

Tentamen - Programmering

DVA117

Akademien för innovation, design och teknik

Tisdagen 2013-11-05

Skrivtid: 14.10 – 17.30
Hjälpmedel: Inga
Lärare: Caroline Uppsäll, 021-101456
Robert Suurna, 021-151790

Preliminära betygsgränser

Betyg 3: 25p
Betyg 4: 34p
Betyg 5: 42p
Max: 45p

Bonuspoäng

Endast bonuspoäng som intjänats under kursinstansen HT13, p1, kan användas på den här tentamen. Bonuspoängen uppgår till max 6 poäng.

Allmänt

- All kod skall skrivas i standard ANSI C.
- Påbörja varje ny uppgift på nytt blad och skriv bara på ena sidan av pappret.
Deluppgifter (a, b, c,...) kan skrivas på samma blad
- Referera inte mellan olika svar.
- Om du är osäker på vad som avses i någon fråga, skriv då vad du gör för antagande.
- *Oläsliga/oförståeliga/ostrukturerade svar rättas inte.*
- Kommentera din kod!
- Tips: Läs igenom hela tentan innan du börjar skriva för att veta hur du ska disponera din tid.

Lycka till!

Prioritet och associativitet hos operatorerna i C

Högsta prioritet högst upp i tabellen, lägsta prioritet längst ner i tabellen. Grupperade operatorer har samma prioritet.

Table A.5 Summary of C Operators (Programming in C, Stephen G. Kochan)

Operator	Description	Associativity
()	Function call	Left to right
[]	Array element reference	
->	Pointer to structure member reference	
.	Structure member reference	
-	Unary minus	Right to left
+	Unary plus	
++	Increment	
--	Decrement	
!	Logical negation	
~	Ones complement	
*	Pointer reference (indirection)	
&	Address	
sizeof	Size of an object	
(type)	Type cast (conversion)	
*	Multiplication	Left to right
/	Division	
%	Modulus	
+	Addition	Left to right
-	Subtraction	
<<	Left shift	Left to right
>>	Right shift	
<	Less than	Left to right
<=	Less than or equal to	
>	Greater than	
>=	Greater than or equal to	
==	Equality	Left to right
!=	Inequality	
&	Bitwise AND	Left to right
^	Bitwise XOR	Left to right
	Bitwise OR	Left to right
&&	Logical AND	Left to right
	Logical OR	Left to right
?:	Conditional	Right to left
=	Assignment operators	Right to left
*= /= %=		
+= -= &=		
^= =		
<<= >>=		
,	Comma operator	Right to left

Uppgift 1 [7p]

- Vad innebär det att ett program kompileras?
- Beskriv i ord och bild vad fält (en array) är och varför man använder fält i sina program?
- Beskriv skillnaden mellan stacken och heapen. Vad lagras på respektive ställe?
- Beskriv en fördel med dynamiskt allokerat minne.
- Vad har funktionen `bind()` för uppgift när vi programmerar sockets?

Uppgift 2[4p]

Vilka värden har följande uttryck i C (vad evakuerar varje uttryck till)?

För samtliga uttryck gäller: `int x=4, y=26, z=45, a=1, b=9;`

- `z > 8 && x != y / 2`
- `!(a == x || a == y) && (b > x && a)`
- `z >= b * 5 && y == x * 6 + 2 && !a`
- `x + a > b || z - y < b + x && z > y`

Uppgift 3 [3p]

Antag följande rad kod: `char str[50] = "Jag lär mig programmera C";`

- Vilka index (från – till) har arrayen?
- Vad finns på index 25 i arrayen?
- Vilken längd har strängen, dvs. vad returnerar `strlen(str)`?
- Vilken storlek har arrayen, dvs. vad returnerar `sizeof(str)`?
- Hur många bitar (bits) allokeras på stacken för `str`?

