

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

fredag 13 januari 2017

Gäller som tentamen för:
DT104G Programmering C, provkod 0100
DT1006 Datateknik A, Programmering C, distans, provkod 0100
(Campuskursen "Programmering grundkurs" har en egen tenta.)

Hjälpmedel:	Ordbok för översättning.
Poängkrav:	Maximal poäng är 35. För godkänt betyg (3 respektive G) krävs 18 poäng.
Resultat och lösningar:	Meddelas via e-post senast fredag 3 februari 2017.
Återlämning av tentor:	Efter att resultatet meddelats kan tentorna hämtas elektroniskt via Studentforum.
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

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 (3 p)

Skriv en C-funktion som heter **add_matrix**, som tar en 5x5-matris med heltal som parameter, och som returnerar summan av alla talen i matrisen.

Exempel: Om variabeln **m** definierats och initierats enligt nedan, ska anropet **add_matrix(m)** ge värdet 50.

```
int m[5][5] = { { 1, 2, 3, 4, 0 },
                { 0, 4, 3, 2, 1 },
                { 2, 3, 4, 0, 1 },
                { 3, 4, 0, 1, 2 },
                { 1, 3, 0, 4, 2 } };
```

Uppgift 3 (5 p)

Skriv ett komplett C-program (med **#include** och **allt**) som ritar upp ett stort X på skärmen. X:et ska skrivas med tecknet '*' och mellanslag. Programmet ska fråga efter hur stort X:et ska vara.

Här är tre olika exempel som visar hur dialogen kan se ut:

```
Hur stort ska X:et vara? 3
```

```
* *
 *
* *
```

```
Hur stort ska X:et vara? 4
```

```
* *
 **
 **
* *
```

```
Hur stort ska X:et vara? 10
```

```
*      *
 *      *
 *      *
  *     *
   **
   **
  *     *
 *      *
 *      *
 *      *
```

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 4 (8 p)

a) (6p)

Skriv en C-funktion "count_words", som tar en sträng som argument, och som returnerar antalet ord i strängen (ett heltal). Med ett "ord" menar man en följd av tecken, som omges av antingen mellanslag eller en av strängens båda ändar.

Några exempel:

```
count_words("abc") = 1
count_words("Hej-Alla-Barn!") = 1
count_words("abc ") = 1
count_words(" abc ") = 1
count_words("abc def") = 2
count_words("abc def") = 2
count_words("") = 0
```

Godtyckligt långa strängar ska kunna hanteras.

b) (2p)

Skriv en **main**-funktion som upprepar läser in en rad från användaren, tills användaren matar in en tom rad, och för varje inläst rad skriver ut hur många ord raden innehåller. Vi ska använda funktionen **count_words**. Vi kan anta att ingen inmatad rad är längre än 1000 tecken (plus radslutstecknet).

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 5 (11 p)

Heltal ska lagras i en länkad lista.

a) (1p) Definiera de datatyper och variabler som behövs.

b) (3p) Skriv funktionen **add_number_first**, som tar ett heltal som argument och på lämpligt sätt stoppar in det först i den länkade listan.

c) (2p) Skriv funktionen **sum_list**, som returnerar summan av alla talen i listan.

d) (2p) Skriv funktionen **in_list**, som tar ett heltal som argument, och som returnerar **1** om talet finns i listan. Om talet inte finns i listan ska **0** returneras.

e) (3p) Skriv en **main**-funktion som läser in heltal från standardinmatningen ända tills talet noll lästs in, som mellanlagrar talen i en länkad lista med hjälp av funktionerna ovan, och till sist skriver ut talens summa.

Uppgift 6 (7 p)

Vi ska skriva ett par funktioner som arbetar med textfiler.

a) (3p)

Skriv en C-funktion som heter **print_file_content** och som skriver ut innehållet i en textfil. Den ska ta en sträng med ett filnamn som parameter, och skriva ut den filens innehåll på standardutmatningen (stdout). (Funktionen måste förstås öppna filen och läsa från den.) Funktionen ska på lämpligt sätt returnera en kod som anger ifall det gick bra eller inte. Det räcker med att kolla ifall filen gick att öppna.

b) (4p)

Antag att vi har en samling textfiler med hjälpinformation om olika ämnen. Varje fil har ett namn som består av det ämne den handlar om, plus ändelsen ".help". Till exempel finns filen "utskrifter.help", som handlar om utskrifter.

Skriv nu en C-funktion som heter **print_help** och som skriver ut hjälpinformation om ett visst ämne genom att anropa funktionen **print_file_content** ovan. Funktionen ska ta en sträng som parameter, och den strängen innehåller det ämne som man vill ha information om. Funktionen ska inte returnera något värde.

Till exempel ska anropet **print_help("utskrifter")** göra att innehållet i filen "utskrifter.help" skrivs ut på standardutmatningen.

Om det inte fanns någon information om det önskade ämnet, dvs om det inte fanns någon fil med det namnet, ska en lämplig text (till exempel "Tyvärr finns ingen hjälpinformation om det ämnet.") skrivas ut.
