



Tentamen - Programmering

DVA117

-----FACIT-----

Akademien för innovation, design och teknik

Måndag 2018-03-19

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, 0704616110
(kan nås på telefon om du frågar tentavakten)

Preliminära betygsgränser

Betyg 3: 14p

Betyg 4: 21p

Betyg 5: 25p

Max: 28p

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.

Lycka till!

/Caroline

-----FACIT-----

Uppgift 1 [1p]

Vad är skillnaden på följande två variabler (x1 och x2)? Förklara.

```
int x1;
float x2;
```

Svar:

x1 kan endast innehålla heltal medan x2 även kan innehålla decimaltal. Ett heltal i x2 lagras som t.ex. 10.0

Uppgift 2 [1p]

Vilken är skillnaden i att använda if-else if-else och switch-case för att utföra selektion (val) i ett program – mer än att de rent syntaktiskt ser olika ut.

Svar:

Switch-case gör val baserat på ett värde (heltalsvärde) medan if-satsen gör val baserat på huruvida ett villkor är sant eller falskt.

Uppgift 3 [1p]

```
int a = 10, b = 25, c = 15, d = 5;
```

```
((b%2 == 0) || (a>=d-c)) && ((d<c%10) || ((1==1) && !(a!=b+c+d)))
```

Vad evaluerar det logiska uttrycket till?

Svar:

0 – falskt

```
((25%2 == 0) || (10>=5-15)) && ((5<15%10) || ((1==1) && !(10!=25+15+5)))
(( 1 == 0) || (10>= -10)) && ((5< 5 ) || (( 1 ) && !(10!= 45 )))
( falskt || sant ) && ( falskt || ( sant && !( sant )))
( falskt || sant ) && ( falskt || ( sant && falskt ))
( sant ) && ( falskt || falskt )
sant && falskt
falskt
```

Uppgift 4 [4p]

Nedan används en while-loop för att åstadkomma ett loop-beteende. Skriv om loop-beteendet (koden innan för `/******LOOP*****`) på två olika sätt. Dels med hjälp av for-konstruktionen och dels med hjälp av do-while konstruktionen. Du kan anta att användaren inte skriver in negativa värden till start och decrementStep. Du kan också anta att inputen till decrementStep inte är 0, däremot kan inputen till start vara 0.

```
#include <stdio.h>
int main(void)
{
    int start, decrementStep, i;
    scanf("%d", &start);
    scanf("%d", &decrementStep);

    /******LOOP*****/
    i = start;

    while(i > 0)
    {
        //do something, tex. printf("%d ", i);
        i = i - decrementStep;
    }
    /******LOOP*****/
}
```

Svar:

For-loopen: 0.5p per korrekt initiering, villkor, förändring (totalt 1.5p)

```
-----
for(i = start; i > 0; i -= decrementStep)
{
    printf("%d ", i);
}
-----
```

do-while: 0.5p per korrekt initiering, villkor, förändring
+ 1p för korrekt hantering av om start är 0 (totalt 2.5p)

```
-----
i = start;
do
{
    if(i == 0)
        break;
    printf("%d ", i);
    i-=decrementStep;
}while(i > 0);
-----
```

Uppgift 5 [2p]

Antag att du har en binärfil som du öppnar med ett program som är till för att läsa textfiler (t.ex. notepad eller TextEdit). Informationen i filen kommer då upplevas som korrupt. Förklara varför (använd inte mer än 10 meningar).

Svar:

Programmet du öppnar filen med förväntar sig en textfil, alltså en fil som innehåller tecken och kommer därför att översätta de binära talen som finns i filen, mha, ASCII-tabellen till tecken som presenteras på skärmen. Ett exempel: om vi sparar talet 97 i den binära filen så lagras den binära representationen av 97 (01100001)(64+32+1) i filen. Om denna läses med en texteditor så kommer 01100001 översätts mha ASCII-tabellen till tecknet 'a', vilket presenteras på skärmen.

Uppgift 6 [1p]

Förklara med några få meningar (max 5) vad en algoritm är.

Svar:

En algoritm är en beskrivning av en funktionalitet eller lösningen av ett problem. Kan ses som ett recept. Algoritmen beskriver i vilken ordning olika saker ska göras för att nå ett önskvärt resultat.

Uppgift 7 [4p]

Skapa funktionen `fillArray`.

Arrayen ska fyllas med slumpade heltal mellan `minLimit` och `maxLimit` (inklusive dessa).

