

# Omtentamen - Programmering

## Re-exam - Programming

### DVA117

*Akademien för innovation, design och teknik*  
*School of innovation, design and technology*

*Torsdag / Thursday 2014-08-12*

**Skrivtid / Write time:** 08.30 – 11.30  
**Hjälpmedel / Accessibility:** Inga / None  
**Lärare / Teacher:** Caroline Uppsäll, 021-101456  
Robert Suurna, 021-151790

#### **Preliminära betygsgränser / Preliminary grading limits**

Betyg 3 / Grade 3: 24p  
Betyg 4 / Grade 4: 32p  
Betyg 5 / Grade 5: 37p  
**Max: 40p**

#### **Bonuspoäng / Bonus Points**

Inga bonuspoäng kan tas med till omtentamen. / *No bonus points can be used on the re-exam.*

#### **Allmänt / Generally**

- All kod skall skrivas i standard ANSI C. / *All code should be written in 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. / *Write each question on a new paper and do only use one side of the paper. Subquestions (a, b, c,...) can be written on the same paper.*
- Referera inte mellan olika svar. / *Do not reference between answers.*
- Om du är osäker på vad som avses i någon fråga, skriv då vad du gör för antagande. / *If you are unsure of a meaning of a question, write down your assumption.*
- Oläsliga/oförståeliga/ostrukturerade svar rättas inte. / *Illegible/inintelligible/unstructured answers will not be marked.*
- Kommentera din kod! / *Comment your code!*
- Tips: Läs igenom hela tentan innan du börjar skriva för att veta hur du ska disponera din tid. / *Tip: Read through the entire exam before you start writing to know how you should allocate your time.*

*Lycka till! / Good luck!*

# Prioritet och associativitet hos operatorerna i C

## *Precedence and associativity of operators in C*

Högsta prioritet högst upp i tabellen, lägsta prioritet längst ner i tabellen. Grupperade operatorer har samma prioritet. / *The highest priority is at the top of the table, the lowest priority is at the bottom of the table. Grouped operators have the same priority.*

**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] / Question 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. (3p)

*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. (3p)

- b) *SV*: Vad kallar man följande rad för i C-program och hur används det? (2p)

*EN*: What do you call the following line in a C program and how is it used? (2p)

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

- 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? (2p)

*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)

## Uppgift 2 [4p] / Question 2 [4p]

SV: Vad returnerar följande logiska uttryck/logik (i C)? Sant(1) eller Falskt(0)?

För samtliga uttryck gäller: `int x=40, y=2, z=4, a=1, b=6;`

EN: What will the following logical expressions/logic (in C) return? True (1) or False (1)?

For all subquestions: `int x=40, y=2, z=4, a=1, b=6;`

a) `return (z != 8 && x > y * b)` (1p)

b) `return ((a == x || b != y + z) && !(b > x && a == 1))` (1p)

c) `if(y < z && a*b > x)` (2p)

```
{
    x = x/2;
    z = b+y;
}
else
    while(z*10 <= x || y+z > b)
    {
        x++;
        y = y+2;
        if(x == 45)
            break;
    }
if (y+x > 50)
    return 1
else
    return 0
```

### Uppgift 3 [3p] / Question 3 [3p]

SV: Antag följande rad av C-kod:

EN: Assume the following line of C code:

```
char str[] = "The summer is over!";
```

- a) SV: Vilka index (från – till) har arrayen? (0,5p)  
EN: What is the index range (from-to) for the array? (0,5p)
  
- b) SV: Vilken längd har strängen, dvs. vad returnerar funktionen `strlen(str)`? (0,5p)  
EN: What is the length of the string, i.e. what does the function `strlen(str)` return? (0,5p)
  
- c) SV: Vilken storlek har arrayen, dvs. vad returnerar funktionen `sizeof(str)`? (0,5p)  
EN: What is the size of the array, i.e. what does the function `sizeof(str)` return? (0,5p)
  
- d) SV: Vad finns lagrat på index 14 i arrayen? (0,5p)  
EN: What is stored at index 14 in the array? (0,5p)
  
- e) SV: Vad händer om man kör följande rad? (1p)  
EN: What happens if you execute the following line? (1p)  
`*(str+4)=str[3];?`

#### Uppgift 4 [9p] / Question 4 [9p]

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

*EN:* You've been asked to create a program that keeps track of and calculates performance of a person running. For each run the following data should be able to save: date, distance, time and speed.

