



# Tentamen - Programmering

DVA117

-----FACIT-----

*Akademien för innovation, design och teknik*

*Tisdag 2018-11-06*

*An English translation of the entire exam follows after the questions in Swedish*

**Skrivtid:** 14.10 – 19.30

**Hjälpmedel:** Valfritt icke-elektroniskt material

**Lärare:** Caroline Uppsäll  
(kan nås på telefon om du frågar tentavakten)

## **Preliminära betygsgränser**

Betyg 3: 15p

Betyg 4: 23p

Betyg 5: 27p

**Max: 30p**

## **Allmänt**

- All kod skall skrivas i standard ANSI C.
- Skriv tydligt vilken uppgift/deluppgift ditt svar anser.
- Skriv endast på bladets ena sida.
- 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!
- Det är inte tillåtet att använda goto-satser och globala variabler
- Tips: Läs igenom hela tentamen innan du börjar skriva för att veta hur du ska disponera din tid.
- Potentiellt intjänade bonuspoäng kan endast användas på kursinstansens ordinarie tentamen.

*Lycka till!*  
*/Caroline*

# -----FACIT-----

*Notera: alla lösningar bedöms individuellt och förändringar från nedanstående poängsättning (inom uppgiften) kan förekomma beroende på hur resterande svar ser ut.*

## Uppgift 1 [3p]

Ange för nedanstående a-f vilken typ de tillhör (char, int, float, char\*/char[], ogiltig datatyp).

- a) 72 - int
- b) '\n' - char
- c) -102.467 - float
- d) "true" -char\*/char[]
- e) & - ogiltig
- f) "T" - char\*/char[]

## Uppgift 2 [3p]

Utgå ifrån nedanstående program. Vilken blir utskriften för nedanstående input?

```
#include <stdio.h>

int main(void)
{
    float tal;

    printf("\nAnge tal: ");
    scanf("%f", &tal);

    if(tal >= 0.0 && tal < 10.0 && tal != 5.0)
    {
        printf("1");
    }
    else if(tal >= 11.5 || (tal < -1.5 && !(tal == -3.0)))
    {
        printf("2");
    }
    else
    {
        printf("3");
    }
    return 0;
}
```

- a) 5.0     -3
- b) -5.0    -2
- c) 5.5     -1
- d) -3.0    -3
- e) -3.1    -2
- f) 10.0    -3

## Uppgift 3 [3p]

Nedan finns ett program som använder en for-konstruktion. Din uppgift är att skriva om samma sak men med en do-while-konstruktion istället.

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

```

int main(void)
{
    int i, sum;
    for(i = 1, sum = 0; sum <= 10; i++, sum = sum + i)
    {
        if(sum%2 == 1)
            i++;
        else
            i--;

        printf("%d ", sum);
    }
    return 0;
}

```

**Svar:**

```

int main(void)
{
    int i = 1, sum = 0; //initiering (1p)
    do{
        if(sum%2 == 1)
            i++;
        else
            i--;

        printf("%d ", sum);
        i++; //förändring (0.5p)
        sum = sum + i; //förändring (0.5p)
    }while(sum <= 10); //villkor (1p)

    return 0;
}

```

**Uppgift 4 [4p]**

Hur ser utskriften ut när nedanstående program körts färdigt?

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

```

int f1(int x,int y)
{
    x++;
    printf("x: %d y: %d \n", x, y);
    return y+x;
}
int f2(int *d,int b)
{
    d = &b;
    *d = 5;
    printf("*d: %d b: %d \n", *d, b);
    return 0;
}
int main(void)
{
    int a=10, b=5, c=1, d=-3;
    a = c - d;
    b = f1(b+2,a);
    printf("a:%d b:%d c:%d d:%d\n", a, b, c, d);
    c = f2(&d,b);
    printf("a: %d b:%d c:%d d:%d\n", a, b, c, d);
}

