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

Tentamen i

Programmering grundkurs och Programmering C

för D1 m fl, även distanskursen

onsdag 5 juni 2013

Gäller som tentamen för:
DT1029 Datateknik A, Programmering grundkurs, provkod 0100
DT1030 Datateknik A, Tillämpad datavetenskap, provkod 0410
DT1006 Datateknik A, Programmering C, distans, provkod 0100
DT1016 Datateknik A, Programmering grundkurs, provkod 0100
DT1007 Datateknik A, Tillämpad datavetenskap, provkod 0410

Hjälpmedel:	Inga hjälpmedel.
Poängkrav:	Maximal poäng är 40. För godkänt betyg (3 respektive G) krävs 20 poäng.
Resultat och lösningar:	Meddelas via e-post eller på kursens hemsida, http://basen.oru.se/kurser/c/2012-2013-p2/, senast onsdag 26 juni 2013.
Återlämning av tentor:	Efter att resultatet meddelats kan tentorna hämtas på universitetets centrala tentamensutlämning.
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 nmemb, 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 nmemb, FILE *stream);
size_t fwrite(const void *ptr, size_t size, size_t nmemb, 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);
```

Uppgift 1 (1 p)

Vilka värden har följande uttryck?

a) 5 * 2 + 3

b) 5 + 2 * 3

c) 5 / 2 % 3

Uppgift 2 (1 p)

Skriv formeln nedan som en tilldelningssats i C.

$$x = \frac{5yz}{5} + \sqrt{2y^2} - 5$$

Uppgift 3 (3 p)

Vad skrivs ut när följande C-program körs?

```
#include <stdio.h>

void f(int varv) {
    int i;
    for (i = 0; i < varv; i++)
        printf("*");
}

int main(void) {
    int i;
    int x;
    int y = 2;
    x = y + 1;
    for (i = 0; i < 3; ++i) {
        f(i);
        printf("x = %d, x + %d = %d\n", x, i, x + i);
    }
    return 0;
}
```

Uppgift 4 (4 p)

Skriv ett komplett C-program (med **#include** och allt) som läser in reella tal, ända tills användaren matar in talet noll. Därefter ska programmet skriva ut summan av de inmatade talen, samt det största och det minsta talet.

> 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.

> 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, och att fönstret med programkörningen försvinner när programmet avslutas.

Uppgift 5 (2 p)

Vi vill hålla reda på djuren i en djurpark. Varje djur har ett unikt namn, till exempel **Richard Parker**, och det tillhör en art, till exempel **bengalisk tiger**. Namn och arter kan vara högst 20 tecken långa. Dessutom har vi vägt en del av djuren och känner till deras vikt, till exempel **221,2** kilogram. Om vi inte känner till ett visst djurs vikt, anges den med ett negativt tal.

a) (1p) Definiera datatypen **struct Djur**, som beskriver ett djur.

b) (1p) Definiera en variabel av typen **struct Djur** och initiera den med data om den bengaliska tigern Richard Parker, som väger 221,2 kilo.

> I den här och alla andra uppgifter på tentan gäller:
> Om du behöver använda något från en tidigare uppgift eller deluppgift, till exempel utnyttja en datatyp eller anropa en funktion som skrevs i den tidigare uppgiften, så behöver du inte skriva samma kod igen. Du får också göra uppgiften även om du inte gjort den tidigare uppgiften.

Uppgift 6 (3 p)

Skriv en funktion som heter **total_vikt**, som tar en array med djurposter som argument, tillsammans med ett heltalsargument som anger antalet djur i arrayen, och returnerar djurens sammanlagda vikt. Negativa vikter, som betyder att vi inte vet vikten på djuret, ska inte vara med i summan.

Uppgift 7 (3 p)

Vi vill provköra funktionen **total_vikt** från uppgiften ovan. Skriv därför en **main**-funktion, som anropar den med lämpliga argument, och skriver ut ett tydligt felmeddelande om den inte ger det förväntade resultatet.

Vi ska bara anropa funktionen en gång, så det finns alltså bara ett enda testfall. Försök göra det testfallet så bra som möjligt!

Uppgift 8 (2 p)

Skriv en funktion som heter **visa_djur**, och som visar ett djurs data på skärmen. Den ska alltså skriva ut djurets uppgifter, med lämpliga ledtexter, på standardutmatningen. Du får själv välja om du vill att funktionshuvudet ska se ut så här:

```
void visa_djur(struct Djur djuret)
```

eller så här:

```
void visa_djur(struct Djur *djuret)
```

Uppgift 9 (3 p)

Skriv en funktion som heter **las_djur**, och som läser in en djurs data (enligt uppgift 4 ovan). Funktionen ska skriva ut lämpliga ledtexter på standardutmatningen, och läsa in data från standardinmatningen (som normalt är kopplad till tangentbordet). Du får själv välja om du vill att funktionshuvudet ska se ut så här:

```
struct Djur las_djur()
```

eller så här:

```
void las_djur(struct Djur* p)
```

Uppgift 10 (3 p)

Skriv en **main**-funktion som har tre lokala variabler av djurtypen från uppgifterna ovan, och som läser in data om tre djur till dessa variabler med hjälp av funktionen **las_djur**. Därefter ska den kopiera de tre djurens data till en array, använda funktionen **total_vikt** för att räkna ut djurens sammanlagda vikt, och skriva ut den vikten.

Uppgift 11 (4 p)

Skriv ett C-program som läser in data om djur från användaren, med hjälp av funktionen **las_djur**, och sparar dem på en fil. Välj själv om det ska vara en text- eller binärfil. Tala om vilket du valde!

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

Välj själv hur användaren ska avsluta inmatningen, dvs hur man anger att man inte vill mata in fler djur, och beskriv vilket sätt du valde.

Glöm inte att beskriva hur inmatningen avslutas, och om du valde en textfil eller en binärfil!

Uppgift 12 (5 p)

Skriv ett C-program som läser filen från uppgiften ovan, och skriver ut vilken art som det tyngsta djuret tillhör.

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

Uppgift 13 (6 p)

Skriv ett komplett C-program (med **#include** och allt) som ritar en trappa. Programmet ska fråga efter antalet trappsteg som trappan består av, höjden på ett trappsteg, och bredden. Därefter ska det rita upp trappan med hjälp av stjärnor.

Ett körexempel, med användarens inmatning understruken:

```
Hur många trappsteg? 4
Hur högt är ett trappsteg? 3
Hur brett är ett trappsteg? 10

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