

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

lördag 2 juni 2018

Gäller som tentamen för:
DT104G Programmering C, provkod 0100

Hjälpmedel:	Ordbok för översättning.
Poängkrav:	Maximal poäng är 40. För godkänt betyg (3 respektive G) krävs 20 poäng.
Resultat och lösningar:	Meddelas via e-post senast lördag 23 juni 2018.
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. Gjorda antaganden får inte förändra den givna uppgiften.
 - 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 *str);
size_t fwrite(const void *ptr, size_t size, size_t nmem, FI
```

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);
```

För uppgifterna på tentan gäller: Om du ska använda något från en tidigare uppgift eller deluppgift, till exempel anropa en funktion från den tidigare uppgiften, behöver du inte skriva koden på nytt. Du får också göra uppgiften även om du inte gjort den tidigare uppgiften som den bygger på.

För uppgifterna 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 1 (1 p)

Vilka värden har följande C-uttryck?

a) $(1 + 2) * 3$

b) $1 + 2 * 3$

c) $1+2 * 3$

d) $3 / 2 * 4$

Uppgift 2 (3 p)

Vad skrivs ut när följande C-program körs?

```
#include <stdio.h>

void pluset(int x) {
    x = x + 1;
    printf("x = %d\n", x);
}

void plustva(int x) {
    pluset(x);
    pluset(x);
}

void plustre(int y) {
    plustva(y);
    pluset(y);
}

int main(void) {
    int a = 2;
    int b = 3;
    printf("a = %d, b = %d\n", a, b);
    plustre(a);
    printf("a = %d, b = %d\n", a, b);
    return 0;
}
```

Uppgift 3 (5 p)

Hur många stjärnor skrivs ut av var och en av följande loopar?

a)

```
int i;
i = 1;
while (i < 10) {
    printf("*");
    ++i;
}
```

b)

```
int i;
i = 0;
while (i <= 10) {
    printf("*");
    i = i + 1;
}
```

c)

```
for (int i = 0; i < 10; ++i)
    printf("*");
```

d)

```
for (int i = 0; i < 2; ++i) {
    for (int j = 5; j < 10; ++j)
        printf("*");
}
```

e)

```
int i;
i = 10;
while (i >= 0) {
    printf("*");
    i = i - 1;
}
```

Uppgift 4 (6 p)

a) (3p)

Skriv en funktion **max2** som hittar det *näst* högsta talet i en array med flyttal. Funktionen ska ta två argument: en array av flyttal, och ett heltal som anger hur många tal arrayen innehåller. Funktionen ska returnera det sökta talet. Vi kan anta att arrayen alltid innehåller minst två tal, och att det finns ett entydigt största och ett entydigt näst största tal.

b) (3p)

Skriv en **main**-funktion som låter användaren skriva in 57 stycken flyttal, placerar dem i en array, och anropar funktionen **max2** från uppgiften ovan för att ta reda på det näst högsta av de inlästa flyttalen. Till slut ska det talet skrivas ut.

Uppgift 5 (5 p)

På fakturor brukar man ofta göra en öresutjämning, dvs. en avrundning till närmaste hela kronor genom att man lägger till eller drar ifrån ett lämpligt belopp. Skriv ett fullständigt C-program som tillverkar en faktura enligt nedan, där fakturerat belopp läses in från tangentbordet och resterande beräkningar utförs och skrivs ut av programmet.

Fakturerat belopp:	1224.56 kr
Moms 25%:	306.14 kr
Totalbelopp:	1530.70 kr
Öresutjämning:	0.30 kr
Belopp att betala:	1531 kr

Uppgift 6 (5 p)

Skriv ett fullständigt C-program som läser in en textrad (högst 100 tecken, radslutstecken ej inräknat) och sedan skriver ut alla initialbokstäver och skiljetecken. En initialbokstav är den första bokstaven i en sekvens av en eller flera bokstäver, där sekvenserna skiljs åt av blanktecken eller skiljetecken. Alla tecken som inte är bokstäver eller blanktecken räknas som skiljetecken.

Exempel på inmatning:

Hej! Vad heter du? Jag heter, tror jag, K. Orvar Juliusson!!!

Utmatning:

H!Vhd?Jh,tj,K.OJ!!!

Uppgift 7 (7 p)

Skriv ett fullständigt C-program som bokar numrerade biljetter till dagens föreställning i en biosalong, bestående av 120 platser, på Filmstaden.

Programmet ska börja med att upprepat rita upp biosalongen på skärmen så att det framgår vilka av salongens platser som är lediga. Programmet läser sedan in önskat platsnummer. Här nedan visas ett utdrag av hur det ser ut på skärmen när platserna 84 och 96 redan bokats tidigare, och plats 76 nu bokas. Programmet avslutas när man matar in platsnummer 0.

Lediga platser:

```

  1    2    3    4    5    6    7    8    9   10
11   12   13   14   15   16   17   18   19   20
21   22   23   24   25   26   27   28   29   30
31   32   33   34   35   36   37   38   39   40
41   42   43   44   45   46   47   48   49   50
51   52   53   54   55   56   57   58   59   60
61   62   63   64   65   66   67   68   69   70
71   72   73   74   75   76   77   78   79   80
81   82   83           85   86   87   88   89   90
91   92   93   94   95           97   98   99  100
101  102  103  104  105  106  107  108  109  110
111  112  113  114  115  116  117  118  119  120

```

Ange platsnummer (avsluta med 0): 76

Lediga platser:

```

  1    2    3    4    5    6    7    8    9   10
11   12   13   14   15   16   17   18   19   20
21   22   23   24   25   26   27   28   29   30
31   32   33   34   35   36   37   38   39   40
41   42   43   44   45   46   47   48   49   50
51   52   53   54   55   56   57   58   59   60
61   62   63   64   65   66   67   68   69   70
71   72   73   74   75           77   78   79   80
81   82   83           85   86   87   88   89   90
91   92   93   94   95           97   98   99  100
101  102  103  104  105  106  107  108  109  110
111  112  113  114  115  116  117  118  119  120

```

Ange platsnummer (avsluta med 0):

Uppgift 8 (4 p)

Skriv ett program som skapar en fil med en miljard framslumpade tal.

Uppgift 9 (4 p)

Skriv ett program som läser filen från programmet ovan, och skriver ut medelvärdet av talen.
