

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

Tentamen i  
**Kompilatorer och interpretatorer**  
*(gamla kursen)*  
för Dataingenjörsprogrammet m fl

måndag 5 november 2012

Gäller som tentamen för:  
DT3004 Datateknik C, Kompilatorer och interpretatorer, provkod 0100

---

<b>Hjälpmedel:</b>	Inga hjälpmedel.
<b>Poängkrav:</b>	Maximal poäng är 37. För godkänt betyg (3 respektive G) krävs 18 poäng.
<b>Resultat:</b>	Meddelas på kursens hemsida eller via e-post senast måndag 26 november 2012.
<b>Återlämning av tentor:</b>	Efter att resultatet meddelats kan tentorna hämtas på universitetets centrala tentamensutlämning.
<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!

## Uppgift 1: Faser (6 p)

En kompilators arbete brukar delas in i ett antal *faser*. Ange vilka det är, och förklara kort vad varje fas gör. Vad är in- och utdata från respektive fas?

## Uppgift 2: Aktiveringsposter (4 p)

Vad är en aktiveringspost?

Vad används den till?

Hur kan den se ut?

Var i minnet placeras den?

Hur många aktiveringsposter har man i ett typiskt program?

## Uppgift 3: Grammatiker (9 p)

a) (3p) Förklara och visa med exempel vad som menas med vänsterrekursion, och på vilket vis det kan vara ett problem. Hur kan man lösa problemet? Visa med exempel!

b) (3p) Förklara och visa med exempel vad som menas med FIRST()-konflikter, och på vilket vis de kan vara ett problem. Hur kan man lösa problemet? Visa med exempel!

c) (3p) Förklara och visa med exempel vad som menas med en tvetydig grammatik, och på vilket vis det kan vara ett problem. Hur kan man lösa problemet? Visa med exempel!

## Uppgift 4: Parsning (7 p)

a) Beskriv vad som menas med en prediktiv recursive-descent-parser. Vilka fördelar och nackdelar har en sådan, jämfört med andra tekniker för att skapa en parser?

b) Visa med ett exempel hur man skriver en prediktiv recursive-descent-parser. Skapa först ett språk, beskriv det språket, och skriv sedan en parser för det.

## Uppgift 5: Mellankod (6 p)

Det här är ett programavsnitt i ett C-liknande språk:

```
a = b;  
c = 17;  
while (x < 19) {  
    x = x + b - c - 1;  
    if (c > 0)  
        c = c - 1;  
    r = r + 2 * b;  
}
```

Översätt ovanstående programavsnitt till *två* av följande tre typer av mellankod.

- a) ett abstrakt syntaxträd (genom att rita upp trädet!)
- b) postfixkod för en stackmaskin
- c) treadsckod

**Observera:** Det finns tre deluppgifter i uppgiften ovan. Välj ut och besvara (högst) *två* av dessa. (Skulle du svara på alla tre, räknas den med högst poäng bort.)

## Uppgift 6: Optimering (5 p)

- a) Vad är ett *basic block*?
  - b) Flera olika typer av optimeringar kan göras på koden inuti ett enskilt basic block. Vilka? Beskriv dem, och använd treadsckod för att ge exempel på hur de fungerar.
-