```

**Svar:**

x: 8 y: 4  
a:4 b:12 c:1 d:-3  
\*d: 5 b: 5  
a: 4 b:12 c:0 d:-3

1p per korrekt rad

Har man svarat med

x: 8 y: -2  
a:-2 b:6 c:1 d:-3  
\*d: 5 b: 5  
a: -2 b:6 c:0 d:-3

så ger rad 1, 2 och 4 0.5p vardera och rad 3 1poäng.

$$1 - (-3) = 1 + 3 = 4$$

**Uppgift 5 [3p]**

- a) Deklarera en statisk heltalsarray som har plats för 10 element och fyll den bakifrån (det sista elementet ska alltså sättas först och det första elementet ska sättas sist) med input från användaren. (1.5p)
- b) Deklarera en sträng som max kan innehålla 20 tecken (inkl strängslutstecknet '\0'), initiera strängen till "Hello world". (1p)
- c) Hur lång är strängen? (0.5p)

**Svar:**

- a) `int arr[10];` (0,5p)  
`for(int i = 9; i >= 0; i--)`  
`scanf("%d", &arr[i]);` (→ totalt med for-loop 1p)
- b) `char str[20] = "Hello world";` (0.5p + 0.5p)
- c) 11 tecken (0,5p) – längden är strängen utan '\0', storleken på strängen är längden + '\0'

**Uppgift 6 [10p]**

- a) Definiera en datatyp som beskriver en hiss, typen ska heta Elevator. En hiss ska ha ett ID-nr (heltal) samt en siffra som beskriver vilken våning den befinner sig på samt beskrivning 'U' eller 'N' beroende på om den är påväg uppåt eller nedåt. (1p)
- b) Skriv en funktion main som dynamiskt allokerar minne för 3 hissar. Arrayen av hissar ska heta elevatorsBuildning1. Sätt den andra hissens:
  - ID till 101 med hjälp av -> operatoren.
  - våning till att den är på fjärde våningen, använd punktoperatoren (.)
  - läga till att den är påväg uppåt, använd valfritt sätt.(3p)
- c) Skapa en funktion som utökar det allokerade minnet med ett element/en hiss. Funktionen ska heta addMemory och ta pekaren till arrayen av hissar samt det befintliga antalet hissar som parameter. Funktionen ska returnera en pekare till det nya, större minnet. Ett anrop kan se ut som följer:  
`elevatorsBuildning1 = addMemory(elevatorsBuildning1, 3);`  
Efter anropet (ovan) ska arrayen elevatorsBuildning1 bestå av 4 hissar.  
Om det inte går att lägga till nytt minne ska en felutskrift göras och pekaren till den gamla minnet returneras. (3p)
- d) Förklara varför funktionen addMemory, som du skrev i deluppgift c), returnerar pekaren till det nya minnet – en pekare till minnet skickas in som argument till funktionen – hur kommer

## FACIT /SOLUTIONS

det sig att det inte räcker (svara med max 5 meningar, det går bra att använda bilder i svaret). (1p)

- e) Skriv en funktion som skriver hela innehållet i arrayen av hissar till en binärfil som heter "elevators.bin". Du ska använda följande funktionshuvud:  
void writeToBinaryFile(struct Elevator \*e, int nrOfElevators); (2p)

### Svar:

a)

```
struct elevator {
    int id;
    int floor;
    char upDown;
};
```

b)

```
int main(void)
{
    struct Elevator *elevatorsBuilding1 = (struct Elevator*)calloc(3,
                                                                    sizeof(struct Elevator)); (0.5p)
    if(elevatorsBuilding1 == NULL) (0.5p)
        printf("error");
    else
    {
        (elevatorsBuilding1+1)->id = 101; (1p)
        elevatorsBuilding1[2].floor = 4; (0.5p)
        elevatorsBuilding1[2].upDown = 'U'; (0.5p)
    }
    free(elevatorsBuilding1);
}
```

c)

```
struct Elevator* addMemory(struct Elevator *e, int sizeOld) (0.5p)
{
    struct Elevator* temp = (struct Elevator*)realloc(e, sizeof(struct
                                                         Elevator)*(sizeOld+1)); (temp = 1p, realloc = 0.5p)
    if(temp != NULL) (0.5p)
        return temp;
    else
    {
        printf("Error creating new memory");
        return e; (0.5p)
    }
}
```

