

**FACIT!!**

## **Tentamen - Programmering**

**DVA117**

*Akademien för innovation, design och teknik*

*Torsdag 2015-08-20*

### **Preliminära betygsgränser**

Betyg 3 (52%): 12p

Betyg 4 (74%): 17p

Betyg 5 (91%): 21p

**Max: 23p**

**FACIT!!**

 <b>MÄLARDALENS HÖGSKOLA</b> <b>ESKILSTUNA VÄSTERÅS</b>	<b>Kod:</b> Code	<b>Kurskod:</b> Course code	<b>Bladnr:</b> Page nr
<b>Uppgift nr:</b> Task nr		<b>Kursnamn:</b> Course title	

Använd den här tabellen för att svara på delfrågorna i **Uppgift 1**. För varje delfråga (A-G) ska du markera det svarsalternativ som du tror är rätt svar med ett kryss (X) i motsvarande ruta.

Uppgift 1	Svar 1)	Svar 2)	Svar 3)	Svar 4)
A)				
B)				
C)				
D)				
E)				
F)				
G)				

**Observera!** Glöm inte att bifoga den här sidan tillsammans med övriga svarsblad när du lämnar in tentamen.

## Uppgift 1 [7p]

Varje delfråga är värd 1p. Ett av svarsalternativen för varje delfråga är rätt. Använd gärna den bifogade svarstabellen eller svara genom att ange bokstaven för delfrågan och den siffra som motsvarar det rätta svarsalternativet (Exempel: H2, K3, osv...).

A. Vad är sant för följande kod?

```
struct person{  
    char name[10];  
    int age;  
};
```

1. Det är en definition
2. Det är en deklaration och en initiering
3. Det är en definition och en initiering
4. Syntax Error

B. följande logiska uttryck om x=40, y=2, z=4, a=1, b=6?

```
res= x - a > b || z + a < b + x || z <= y;
```

5. res=0
6. res=1
7. res=5
8. Odefinierat

C. På ett 32 bitars system, vad är den mest sannolika storleken i *byte* för en `int`?

9. 2
10. 4
11. 16
12. 32

D. Vilken av följande kan inte användas i en switch-case-sats?

1. Character
2. Integer
3. Float
4. Enum

E. Vilken operator används för att få minnesadressen till en variabel?

1. \*
2. &
3. &&
4. ||

F. Vilken typ returnerar minnesallokeringsfunktionerna `malloc()` och `calloc()`

1. `void *`
2. Pekare av samma typ som typen av det allokerade minnet
3. `void **`
4. `int *`

G. Vilket påstående är rätt för följande program?

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

```
int main(){
    int i=10;
    int *j=&i;
    return 0;
}
```

1. j och i är pekare till integer
2. i är en pekare till en integer och sparar adressen för j
3. j är en pekare till en integer och sparar adressen för i
4. j är en pekare till en pekare till en integer och sparar adressen av i

## Uppgift 2 [6p]

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

```
int consonant(char ch){
    switch (ch) {
        case 'a': return 0;
        case 'e': return 0;
        case 'i': return 0;
        case 'o': return 0;
        case 'u': return 0;
        case 'y': return 0;
        case ' ': return 0;
        default : return 1;
    }
}
```

```
int func(FILE *fp, char ch) {
    fprintf(fp, "%c",ch);

    return 0;
}
```

```
int main()
{
    FILE *fp;
    char in[30];
    int i;

    if((fp = fopen("result.txt","w")) == NULL){
        printf("File could not be opened\n");
        return 0;
    }

    printf("Input: ");
    gets(in);

    for(i=0; i<30 && in[i]!='\0'; i++)
        if(consonant(*(in+i)))
            printf("%co%c ",in[i],*(in+i));
        else{
```

```

        printf("%c", in[i]);
        func(fp, in[i]);
    }

    fclose(fp);
    return 0;
}

```

- A. Beskriv vad funktionen `consonant` gör.

Svar (1p): Tar in ett tecken och returnerar 0 om det är en vokal (a,e,i,o,u,y) och 1 om det är en konsonant

- B. Beskriv vad funktionen `func` gör.

Svar (1p): Tar in ett tecken och skriver ut den till filen som filpekaren `fp` pekar mot

- C. Beskriv vad funktionen `main` gör.

Svar (2p): Öppnar en TEXT-fil med namnet `result.txt`

Frågar efter en textsträng

Loppar igenom textsträngen fram till `'\0'` eller position 29. Varje tecken i strängen kollas av i funktionen `consonant`. Är det en konsonant skrivs tecknet ut på skärmen två ggr med ett "o" i mitten + ett mellanslag. Är det inte en konsonant (utan en vokal) skrivs tecknet ut på skärmen utan mellanslag efteråt.

Varje vokal skickas även till funktionen `func`

- D. Vad kommer skrivas ut på skärmen när man skriver in texten `"examina-"`

Svar (1p): `exox amom inon a`

- E. Vad kommer samtidigt skrivas till filen?

Svar (1p): `eaia`

### Uppgift 3 [10p]

Följ specifikationerna i deluppgifterna nedan – en fungerande lösning som ej är uppbyggd enligt specifikationen ger poängavdrag.

- A. Skapa en datatyp `City` som håller information om en stad. Den information som ska finnas om staden är namn, antal invånare samt koordinater. För att lagra koordinaterna för staden så ska följande datatyp (`Coordinates`) användas.

```

typedef struct
{
    int longitude;
    int latitude;
}Coordinates;

```

SVAR:

```

typedef struct
{
    char name[20];
    int population;
    Coordinates coordinates;
}city;

```

- B. Skapa en dynamiskt allokerad array med plats för `SIZE` städer. Glöm inte felhantering. Du kan anta att `SIZE` är ett definierat makro.



SVAR:

```
city *cities = (city*)calloc(SIZE, sizeof(city));
if(cities == NULL)
{
    return;
}
```

- C. Skriv en funktion (`addCity`) vars uppgift är att fylla en stad med information (namn, antal invånare, longitud och latitud). Funktionen ska som enda parameter ta en pekare till den stad (det element i den dynamiska arrayen) som ska editeras. *Funktionen får inte ta hela den dynamiska arrayen som parameter.* Ingenting ska returneras.

Visa också hur koden som fyller hela den dynamiska arrayen (från uppgift b) med städer ser ut.

SVAR:

```
void addCity(city *cityToAdd)
{
    printf("\nName: ");
    scanf("%s%c", cityToAdd->name);
    printf("Population: ");
    scanf("%d", &cityToAdd->population);
    printf("coordinates: ");
    scanf("%d %d", &cityToAdd->coordinates.longitude,
              &cityToAdd->coordinates.latutude);
}
```

Anrop:

```
int i;
for(i = 0; i < SIZE; i++)
    addCity(&cities[i]);
```

- D. Skriv en funktion (`largestCity`) vars uppgift är att hitta den stad i arrayen som har flest invånare. Funktionen ska som parameter ta en pekare till arrayen av städer samt returnera antalet invånare i den största staden.

SVAR:

```
int largestCity(city *cities)
{
    int max = 0, i;
    for(i = 0; i < SIZE; i++)
        if(cities[i].population > max)
            max = cities[i].population;
    return max;
}
```