Om `minLimit` är 5 och `maxLimit` är 10 ska alltså talen 5, 6, 7, 8, 9 och 10 kunna slumpas.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 10

int main(void)
{
    int arr[SIZE];
    int minLimit, maxLimit;
    printf("Enter limits for randomized integers: ");
    scanf("%d %d", &minLimit, &maxLimit);

    fillArray(arr, minLimit, maxLimit);

    return 0;
}
```

Lösningsförslag:

```
Funktionshuvud – 1p
Loop – 0.5p
Srand (startvärde) – 1p
Rand() – 0.5p
Korrekt intervall – 1p

void fillArray(int *arr, int min, int max)
{
    srand(time(0));
    int i;
    for(i = 0; i < SIZE; i++)
    {
        arr[i] = rand()%(max+1-min)+min;
    }
}
```

Slumpningen steg för steg:

min = 10
max = 100

rand()%max → 0 till 99
rand()%max+min → 0+10 till 99+10 → 10 till 109
rand()%(max-min) → rand()%90 → 0 till 89
rand()%(max-min)+min → 0+10 till 89+10 → 10 till 99
rand()%(max+1-min) → rand()%91 → 0 till 90
rand()%(max+1-min)+min → 0+10 till 90+10 → 10 till 100

Uppgift 8 [2p]

Förklara (med ord och potentiellt bild) hur man skulle kunna göra för att ta reda på hur många meningar en sträng består av. En mening kan avslutas med punkt (.), frågetecken (?) eller utropstecken (!).

Svar:

Man kan loopa igenom strängen element för element (tecken för tecken). Om man hittar en punkt, ett frågetecken eller ett utropstecken så ökar man en (från början nollställd) variabel som håller reda på antal meningar. Loopen körs till strängslutet (\0).

En mening som inte avslutas med . ? eller ! räknas inte som en fullständig mening och räknas därför inte.

Uppgift 9 [3p]

Skriv en funktion som byter plats på två heltal. Visa också hur funktionsanropet ser ut i nedanstående program. Funktionens returtyp ska vara void.

Om input till nedanstående program är: 10 och 20 så ska utskrifterna bli:

Before funk: 10 and 20

After funk: 20 and 10

```
#include <stdio.h>

int main(void)
{
    int number1, number2;
    printf("Enter two numbers: ");
    scanf("%d %d", &number1, &number2);

    printf("Before funk: %d and %d", number1, number2);
    //funktionsanrop
    printf("After funk: %d and %d", number1, number2);

    return 0;
}
```

Lösningsförslag:

Funktionshuvud – 1p
Byte av tal i variabler – 1p
Anrop – 1p

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

Anrop: swap(&number1, &number2);

Uppgift 10 [3p]

När vi bygger vårt program sker tre saker innan vi får ut en exekverbar fil, en av dem är kompilering (som faktiskt består av två saker). Förklara vilka dessa tre saker är och vad de gör.

Svar:

Preprocessorn (1p) – hanterar alla # (includes och defines)

Kompilatorn (1p) – analyserar och optimerar koden samt genererar maskinkod (.o-fil)

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

Uppgift 11 [6p]

I nedanstående program finns totalt 12 fel, både syntax, semantik och logik. Vissa fel kan ligga på samma rad och vissa existerar på flera rader. Existerar felet på flera rader räknas det fortfarande som ett fel.

Ange för varje fel du hittar vilken/vilka rad/rader det finns på samt hur satsen ska se ut.
0,5p per korrekt hittat och rättat fel.

*/*Programmet ska allokera minne för 10 filmer och sedan fylla alla filmer med information om titel och produktionsår*/*

```

1      #include <stdio.h>
2      #include <stdlib.h>
3
4      struct movie
5      {
6          char title[STRSIZE];
7          int productionYear;
8      };
9
10     typedef struct
11     {
12         movie *list;
13         int numberOf;
14     }movieList;
15
16     int main(void)
17     {
18         movieList *myMovies;
19         myMovies->numberOf = 10;
20         myMovies = (movieList*)calloc(myMovies->numberOf, sizeof(movieList));
21
22         if(myMovies->list == NULL)
23         {
24             printf("Enter movie information\n-----\n");
25             addInformation(&myMovies);
26             free();
27         }
28         return 0;
29     }
30
31     void addInformation(movieList *myMovies)
32     {
33         for(i = 0; i <= myMovies->numberOf; i++)
34         {
35             printf("Title:");
36             fgets(myMovies[i]->list.title, STRSIZE, stdin);
37             printf("ProductionYear: ");
38             scanf("%d", myMovies[i]->list.productionYear);
39         }
40     }

```

Svar:

Rad 6 & 36: #define STRSIZE 20 (0.5p)
 Rad 12: struct movie (0,5p) || rad 4: typedef struct{...}movie;
 Rad 20: myMovies->list =
 (struct movie*)calloc(myMovies->numberOf, sizeof(struct movie));
 (0.5p + 0.5p)
 Rad 23: if(myMovies->list != NULL) (0.5p)
 Rad 25: addInformation(myMovies) (0.5p)
 Rad 26: free(myMovies->list) (0.5p)
 Rad 31: funktionen är ej deklarerad || ska ligga ovanför main (0.5p)
 Rad 33: for(int i = 0; i < myMovies->numberOf; i++) (0.5p + 0.5p)
 Rad 36: 38: myMovies->list[i].title (0.5p)
 Rad 38: scanf("%d", &(myMovies[i]->list.productionYear)); (0,5p)

Exam - Programming

DVA117

-----FACIT-----

School of Innovation, design and technology

Monday 2018-03-19

Writing time: 14.10 – 19.30

Aids: Any non-electronic material

Examiner: Caroline Uppsäll, 0704616110
(Can be reached by telephone if you ask the exam guard)

Preliminary grading limits

Grade 3: ??p

Grade 4: ??p

Grade 5: ??p

Max: ??p

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.

-----FACIT-----

Good luck!

/Caroline

Question 1 [1p]

What is the difference of the two variables below (x1 and x2)? Explain.