d). pekaren e är lokal i funktionen, om realloc tvingas lägga minnet någon annanstans än där det låg från början så kommer den lokala pekaren e peka på det nya minnet, den pekare som skickas in (elevatorsBuilding1) kommer fortfarande att peka på det tidigare minnet. (1p)

e).

```
void writeToBinaryFile(struct Elevator *e, int nrOfElevators)
{
    FILE *fp = fopen("elevators", "wb"); (0.5p)
    if(fp != NULL) (0.5p)
        fwrite(e, sizeof(struct Elevator), nrOfElevators, fp); (0.5p)

    fclose(fp); (0.5p)
}
```

**Uppgift 7 [2p]**

I nedanstående program finns ett problem som leder till att utskriften blir

```
U
0.000000
```

Vilket är problemet?

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

int main(void)
{
    FILE *fp = fopen("fil.txt", "w");
    char str1[30], str2[7];
    float p;

    if(fp != NULL)
    {
        fputs("programmering med C", fp);
        fprintf(fp, "\n%s\n%.1f\n", "DVA117", 7.5);
    }

    fp = fopen("fil.txt", "r");
    if(fp != NULL)
    {
        fgets(str1, 30, fp);
        fgets(str2, 7, fp);
        fscanf(fp, "%f", &p);
    }

    fclose(fp);

    printf("%s\n%s\n%f\n", str1, str2, p);

    return 0;
}
```

**Svar:**

Problemet är att filen (som är öppen i w-läge) inte stängs innan man försöker öppna den i r-läge). Lägg in raden `fclose(fp);` innan raden `fp = fopen("fil.txt", "r");` (2p)

**Uppgift 8 [2p]**

När vi bygger vårt program sker tre saker innan vi får ut en exekverbar fil, koden går igenom pre-processor, kompilatorn och länkaren. Förklara vad dessa tre saker gör (max 2 meningar per del).

**Svar:**

Preprocessor (0.5p) – hanterar alla # (includes och defines)

Kompilatorn (1.5p) – analyserar och optimerar koden samt genererar maskinkod (.o-fil)

Länkaren (1p) – länkar ihop .o-filerna med biblioteksfilerna och skapar en körbar fil (.exe)

## Exam - Programming

DVA117

# -----Solutions-----

*School of Innovation, design and technology*

*Tuesday 2018-11-06*

**Writing time:** 14.10 – 19.30

**Aids:** Any non-electronic material

**Examiner:** Caroline Uppsäll

(Can be reached by telephone if you ask the exam guard)

### **Preliminary grading limits**

Grade 3: 15p

Grade 4: 23p

Grade 5: 27p

**Max: 30p**

### **Generally**

- All code should be written in standard ANSI C.
- Write clearly what task/sup-task your answers consider.
- Do only use **one** side of the paper.
- Do not refer between answers.
- If you are unsure of a meaning of a question, write down your assumption.
- *Unreadable/incomprehensible answers will not be marked.*
- Comment your code!
- It is not allowed to use goto-statements or global variables
- Hint: To know how to allocate your time, read through the entire exam before you start writing.

# -----Solutions-----

***For solutions see the Swedish version***

*Good luck!*

*/Caroline*

**Question 1 [3p]**

For the following data below (a-f) state which type they have (char, int, float, char\*/char[], undefined)

- a) 72
- b) '\n'
- c) -102.467
- d) "true"
- e) &
- f) "T"

**Question 2 [3p]**

Based on the program below, state the output the program makes based on the different input given in a – f.

