

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

## Tentamen i distanskursen

# Programmering C

torsdag 17 mars 2016

Gäller som tentamen för:  
DT104G Programmering C, provkod 0100  
DT1006 Datateknik A, Programmering C, distans, provkod 0100

*Obs! Campuskursen "Programmering grundkurs" har en egen tenta.*

---

<b>Hjälpmedel:</b>	Ordbok för översättning.
<b>Poängkrav:</b>	Maximal poäng är 40. För godkänt betyg (3 respektive G) krävs 20 poäng.
<b>Resultat och lösningar:</b>	Meddelas via e-post senast torsdag 7 april 2016.
<b>Återlämning av tentor:</b>	Efter att resultatet meddelats kan tentorna hämtas elektroniskt via Studentforum.
<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

# Några användbara biblioteksfunktioner

## stdlib.h

```
int rand(void);
void srand(unsigned int seed);
void *malloc(size_t size);
void *realloc(void *ptr, size_t size);
void free(void *ptr);
void exit(int status);
void qsort(void *base, size_t nmem, size_t size,
           int(*compar)(const void *, const void *));
```

## stdio.h

```
FILE *fopen(const char *path, const char *mode);
int fclose(FILE *stream);
int getc(FILE *stream);
int getchar(void);
int ungetc(int c, FILE *stream);
char *fgets(char *s, int size, FILE *stream);
char *gets(char *s);
int putc(int c, FILE *stream);
int printf(const char *format, ...);
int fprintf(FILE *stream, const char *format, ...);
int sprintf(char *str, const char *format, ...);
int snprintf(char *str, size_t size, const char *format, ...);
int scanf(const char *format, ...);
int fscanf(FILE *stream, const char *format, ...);
int sscanf(const char *str, const char *format, ...);
size_t fread(void *ptr, size_t size, size_t nmem, FILE *stream);
size_t fwrite(const void *ptr, size_t size, size_t nmem, FILE *stream);
```

## string.h

```
size_t strlen(const char *s);
char *strcpy(char *dest, const char *src);
char *strncpy(char *dest, const char *src, size_t n);
int strcmp(const char *s1, const char *s2);
int strncmp(const char *s1, const char *s2, size_t n);
char *strcat(char *dest, const char *src);
char *strncat(char *dest, const char *src, size_t n);
char *strstr(const char *haystack, const char *needle);
void *memmove(void *dest, const void *src, size_t n);
```

## ctype.h

```
int isalnum(int c);
int isalpha(int c);
int isblank(int c);
int isdigit(int c);
int islower(int c);
int isprint(int c);
int ispunct(int c);
int isspace(int c);
int isupper(int c);
```

## math.h

```
double sqrt(double x);
double pow(double x, double y);
```

## Uppgift 1 (1 p)

Vilka värden har följande C-uttryck?

a)  $1 / 2 - 3 + 4$

b)  $1 * 2 * 3 / 4$

c)  $1 \% 2 \% 3 \% 4$

d)  $1 / 2 + 3 / 4$

## Uppgift 2 (2 p)

Vad skrivs ut när vi kör följande C-program?

```
#include <stdio.h>

int main(void) {
    int a, b, c, d;

    a = 0;
    for (b = 0; b < 1000; ++b) {
        for (c = 0; c < 1000; ++c) {
            a = a + 1;
            if (a == 2000)
                printf("Hej!\n");
        }
    }
    printf("a = %d\n", a);
    printf("d = %d\n", d);

    return 0;
}
```

En del variabler kan ha ett odefinierat värde ("skräp") när de ska skrivas ut. Ange då detta!

## Uppgift 3 (4 p)

Vi vill kunna beräkna följande uttryck:

$$\frac{1}{\sqrt{x-1}}(\sqrt{3-x^2}-\sqrt{x-1})^3$$

Skriv därför ett komplett C-program (med **#include** och allt) som skriver ut x och värdet av uttrycket för alla x mellan 0 och 10 i steg om 0.1. Om nämnaren i divisionen är lika med noll, eller om ett deluttryck under ett rottecken är mindre än noll, går uttrycket inte att beräkna. I så fall ska programmet inte försöka beräkna uttrycket, utan det ska hoppa över den utskriften.

I den här och alla andra uppgifter på tentan gäller:  
 Man kan strunta i detaljer som bara behövs just när man utvecklar konsolprogram i Visual Studio, som konstiga teckenkoder för ÅÄÖ, eller att fönstret med programkörningen försvinner när programmet avslutas.

## Uppgift 4 (3 p)

Skriv ett C-program som läser in en textrad (på högst 100 tecken) och ett heltal, och sen skriver ut den textraden så många gånger som angavs av heltalet.

