet att det inte finns några positiva tal alls i arrayen.

Uppgift 7 (2 p)

Skriv en main-funktion som anropar funktionen **array_min_positive** från uppgiften ovan, och skriver ut resultatet.

main-funktion måste skicka med lämpliga argument i anropet, så att den anropade funktionen kan returnera ett meningsfullt svar. Välj själv var värdena som skickas med ska komma ifrån: inmatning från användaren, konstanter, eller kanske något annat.

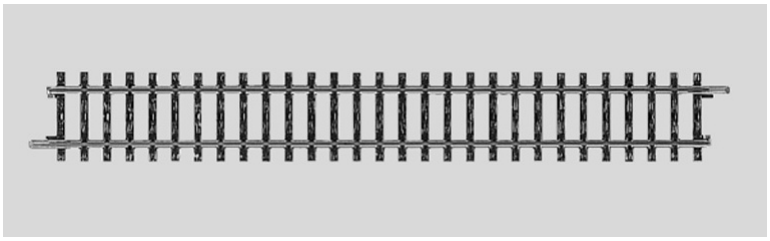
Uppgift 8 (2 p)

Skriv ett komplett C-program (med **#include** och allt) som frågar användaren efter tre namn, och sen, när inmatningen är klar, skriver ut de tre namnen igen, i samma ordning som de matades in.

Ett namn kan innehålla högst tio tecken (som **Anna-Karin**), men innehåller inga mellanslag.

Scenario till uppgift 9-16

Modelltågstillverkaren Märklin, som bland annat tillverkar avancerade datorstyrda modelljärnvägar, ligger lite illa till ekonomiskt. För att spara pengar har de avskedat alla sina programmerare. Styrprogrammen för de datorstyrda modelljärnvägarna ska nu i stället skrivas av tentander i grundkurser i C-programmering.



En modelljärnväg kan innehålla tusentals rälsbitar, och vår uppgift är att skriva programkod för att hantera de rälsbitarna. Varje rälsbit i en modelljärnväg har ett eget, unikt nummer, som ska lagras som ett heltal. Den har dessutom en längd, som anges med ett flyttal och mäts i centimeter.

Uppgift 9 (1 p)

Definiera posttypen **struct Ralsbit**, som ska användas för att lagra data om en rälsbit.

Uppgift 10 (1 p)

Definiera en variabel av typen **struct Ralsbit** och initiera den med rälsbit nummer **4711**, som är **27.1** cm lång.

Uppgift 11 (1 p)

Vi vill kunna visa rälsbitposternas innehåll på skärmen. Skriv en funktion som heter **visa_ralsbit**, som skriver ut en rälsbit på skärmen. Funktionen ska ta rälsbitsposten som parameter. Exempel på hur en utskrift skulle kunna se ut:

```
Nummer: 4711
Längd: 27.1 cm
```

Uppgift 12 (2 p)

Skriv en funktion som heter **las_ralsbit**, och som läser in data om en rälsbit. 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 Ralsbit las_ralsbit()
```

eller så här:

```
void las_ralsbit(struct Ralsbit *p)
```

Uppgift 13 (3 p)

Skriv en main-funktion som har tre (3) lokala variabler av typen **struct Ralsbit**, och som läser in data om tre rälsbitar till dessa variabler med hjälp av funktionen **las_ralsbit**. Avslutningsvis ska den använda funktionen **visa_ralsbit** för att skriva ut data om den *kortaste* av de tre rälsbitarna.

Uppgift 14 (4 p)

Skriv ett C-program som upprepat läser in rälsbitar, med hjälp av funktionen **las_ralsbit**, och sparar dem på en fil. Programmet ska läsa rälsbitar ända tills användaren avslutar genom att mata in en rälsbit som antingen har numret noll eller längden noll. Välj själv om det ska vara en text- eller en binärfil.

Uppgift 15 (6 p)

Skriv ett C-program som läser alla rälsbitarna från filen ovan, och sen skriver ut följande uppgifter:

- antalet rälsbitar
- rälsbitarnas sammanlagda längd
- rälsbitarnas genomsnittliga längd
- numret och längden på den kortaste rälsbiten
- numret och längden på den längsta rälsbiten

Uppgift 16 (4 p)

En *kopplingsbit* är en mojäng som används för att koppla samman två rälsbitar. Posttypen **struct Kopplingsbit** ska användas för att lagra data om en kopplingsbit. Det enda vi behöver lagra i en kopplingsbitspost är vilka två rälsbitar som den kopplingsbiten kopplar ihop, och det gör vi genom att posten innehåller två pekare till de två hopkopplade rälsbitarna.

a) (1p) Definiera posttypen **struct Kopplingsbit**.

b) (1p) Definiera funktionen **kopplahop**, som tar tre argument: en pekare till en kopplingsbitspost och två pekare till två olika rälsbitsposter. Funktionen ska göra de ändringar som behövs för att ange att de två rälsbitarna är hopkopplade.

c) (2p) Skriv en main-funktion läser in data om två rälsbitar med funktionen **las_ralsbit**, och sen kopplar ihop dem med **kopplahop**. Glöm inte de lokala variabler som behövs.

Uppgift 17 (6 p)

Skriv ett C-program som läser in rader, ända tills användaren matar in en tom rad. När inmatningen är slut ska programmet skriva ut de av raderna som innehåller minst en siffra.

Vi kan anta att ingen rad i inmatningen kommer att vara längre än 487 tecken, och att man aldrig kommer att mata in fler än 1019 rader.
