

Örebro universitet  
Akademin för naturvetenskap och 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

torsdag 20 augusti 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

---

<b>Hjälpmedel:</b>	Inga hjälpmedel.
<b>Poängkrav:</b>	Maximal poäng är 30. För godkänt betyg (3 respektive G) krävs 15 poäng.
<b>Resultat och lösningar:</b>	Meddelas via e-post eller på campuskursens hemsida, <a href="http://www.aass.oru.se/~tpy/c/2008-2009-p2/">http://www.aass.oru.se/~tpy/c/2008-2009-p2/</a> , senast torsdag 10 september 2009.
<b>Återlämning av tentor:</b>	Efter att resultatet meddelats kan tentorna hämtas på institutionen. Man kan också få sin rättade tenta hemskickad.
<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 , (typ)	höger
	Multiplikation mm	* , / , %	vänster
	Addition mm	+ , -	vänster
	Jämförelser	< , <= , >= , >	vänster
	Likhetsjämförelser	== , !=	vänster
	Logiskt OCH	&&	vänster
	Logiskt ELLER		vänster
Lägsta	Tilldelning	= , += , -= , *= , /= , %=	höger

## Scenario till uppgifterna

Det här är en tvättmaskin:



Nu är just den tvättmaskinen förmodligen inte datorstyrd, men mer moderna tvättmaskiner innehåller ofta en liten inbyggd dator som styr motor, värme och ventiler, beroende på vilket tvättprogram som användaren ställt in.

Den datorn har för det mesta begränsat med processorkraft och minne, och programmeras ofta i C.

Ett **tvättprogram** (som inte ska förväxlas med ett datorprogram!) består av ett antal **programsteg**, som utförs efter varandra. Det finns inga val eller loopar, utan programstegen i ett tvättprogram utförs alltid i samma sekvens. Ett programsteg kan vara till exempel vattenpåfyllning, sköljning eller centrifugering.

## Uppgift 1 (3 p)

Ett **programsteg** har ett namn på högst trettio tecken, till exempel **Centrifugering**. Samma namn (till exempel **Sköljning**) kan förekomma flera gånger i samma tvättprogram. För programsteget anger man dessutom följande uppgifter:

- **Tid**, som är hur många sekunder programsteget ska vara. Centrifugeringen kanske ska pågå i två minuter, och då anges tiden som **120**.
- **Temperatur**, som är ett heltal som anger vilken temperatur (mätt i vanliga Celsius-grader) som vattnet ska värmas till. Det finns en termostat som stänger av värmen när vattnet nått den temperaturen. Värdet noll betyder att värmen är avstängd hela tiden. (I alla fall om man inte fyllt maskinen med isbitar som är kallare än noll grader!)
- **Trumfart**, som anger med vilken fart trumman ska snurra under programsteget, mätt i antal varv per sekund. Det är ett flyttal.
- **Inpumpning**, som anger om maskinen ska släppa in vatten eller inte. Det finns ingen nivågivare eller mätare för vattenmängden, utan mängden vatten som ska släppas in styrs med hjälp av tiden för programsteget.
- **Utpumpning**, som anger om tömningspumpen ska vara igång eller inte. Det finns ingen givare som talar om när maskinen är tom, utan mängden vatten som ska pumpas ut styrs med hjälp av tiden för programsteget.

Skapa posttypen **struct Programsteg**, som används för att lagra data om ett programsteg.

## Uppgift 2 (3 p)

Skriv funktionen **visa\_ett\_programsteg**, som skriver ut data om ett programsteg på skärmen. (Nu har tvättmaskinen nog ingen vanlig skärm, men det struntar vi i nu.)

Utskriften behöver inte se ut exakt så här, men samma informationen ska finnas med:

Centrifugering: 30 sekunder, 0 grader, 10.00 varv/s, in: AV, ut: PÅ

## Uppgift 3 (3 p)

Ett **tvättprogram** har ett namn på högst trettio tecken, till exempel **Kulörtvätt**. Tvättprogrammet består av ett antal **programsteg** enligt uppgiften ovan, högst 100 stycken.

Skapa posttypen **struct Program**, som används för att lagra data om ett tvättprogram (namnet och alla programstegen).

## Uppgift 4 (3 p)

Skriv funktionen **visa\_ett\_program**, som skriver ut ett helt tvättprogram på skärmen. Funktionen ska anropa funktionen **visa\_ett\_programsteg**.