```
#include <stdio.h>

int main(void)
{
    float tal;

    printf("\nAnge tal: ");
    scanf("%f", &tal);

    if(tal >= 0.0 && tal < 10.0 && tal != 5.0)
    {
        printf("1");
    }
    else if(tal >= 11.5 || (tal < -1.5 && !(tal == -3.0)))
    {
        printf("2");
    }
    else
    {
        printf("3");
    }
    return 0;
}
```

- a) 5.0
- b) -5.0
- c) 5.5
- d) -3.0
- e) -3.1
- f) 10.0

**Question 3 [3p]**

Below is a program that uses a for-construction to create a loop behavior. Your job is to write the same thing using a do-while construction.

```
#include <stdio.h>

int main(void)
{
```



```
int i, sum;
for(i = 1, sum = 0; sum <= 10; i++, sum = sum + i)
{
    if(sum%2 == 1)
        i++;
    else
        i--;

    printf("%d ", sum);
}
return 0;
}
```

#### Question 4 [4p]

What will the output look like when the following program is finished executing?

```
#include <stdio.h>

int f1(int x,int y)
{
    x++;
    printf("x: %d y: %d \n", x, y);
    return y+x;
}
int f2(int *d,int b)
{
    d = &b;
    *d = 5;
    printf("*d: %d b: %d \n", *d, b);
    return 0;
}
int main(void)
{
    int a=10, b=5, c=1, d=-3;
    a = c - d;
    b = f1(b+2,a);
    printf("a:%d b:%d c:%d d:%d\n", a, b, c, d);
    c = f2(&d,b);
    printf("a: %d b:%d c:%d d:%d\n", a, b, c, d);
}
```

#### Question 5 [3p]

- Declare an array that can hold 10 integers (not with dynamic memory) and fill it with input from the user where the last integer in the array gets input first and the first integer in the array get the last input (fill it backwards). (1,5p)
- Declare a string that can hold a maximum of 20 characters (including '\0'). Initiate the string to "Hello world". (1p)
- What's the length of the string in b)? (0.5p)

#### Question 6 [10p]

- Define a new datatype that describes an elevator, the name of the new type should be elevator. An elevator should be described with an ID (integer), the level it's on and a 'U' or 'D' that describes if it's on it's way up or down. (1p)
- Write a function main that dynamically allocates an array of 3 elevators. The array of elevators should be named elevatorsBuilding1. Set the following information in the second elevator in the array:

The ID should be 101, use the -> operator.

The elevator is on the fourth floor, use the . operator.

The elevator is on it's way up.

(3p)

- c) Create a function that adds dynamic memory to the array you created in b). The name of the function should be addMemory and take a pointer to the array of elevators as well as the number of elevators in the array as parameters. The function should return a pointer to the new, bigger memory that contains one additional element/elevator. The function call could look like:  
`elevatorsBuildning1 = addMemory(elevatorsBuildning1, 3);`  
 After the function call the array elevatorsBuildning1 should contain 4 elevators. (3p)
- d) Explain why the function addMemory that you created in c) above returns a pointer to the new memory – a pointer to the memory is entered as input to the function shouldn't that be enough? (Answer with a max of 5 sentences, pictures is ok). (1p)
- e) Write a function that prints all of the content in the elevator array to a binary file named "elevators.bin". You should use the following function declaration:  
`void writeToBinaryFile(struct elevator *e, int nrOfElevators);` (2p)

## Question 7 [2p]

In the following program there's a problem that leads to the following output. Explain what that problem is.

```

U
0.000000
  
```

```

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

int main(void)
{
    FILE *fp = fopen("fil.txt", "w");
    char str1[30], str2[7];
    float p;

    if(fp != NULL)
    {
        fputs("programming med C", fp);
        fprintf(fp, "\n%s\n%.1f\n", "DVA117", 7.5);
    }

    fp = fopen("fil.txt", "r");
    if(fp != NULL)
    {
        fgets(str1, 30, fp);
        fgets(str2, 7, fp);
        fscanf(fp, "%f", &p);
    }

    fclose(fp);

    printf("%s\n%s\n%f\n", str1, str2, p);

    return 0;
}
  
```

}

**Question 8 [2p]**

When we build a program three things happen before we get the executable file, the code goes through the pre-processor, the compiler and the linker. Explain these three steps (max 2 sentences per step).