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 5 (1 p)

På himlen finns många stjärnor, till exempel Polstjärnan och Betelgeuse. Vi behöver en posttyp (med C-terminologi en **struct**) som kan användas för att lagra data om stjärnor. Det som ska finnas med i posten är:

- Stjärnans namn, till exempel **Polstjärnan**. Namnet kan vara högst 30 tecken långt, och består av ett eller flera ord.
- Hur starkt stjärnan lyser, sett från jorden. Det kallas för **skenbar magnitud**, och anges med ett reellt tal som kan vara positivt eller negativt. (Lägre tal betyder starkare ljus.)
- Hur starkt stjärnan lyser, i sig. Det kallas för **absolut magnitud**, och anges med ett reellt tal på samma sätt som den skenbara magnituden.

Definiera posttypen. Den ska heta **struct Stjarna**.

## Uppgift 6 (1 p)

Definiera en variabel av typen **struct Stjarna** och initiera den med data om stjärnan **Sirius**, som har den skenbara magnituden **-1.46** och den absoluta magnituden **1.42**.

## Uppgift 7 (2 p)

Skriv en funktion som heter **visa\_stjarna**, som skriver ut en stjärna (som i uppgift 5) på skärmen. Funktionen tar stjärnposten som parameter, och skriver ut den. Exempel på hur utskriften kan se ut:

```
Namn: Canopus
Skenbar magnitud: -0.72
Absolut magnitud: -5.65
```

## Uppgift 8 (3 p)

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

## Uppgift 9 (1 p)

Skriv en funktion som heter **ljusast**, som tar två stjärnor som argument, och som returnerar den ljusaste stjärnan, dvs den som har den *lägsta* skenbara magnituden.

## Uppgift 10 (2 p)

Skriv en main-funktion som har två lokala variabler av typen **struct Stjarna**, som läser in data om två stjärnor till dessa variabler med hjälp av funktionen **las\_stjarna**, använder funktionen **ljusast** för att avgöra vilken av dem som lyser starkast, och sen skriver ut den ljusaste stjärnans data med hjälp av funktionen **visa\_stjarna**.

I den här och alla andra uppgifter på tentan gäller: Om du ska anropa en funktion från en tidigare uppgift, behöver du inte skriva koden för den funktionen på nytt. Du får också anropa funktionen även om du inte gjort uppgiften där man skulle skriva den.

## Uppgift 11 (5 p)

Skriv ett program som först läser in ett månadsnummer, (mellan **1** och **12**) och därefter läser in personnummer på formen **ÅÅMMDD-SSSS**, med ett personnummer per rad, upprepat tills användaren ger en tom rad.

Därefter ska programmet ange hur många personer som var födda den angivna månaden.

Exempel på hur en programkörning skulle kunna se ut (med användarens inmatning understruken):

```
Vilken månad (1-12): 12
Ange personnummer (ett per rad, avsluta med tom rad):
631211-1658
451019-2270
811214-3360
821001-6720
851001-2273
```

```
Antal personer födda i månad 12: 2
```

## Uppgift 12 (5 p)

Vi vill skapa en fil med heltal. Välj själv om det ska vara en textfil eller en binärfil. Skriv ett program som frågar efter filens namn, antalet tal, och det minsta och största tillåtna talet. Därefter ska programmet skapa filen och fylla den med framslumpade tal, i det angivna intervallet.

Om filen inte går att öppna, ska ett felmeddelande skrivas ut, och programmet ska avslutas.

*Ledtråd:* Använd anropet **rand()** för att få ett stort, slumpmässigt heltal. Innan man kan använda den funktionen, måste slumpalsgeneratorn initieras, till exempel med anropet **srand(time(NULL));**

## Uppgift 13 (5 p)

Vi använder programmet från uppgiften ovan för att skapa en fil med flera miljarder tal mellan ett och en miljon. Det är mer data än vad som får plats i datorns primärminne.

Skriv ett C-program som läser filen och talar om dels hur många tal som totalt finns på filen, och dels hur många av talen som avviker med mer än 10 procent från talens medelvärde.

Om filen inte går att öppna, ska ett felmeddelande skrivas ut, och programmet ska avslutas.

*Ledtråd:* Man måste läsa igenom filen två gånger, först en gång för att räkna ut medelvärdet, och sen en gång för att räkna antalet avvikande tal.

## Uppgift 14 (5 p)

Filen i uppgiften ovan innehåller alltså flera miljarder heltal, vilket är mer data än vad som får plats i datorns primärminne. Men eftersom talen slumpades fram av programmet från uppgift 12, vet vi att det är heltal mellan ett och en miljon, och alltså finns det högst en miljon *olika* tal. En miljon tal får lätt plats i minnet.

Skriv ett C-program som läser filen och skriver ut de unika talen (dvs med alla dubletter borttagna) på en annan fil.

(Vi har inga prestandakrav. Programmet får gärna ta lång tid att köra, bara det ger rätt utdata.)

Om någon av filerna inte går att öppna, ska ett felmeddelande skrivas ut, och programmet ska avslutas.

*Ledtråd:* Man kan lagra de unika talen i minnet, till exempel i en array.

---