Utskriften behöver inte se ut exakt så här, men samma informationen ska finnas med:

```
Tvättprogrammets namn: Snabbtvätt
13 programsteg:
1. Vattenpåfyllning: 30 sekunder, 0 grader, 0.00 varv/s, in: PÅ, ut: AV
2. Tvättning: 300 sekunder, 40 grader, 1.50 varv/s, in: AV, ut: AV
3. Vattentömning: 30 sekunder, 0 grader, 0.00 varv/s, in: AV, ut: PÅ
4. Centrifugering: 30 sekunder, 0 grader, 10.00 varv/s, in: AV, ut: PÅ
5. Vattenpåfyllning: 30 sekunder, 0 grader, 0.00 varv/s, in: PÅ, ut: AV
6. Sköljning: 300 sekunder, 0 grader, 1.00 varv/s, in: AV, ut: AV
7. Vattentömning: 30 sekunder, 0 grader, 0.00 varv/s, in: AV, ut: PÅ
8. Centrifugering: 30 sekunder, 0 grader, 10.00 varv/s, in: AV, ut: PÅ
9. Vattenpåfyllning: 30 sekunder, 0 grader, 0.00 varv/s, in: PÅ, ut: AV
10. Sköljning: 300 sekunder, 0 grader, 1.00 varv/s, in: AV, ut: AV
11. Vattentömning: 30 sekunder, 0 grader, 0.00 varv/s, in: AV, ut: PÅ
12. Centrifugering: 30 sekunder, 0 grader, 10.00 varv/s, in: AV, ut: PÅ
13. Slutcentrifugering: 60 sekunder, 0 grader, 20.00 varv/s, in: AV, ut: PÅ
```

## Uppgift 5 (6 p)

Tvättprogrammen lagras på filer. Här är till exempel innehållet i filen **snabbtvätt.txt**, som innehåller tvättprogrammet **Snabbtvätt**, ett tvättprogram vi känner igen från uppgiften ovan:

```
Snabbtvätt
Vattenpåfyllning 30 0 0.0 1 0
Tvättning 300 40 1.5 0 0
Vattentömning 30 0 0.0 0 1
Centrifugering 30 0 10.0 0 1
Vattenpåfyllning 30 0 0.0 1 0
Sköljning 300 0 1.0 0 0
Vattentömning 30 0 0.0 0 1
Centrifugering 30 0 10.0 0 1
Vattenpåfyllning 30 0 0.0 1 0
Sköljning 300 0 1.0 0 0
Vattentömning 30 0 0.0 0 1
Centrifugering 30 0 10.0 0 1
Slutcentrifugering 60 0 20.0 0 1
```

(Egentligen har tvättmaskinen nog ingen hårddisk med filer, men det struntar vi i nu.)

Skriv en funktion som heter **las\_ett\_program**, som tar ett filnamn som argument, öppnar filen och läser in det tvättprogram som finns på filen, och på lämpligt sätt returnerar tvättprogrammet till den anropande funktionen.

Det är tillåtet att göra en hjälpfunktion som heter **las\_ett\_programsteg**, men det är inget krav.

## Uppgift 6 (2 p)

a)

För att lagra de inlästa tvättprogrammen behöver vi en array av tvättprogram. Skapa därför variabeln **alla\_program**, som är en array av 100 stycken **struct Program**. Skapa också en heltalsvariabel som heter **antal\_program**, som får startvärdet noll, och som anger antalet poster som vi lagrat i den arrayen.

b)

Men behöver vi verkligen variabeln **antal\_program**? Vi har ju redan anggett en storlek på arrayen? Förklara!

## Uppgift 7 (5 p)

En fil med namnet **tvättprogram.txt** innehåller filnamnen på alla filer med tvättprogram som tvättmaskinen har:

```
snabbtvätt.txt
ylletvätt.txt
kulörtvätt.txt
fintvätt.txt
vittvätt.txt
```

Skriv en funktion som heter **las\_alla\_program**, som öppnar den filen, läser in de filnamn som finns på filen, och sen använder funktionen **las\_ett\_program** för att läsa in alla de olika tvättprogrammen.

Tvättprogrammen ska hamna i variabeln **alla\_program**.

## Uppgift 8 (2 p)

Skriv en funktion som heter **visa\_alla\_program**, som helt enkelt skriver ut alla tvättprogrammen i **alla\_program**. Den ska anropa funktionen **visa\_ett\_program**

## Uppgift 9 (2 p)

För att provköra alltihop behöver vi en **main**-funktion som först anropar **las\_alla\_program** och sen **visa\_alla\_program**. Skriv den!

## Uppgift 10 (1 p)

Vi samlar ihop hela tentasvaret till ett komplett C-program. Vilka **#include**-rader behövs i början av programmet?

---