- a) *SV:* 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)

*EN:* Write the code for the appropriate type (struct) that can hold information about a run as described above. Dates can be stored as an integer (for example YYMMDD), the distance must be specified as a float and be measured in kilometers, time to be given as whole minutes, the speed shall be specified as a float and be measured in kilometers per hour. (2p)

- b) *SV:* 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)

*EN:* Write the code to create an array that can hold 30 runs for one person by using the function `malloc()` or `calloc()`. Error handling for allocation of dynamic memory is required. (2p)

- c) *SV:* 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)

*EN:* Create a function that sets the date, distance and time for a run. The current run will be the next free index in the array. The function should also calculate and set the speed for the run. At a minimum the function should take the date, distance and time as parameters and return 1 (true) if the operation succeeded and 0 (false) if it did not succeed. You do not know if the array is empty or if there already exist runs in it but you can assume that the index is free if the date is set to 0. (4p)

- d) *SV:* Skriv en kodrad som tilldelar datum 140812 till löprundan på den sista posten. Använd piloperatorn. (1p)

*EN:* Write a line of code that assigns the date 140812 to your run on the last index in the array by using the arrow operator ( $\rightarrow$ ). (1p)

## Uppgift 5 [10p] / Question 5 [10p]

SV: Delfråga a till b nedan handlar om följande programkod.

EN: Subquestion a to b below deals with the following program.

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

void output()
{
    switch(subNumber)
    {
        case 0: printf("zero "); break;
        case 1: printf("one "); break;
        case 2: printf("two "); break;
        case 3: printf("three "); break;
        case 4: printf("four "); break;
        case 5: printf("five "); break;
        case 6: printf("six "); break;
        case 7: printf("seven "); break;
        case 8: printf("eight "); break;
        case 9: printf("nine "); break;
        default: break;
    }
}

float funk(int subNumber[4], int number, int position)
{
    if(number>0)
    {
        subNumber[position] = number%10;
        number = number/10;
    }
    return number;
}

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

    printf("Enter an integer between 0 and 9999: ");
    scanf("%d", number);

    for( ; i<4; i++)
        number = funk(subNumber, number, i);

    for(i = 3; i>=0; i++)
        output(subNumber[i]);

    return 0;
}
```

SV: Frågorna finns på nästa sida

EN: You will find the questions on the next page

- a) *SV*: 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).*

*EN*: In the program above there are a number of errors (cognitive biases and errors that the compiler will find). It can also be something missing from the code. Find the faults, write down the faulty line and show how it should be corrected. (5p)

*Claims of errors that is in fact correct code gives points deduction (minimum score in 0).*

- b) *SV*: Beskriv vad programmet utför. dvs. vad matas in och vad blir resultatet (inte rad för rad). (2p)

*EV*: Describe what the program is doing. ie. what is input and what the result will be (not line by line). (2p)

- c) *SV*: Ge ett exempel från koden på en variabeldeklaration om det finns. (1p)

*EN*: Give an example, from the code above, of an variable declaration, if available. (1p)

- d) *SV*: Ge ett exempel från koden på en variabelinitiering om det finns. (1p)

*EN*: Give an example, from the code above, of an initiation on an variable, if available.(1p)

- e) *SV*: Ge ett exempel från koden på en definition om det finns. (1p)

*EN*: Give an example, from the code, of a definition, if available. (1p)



### Uppgift 6 [7p] / Question 6 [7p]

- a) *SV:* Deklarera en pekare till ett heltal som heter *address*. (1p)  
*EN:* Declare a pointer to an integer called *address*. (1p)
- b) *SV:* Tilldela flyttalspekaren *temp* till adressen där flyttalsvariabeln *balance* ligger. (1p)  
*EN:* Assign the address of a float variable *balance* to the float pointer *temp*. (1p)
- c) *SV:* Tilldela variabeln som charpekaren *letter* pekar på till tecknet 'C'. (1p)  
*EN:* Assign the character value 'C' to the variable pointed to by the char pointer *letter*. (1p)

- d) *SV:* Vilken output ger följande programsegment? (1p)  
*EN:* What is the output of the following program segment? (1p)

```
int count = 10, temp, sum = 0;
temp = &count;
temp = 20;
temp = &sum;
*temp = count;
printf("count = %d, *temp = %d, sum = %d\n", count, *temp, sum);
```

- e) *SV:* Deklarera en pekare (*message*) till textsträngen "Hello". (1p)  
*EN:* Declare a pointer to the text string "Hello" called *message*. (1p)
- f) *SV:* Antag att en funktion tar in en heltalsarray, beräknar summan av alla element i arrayen samt tar reda på minsta och största heltalet. Denna data (summa, största och minsta) ska sedan returneras ur funktionen för att tas om hand någon annan stans i programmet. Skriv ett förslag till hur funktionshuvudet samt retursatsen ska se ut. (2p)

*EN:* Suppose that a function takes an array of integers as parameter, calculates the sum of all elements in the array and determine the smallest and largest integer. This data (sum, largest and smallest) are then returned from the function to be taken care of elsewhere in the program. Write a proposal for how the function header and return statement will look. (2p)