```
int x1;
float x2;
```

Answer:

x1 can only hold integer numbers, x2 can also hold decimals. An integer in x2 will be stored as for example 10.0

Question 2 [1p]

What is the difference of using if-else if-else and switch-case to create selection in a program – more than that the syntax looks different.

Answer:

Switch-case makes choices based on a value (integer), if-else makes choices based on if a condition is true or false.

Question 3 [1p]

```
int a = 10, b = 25, c = 15, d = 5;
```

```
((b%2 == 0) || (a>=d-c)) && ((d<c%10) || ((1==1) && !(a!=b+c+d)))
```

What is the result of the logical expression (true or false)?

Svar:

0 – false

```
((25%2 == 0) || (10>=5-15)) && ((5<15%10) || ((1==1) && !(10!=25+15+5)))
(( 1 == 0) || (10>= -10)) && ((5< 5 ) || (( 1 ) && !(10!= 45 )))
( false || true ) && ( false || ( true && !( true ) ) )
( false || true ) && ( false || ( true && false ) )
( true ) && ( false || false )
true && false
false
```

Question 4 [4p]

In the small program below a while-loop is used to create a loop behaviour. Rewrite the loop behaviour (the code sorunded by `/******LOOP*****/`) in two different ways. Once by using a for-loop and once by using a do-while-loop. You can assume that the user does'n enter negative numbers to start and decrementStep and that the input to decramentStep isn't 0. 0 is a possible input to start.

```
#include <stdio.h>
int main(void)
{
    int start, decrementStep, i;
    scanf("%d", &start);
    scanf("%d", &decrementStep);

    /******LOOP*****/
    i = start;

    while(i > 0)
    {
        //do something, tex. printf("%d ", i);
        i = i - decrementStep;
    }
    /******LOOP*****/
}
```

Answer:

For-loop: 0.5p per correct initiation, condition, change (total 1.5p)

```
-----
for(i = start; i > 0; i -= decrementStep)
{
    printf("%d ", i);
}
-----
```

do-while: 0.5p per correct initiation, condition, change
+ 1p for handling start being 0 in a correct way (total 2.5p)

```
-----
i = start;
do
{
    if(i == 0)
        break;
    printf("%d ", i);
    i-=decrementStep;
}while(i > 0);
-----
```

Question 5 [2p]

Assume that you have a binary file that you open with a program applicable for opening text files (for example notepad or TextEdit). The information in the file will be perceived as it is corrupted. Explain why (use no more than 10 sentences)

Answer:

The program you use to open the file is expecting a textfile, a file containing characters. The program will therefor translate the binary information in the file as characters using the ASCII-table.

An example: if we save the number 97 in the binary file, the binary representation of 97 (01100001)(64+32+1) will be saved to the file. When it is read with a texteditor, 01100001 will be translated using the ASCII-table to the character 'a'.

Question 6 [1p]

What is an algorithm? Explain using a maximum of 5 sentences.

Svar:

An algorithm is a description of a functionality or solution of a problem. Can be seen as a recepy. The algorithm describes in what order different things will have to be done to be able to create a desirable result.

Question 7 [4p]

Create the function `fillArray`.

The Array are to be filled with random integer numbers between `minLimit` and `maxLimit` (including these)

If `minLimit` is 5 and `maxLimit` is 10 then 5, 6, 7, 8, 9 and 10 are possible random numbers in the interval.

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define SIZE 10

int main(void)
{
    int arr[SIZE];
    int minLimit, maxLimit;
    printf("Enter limits for randomized integers: ");
    scanf("%d %d", &minLimit, &maxLimit);

    fillArray(arr, minLimit, maxLimit);

    return 0;
}
```