Uppgift 4 [8p]

Skapa en egendefinierad typ (struct) för ett bankkonto. Den nya typen ska innehålla följande information

- Clearingnummer (4-5 siffror)
 - Kontonummer (9 siffror)
 - Kontoinnehavare (för- och efternamn)
 - Saldo (ska kunna innehålla ören)
- Skriv kod som definiera den nya typen (structen).
 - Skriv kod för att skapa ett fält med 10 poster (array) av den nya typen genom att använda funktionen `malloc()`. Felhantering för allokering av dynamiskt minne är inget krav.
 - Skriv en kodrad som tilldelar värdet 398.50 till fältet "saldo" i den tredje posten. Använd punktoperatoren.
 - Skriv en kodrad som tilldelar kontonumret 959500123 till fältet "kontonummer" i sista posten. Använd piloperatoren.

Uppgift 5 [4p]

Skriv om alla if-else-if-satser i koden nedan till en switch-case-konstruktion.

```
int main(void)
{
    int val = 0;

    puts("          MENY");
    puts("-----");
    puts("1 - Lägg till post");
    puts("2 - Ta bort post");
    puts("3 - Sök efter post");
    puts("4 - Skriv ut samtliga poster");
    puts("5 - Avsluta programmet");

    scanf("%d", &val);

    if(val == 1){
        laggtill();
    }
    else if(val == 2){
        taBort();
    }
    else if(val == 3){
        sok();
    }
    else if(val == 4){
        skrivUt();
    }
    else if(val == 5){
        avsluta();
    }
    else {
        puts("Felaktigt val");
    }

    return 0;
}
```

Skriv om
denna del

Uppgift 6 [10p]

Delfråga a till b nedan handlar om följande programkod.

```
#include <stdio.h>
#define STORLEK 10

void swap(int *x, int *y);
void printArray(int arr[]);

int main(void) {
    int arr[STORLEK] = {};
    int i, j;

    printf("Mata in %d tal\n", STORLEK);
    for(i = 0; i <= STORLEK; i++)
        scanf("%d", arr[i]);

    printf("Arrayen före sortering:");
    printArray(arr);

    for(; i < STORLEK-1; i++)
        for(j = STORLEK-1; j>0; j--)
            if(arr[j] < arr[j-1])
                swap(arr[j-1], arr[j]);

    printf("Arrayen efter sortering:");
    printArray(arr);

    return 0;
}

void swap(int *x, int *y) {
    int temp = *x;
    x = y;
    *y = temp;
}

void printArray(int arr[]) {
    int i;
    for(i = 0; i < STORLEK; i++)
        printf("%d ", arr[i]);
}
```

- I koden finns ett antal fel (tankefel och fel som av en kompilator ger error). Hitta felen, skriv ner hela den felaktiga raden och visa hur den ska rättas till (ändras).
Påståenden om fel som i själva verket är korrekt kod ger poängavdrag (lägsta poäng är 0).
- Beskriv vad programmet utför. dvs. vad matas in och vad blir resultatet (inte rad för rad).
- Ge ett exempel från koden på en variabeldeklaration om det finns.
- Ge ett exempel från koden på en variabelinitiering om det finns.
- Ge ett exempel från koden på en definition om det finns.
- Vad är skillnaden mellan att lägga en implementation av en funktion före eller efter funktionen `main()`?

Uppgift 7 [4p]

Skriv kod för två olika loop-konstruktioner (två olika main-funktioner) som implementerar följande beskrivning:

*För årets alla månader under ett år, summera alla månadernas motsvarande tal. I loopen ska alla tal som summeras skrivas ut med ett + mellan talen. Avsluta med ett = och summan.
(dvs. $1+2+3+ \dots +12=78$).*

- a) Använd en for-loop
- b) Använd en do-while-loop

Uppgift 8 [5p]

Skriv ett program som bara innehåller `main()` och en funktion `create_number()`.

`create_number()`: Funktionens uppgift är att allokera minnesplats på heapen motsvarande typen `long` och spara ett slumpstal mellan 1 och 15000 på den platsen. Funktionen skall inte ta några inparametrar.

`main()`: Här ska funktionen `create_number()` anropas en gång, talet från funktionen ska dubbleras och skrivas ut på skärmen.

Felhantering för allokering av dynamiskt minne är ett krav i denna uppgift.