

FACIT!!

Omtentamen - Programmering DVA117

Akademien för innovation, design och teknik

Onsdag 2014-08-12

Skrivtid: 08.30 - 11.30
Hjälpmedel: Inga
Lärare: Caroline Uppsäll, 021-101456
Robert Suurna, 021-151790

Preliminära betygsgränser

Betyg 3: 24p
Betyg 4: 32p
Betyg 5: 37p
Max: 40p

Bonuspoäng

Inga bonuspoäng kan tas med till omtentamen.

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!

FACIT!!

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	Left to right
sizeof	Size of an object	
(type)	Type cast (conversion)	
*	Multiplication	Left to right
/	Division	
%	Modulus	Left to right
+	Addition	
-	Subtraction	Left to right
<<	Left shift	
>>	Right shift	Left to right
<	Less than	
<=	Less than or equal to	
>	Greater than	
>=	Greater than or equal to	Left to right
==	Equality	
!=	Inequality	Left to right
&	Bitwise AND	
^	Bitwise XOR	Left to right
	Bitwise OR	
&&	Logical AND	Left to right
	Logical OR	
?:	Conditional	Right to left
=		
*= /= %=	Assignment operators	Right to left
+= -= &=		
^= =		
<<= >>=	Comma operator	Right to left
,		

Uppgift 1 [7p]

- a) SV: För att göra en fil innehållande C-kod till ett körbart program går koden igenom flera steg, *preprocessorn*, *kompilatorn* och *länkaren*. Beskriv kortfattat vad som händer i de tre stegen.

EN: To make the C source code file into an executable program the code passes through a multi-step process, the preprocessor, the compiler and the linker. Briefly describe what happens in the three steps.

(1p): Preprocessorn är första steget i byggstegen, den tar hand om alla rader som börjar med # (#define, #include, etc.)

(1p): Sen följer kompilering (analysering, optimering och generering av maskinkod) som sparas i en binärfil (O-fil)

(1p): I sista steget länkas O-filen ihop med div. extra kod och skapar den slutliga körbara (exekverbara) filen

- b) SV: Vad kallar man följande rad för i C-program och hur används det?

EN: What do you call the following line in a C program and how is it used?

```
#define BASBELOPP2014 4400 + 400
```

(2p): Kallas för makro och är en identifierare som definierats att representera en viss textsträng. När preprocessorn träffar på ett makro i koden byts den delen av koden ut mot den definierade textsträngen (kallas för makroexpansion). Underlättar snabb och smidig ändring av tex. konstanter.

- c) SV: Nämn en anledning till att arrayer kan vara oflexibla och inte lämplig att använda när storleken av data varierar. Vad kan man använda istället för arrayer i denna situation?

EN: Name one reason why arrays can be inflexible and not suitable to use when the size of data may vary. What could be used instead of arrays in this situation?

(2p): När man inte vet storleken på data måste man allokera en fast storlek på arrayen som motsvarar den största möjliga storleken på datat. Går inte att ändra på under exekvering av programmet (ev. onödigt stor allokering). Istället bör dynamisk allokering av data användas.

Uppgift 2[4p]

Uppgift 3 [3p]

SV: Antag följande rad kod: `char str[] = "The summer is over!";`

EN: Assume the following line of code: `char str[] = "The summer is over!";`

- a) SV: Vilka index (från – till) har arrayen?

EN: What is the index range (from-to) for the array?

(0.5p): 0-19 (position 19 har \0)

- b) SV: Vilken längd har strängen, dvs. vad returnerar funktionen `strlen(str)`?

EN: What is the length of the string, i.e. what does the function `strlen(str)` return?

(0.5p): 19

- c) SV: Vilken storlek har arrayen, dvs. vad returnerar funktionen `sizeof(str)`?

EN: What is the size of the array, i.e. what does the function `sizeof(str)` return?

(0.5p): 20

- d) SV: Vad finns lagrat på index 14 i arrayen?

EN: What is stored at index 14 in the array?

(0.5p): o (lilla bokstaven o)

- e) SV: Vad händer om man kör följande rad?

EN: What happens if you execute the following line?

```
*(str+4)=str[3];?
```

(1p): Bokstav s i summer ersätts med ett tomt tecken, det som finns på position 3 i arrayen

Uppgift 4 [9p]

Du har blivit ombedd att skapa ett program som håller reda på samt utför beräkningar på en persons löparrundor. **Den** Det data som ska kunna sparas om varje runda är datum, distans, tid samt hastighet.

- a) Skriv koden för en lämplig typ (struct) som kan hålla informationen om en löprunda enligt beskrivningen ovan. Datum kan sparas som ett heltal (ex ÅÅMMDD), distans ska anges som ett flyttal och mätas i km, tid ska anges som hela minuter, hastigheten ska anges som ett flyttal och mätas i km/h. --2p
- b) Skriv koden för att skapa ett fält (array) för en person som kan hålla 30 löprundor genom att använda funktionen `malloc()` eller `calloc()`. Felhantering för allokering av dynamiskt minne är ett krav. --2p
- c) Skapa en funktion som sätter datum, distans och tid för en löprunda. Den aktuella löprundan är nästa lediga index i arrayen. Funktionen ska också beräkna och sätta hastigheten för rundan. Funktionen ska som inparametrar minst ta datum, distans och tid. Returvärde från funktionen ska vara 1 om operationen lyckades och 0 om den inte lyckades. Du vet inte om det redan ligger data i arrayen eller inte men du kan anta att datum för ej satta element är 0. --4p
- d) Skriv en kodrad som tilldelar datum **till** 140812 till löprundan på den sista posten. Använd piloperatorn. --1p

Uppgift 5 [10p]

Uppgift 6 [7p]

Delfråga a till b nedan handlar om följande programkod.

```

#include <stdlib.h>
#include <stdio.h>

void utskrift()
{
    switch(deltal)
    {
        case 0: printf("noll "); break;
        case 1: printf("ett "); break;
        case 2: printf("två "); break;
        case 3: printf("tre "); break;
        case 4: printf("fyra "); break;
        case 5: printf("fem "); break;
        case 6: printf("sex "); break;
        case 7: printf("sju "); break;
        case 8: printf("åtta "); break;
        case 9: printf("nio "); break;
        default: break;
    }
}

float plockaUtDeltal(int deltal[4], int tal, int plats)
{
    if(tal>0)
    {
        deltal[plats] = tal%10;
        tal = tal/10;
    }
    return tal;
}

int main(void)
{
    int i; tal = 0;
    int deltal[4] = {-1,-1,-1,-1};

    printf("Ange ett heltal mellan 0 och 9999: ");
    scanf("%d", tal);

    for( ; i<4; i++)
        tal = plockaUtDeltal(deltal, tal, i);

    for(i = 3; i>=0; i--)
        utskrift(deltal[i]);

    return 0;
}

```

- I koden finns ett antal fel (tankefel och fel som av en kompilator ger error). Det kan också vara någonting som saknas. Hitta felen, skriv ner hela den felaktiga raden och visa hur den ska rättas till (ändras). --5p
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). --2p
- Ge ett exempel från koden på en variabeldeklaration om det finns. --1p
- Ge ett exempel från koden på en variabelinitiering om det finns. --1p
- Ge ett exempel från koden på en definition om det finns. --1p