Answer:

Head of function – 1p
 Loop – 0.5p
 Srand (startvärde) – 1p
 Rand() – 0.5p
 Correct interval – 1p

```
void fillArray(int *arr, int min, int max)
{
    srand(time(0));
    int i;
    for(i = 0; i < SIZE; i++)
    {
        arr[i] = rand()%(max+1-min)+min;
    }
}
```

The logic of the randomization

min = 10
 max = 100

rand()%max —> 0 to 99
 rand()%max+min —> 0+10 to 99+10 —> 10 to 109
 rand()%(max-min) —> rand()%90 —> 0 to 89
 rand()%(max-min)+min —> 0+10 to 89+10 —> 10 to 99
 rand()%(max+1-min) —> rand()%91 —> 0 to 90
 rand()%(max+1-min)+min —> 0+10 to 90+10 —> 10 to 100

Question 8 [2p]

Explain (with words and potentiellay pictures) how you would (in a c program) find out how many sentence a string contains of. A sentence can end with a dot (.), question mark (?) or exclamation mark (!)

Answer:

Loop through the string element by element (character by character) and loop for '.', '?' and '!'. When one of these are found a variable counting the number of sentences (starting from zero) will be incremented (by 1). The loop will run until the end of the string ('\0').

A sentence that's not ended with . ? or ! does not count as a full sentence and are not counted.

Question 9 [3p]

Write a function that swaps the content of two integers. You also need to show what the function call looks like. The return type of the function should be void.

If the input to the program below is: 10 and 20, the output should be:

Before funk: 10 and 20

After funk: 20 and 10

```
#include <stdio.h>

int main(void)
{
    int number1, number2;
    printf("Enter two numbers: ");
    scanf("%d %d", &number1, &number2);

    printf("Before funk: %d and %d", number1, number2);
    //function call
    printf("After funk: %d and %d", number1, number2);

    return 0;
}
```

Lösningsförslag:

Head of function – 1p
Swap of numbers in the variable – 1p
Function call – 1p

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

Call: swap(&number1, &number2);
```

Question 10 [3p]

When a C program is building, three things happens before an executable file is created, one of them is the compiler (which actually consists of two things). Explain all three thing together with what they do (their task).

Answer:

Preprocessor (1p) – Takes care of all # (includes och defines)

Compiler (1p) – Analyze and optimize the code, generates machine code (.o-fil)

Linker (1p) – links the.o-files together with the library files and create an executable (.exe)

Question 11 [6p]

In the program below there is a total of 12 errors, both syntax, semantics and logic. Some of the errors can be on the same row and some of the errors exists on multiple rows. If an error exists on multiple rows it still counts as one error.

For each of the errors you find – write down the row and how it should look to be correct. Each correct error found (row) and corrected gives 0.5p

*/*The program should allocate memory for 10 movies and then fill all the movies with information about title and production year */*

```

1      #include <stdio.h>
2      #include <stdlib.h>
3
4      struct movie
5      {
6          char title[STRSIZE];
7          int productionYear;
8      };
9
10     typedef struct
11     {
12         movie *list;
13         int numberOf;
14     }movieList;
15
16     int main(void)
17     {
18         movieList *myMovies;
19         myMovies->numberOf = 10;
20         myMovies = (movieList*)calloc(myMovies->numberOf, sizeof(movieList));
21
22         if(myMovies->list == NULL)
23         {
24             printf("Enter movie information\n-----\n");
25             addInformation(&myMovies);
26             free();
27         }
28         return 0;
29     }
30
31     void addInformation(movieList *myMovies)
32     {
33         for(i = 0; i <= myMovies->numberOf; i++)
34         {
35             printf("Title: ");
36             fgets(myMovies[i]->list.title, STRSIZE, stdin);
37             printf("ProductionYear: ");
38             scanf("%d", myMovies[i]->list.productionYear);
39         }
40     }

```

Answer:

- Row 6 & 36: #define STRSIZE 20 (0.5p)
- Row 12: **struct** movie (0,5p) || row 4: **typedef struct**{...}movie;
- Row 20: myMovies->list =
(**struct movie***)calloc(myMovies->numberOf, sizeof(**struct movie**));
(0.5p + 0.5p)
- Row 23: if(myMovies->list **!=** NULL) (0.5p)
- Row 25: addInformation(myMovies) (0.5p)
- Row 26: free(myMovies->list) (0.5p)
- Row 31: The function is not declared || should be places above main (0.5p)
- Row 33: for(**int** i = 0; i **<** myMovies->numberOf; i++) (0.5p + 0,5p)
- Row 36: 38: myMovies->list[i].title (0.5p)
- Row 38: scanf("%d", **&**(myMovies[i]->list.productionYear)); (0,5p)