



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Lösningar till tentauppgifterna sätts ut på kurssidan på nätet idag kl 19.

Denna tenta kommer att vara färdigrättad On 14/1 -04 och kan då hämtas på mitt tjänsterum, T2221 mellan 13 och 15 då du också har möjlighet att diskutera rättningen med mig. Tentor som ej hämtats då, placeras på studentexpedition.

Omtentamen i Programmering C, 5p, fristående, kväll, 040110.

Hjälpmedel : Inga
Tid : 14-19
Ansvarig lärare : Gunnar Joki 303317, 274825(hem)

Svar till samtliga uppgifter 1-15 ska skrivas på utdelat extra papper. Använd ett papper till uppgifterna 1-5, två papper till uppgifterna 6-10 och ett papper per uppgift till uppgifterna 11-15. Skriv din tentamenskod på varje inlämnat extra papper.

Den maximala poängen för respektive uppgift står angiven efter uppgiftens nummer. Totalt kan 40 poäng erhållas. För betyget 3 krävs ca 20, för betyget 4 ca 28 och för betyget 5 ca 34 poäng.

Om inget speciellt anges gäller frågorna Borland C.

Detta häfte ska du behålla.

Lycka till!

1) (1p) Skriv den tilldelningssats i C som motsvarar formeln :

$$a = \frac{2}{b-c} + 3d^2$$

2) (1p) Vilka värden har följande uttryck?

- a) `7 % 6 / 5`
 b) `'A' > 'a' && 3 != 2`
-

3) (1p) Definiera en lämplig structvariabel, lampa, som avbildar en glödlampa med termerna effekt och spänning samt initiera variabeln med effekten 40.0 och spänningen 220.0.

4) (1p) Vad skrivs ut då nedanstående kod körs om a och b är heltal ?

```
a = 3;
b = 8;
for (k = a; k < b; k += 2)
{
    a++;
    b--;
}
printf("a = %d och b = %d", a, b);
```

5) (1p) Antag att du har en sträng str med max 10 tecken. Skriv de satser som byter plats mellan sista och första bokstaven i strängen dvs om str från början har värdet "hejsan" ska den efteråt ha värdet "nejsah". Inga färdiga funktioner får användas.

6) (2p) Skriv en funktion, `is_oktal_siffra`, som tar ett tecken som parameter och som returnerar sant (1) om tecknet är en oktal siffra alltså ligger mellan 0 eller 7, annars falskt (0).

7) (2p) Skriv en funktion, `is_oktaltal`, som tar en sträng som parameter och returnerar sant (1) om strängen är ett oktalt tal, alltså ett tal som består av siffror mellan 0 och 7, annars ska funktionen returnera falskt (0). Du ska anropa funktionen i uppgift 6 ovan.

8) (2p) Skriv en funktion, `lampstrom`, som tar en struct av typen enligt uppgift 3 ovan, som parameter och returnerar strömmen (effekten / spänningen) som lampan drar.

9) (2p) Skriv en funktion, `summa`, som tar antalet termer (nedan 5) i serien

$$1.0/1 + 1.0/2 + 1.0/4 + 1.0/8 + 1.0/16 \dots$$

som parameter och som returnerar seriens summa.

10)(2p) Skriv en funktion, normalisera, som tar en reell vektor och antalet element i vektorn, som parametrar och normaliserar vektorn, dvs. dividerar alla element i vektorn med det största elementvärdet. Elementen i den normaliserade vektorn ska alltså alla bli mindre än eller lika med 1.

11)(5p) Skriv ett fullständigt program som initierar en vektor med 5 valfria reella tal, normaliserar vektorn genom att anropa funktionen normalisera i uppgift 10 ovan och skriver ut den normaliserade vektorn.

12)(5p) Skriv ett fullständigt program som upprepat frågar efter en lampas effekt och spänning och skriver ut lampans ström. Lampans ström ska beräknas genom att funktionen lampstrom, enligt uppgift 8 ovan, anropas. Programmet ska avslutas direkt om en lampeffekt som är mindre än eller lika med 0.0 inmatas och skriva ett felmeddelande om spänningen 0.0 inmatas.

13)(5p) Skriv ett fullständigt program som läser in ett litet positivt reellt tal och som anropar funktionen summa i uppgift 9 ovan med ett ökande antal termer (1, 2, 3, ...), så länge skillnaden mellan en summa och närmast föregående summa är större än det inlästa lilla talet. Avslutningsvis skrivs de två sista summorna ut.

14)(5p) Skriv ett fullständigt program som från textfilen oktalatal.txt läser alla oktala tal och skriver ut dessa tals decimala värden om de är oktala tal enligt uppgift 7 ovan. Är det inte oktala tal skrivs felmeddelanden ut. Oktala tal har basen 8. Exempelvis är det oktala talet 234 lika med $2 \cdot 8^2 + 3 \cdot 8^1 + 4 \cdot 8^0 = 156$. Talen finns radvis i filen enligt:

```
12
23
234
..
```

15)(5p) Skriv ett fullständigt program, som simulerar kortspelet 21 genom att programmet ger dig ett spelkort åt gången så länge du vill ha ett kort. Summan av kortens värden ska komma så nära 21 som möjligt. Får du exakt 21 har du vunnit direkt. Är summan av dina kort mer än 21 har du förlorat direkt. Har du kort vars summa är mindre än 21 ska programmet ge kort åt din motspelare och den av er som kommer närmast 21 har då vunnit. Låt kortleken vara en vektor av 52 kort med värde och färg enligt:

```
struct kort
{
    int value;      /* 1-13, äss räknas som 1 */
    int color;     /* 1-4 */
};
```

Du ska skriva egna funktioner för att skapa kortleken och för att blanda enligt:

```
void skapa(struct kort kortlek[]);
void blanda(struct kort kortlek[]);
```

Lösningar till tentamen i Programmering , 040110.

- 1) `a = 2/(b - c) + 3*pow(d, 2);`
- 2) a) 0 b) 0
- 3)

```
struct lamptyp
{
    float effekt, spanning;
} lampa = {40.0, 220.0};
```
- 4) `a = 5 och b = 6;`
- 5)

```
char str[10];
char temp;
int k;

for (k = 0; str[k] != '\0'; k++)
{
}

if (k > 0)
{
    temp = str[0];
    str[0] = str[k-1];
    str[k-1] = temp;
}
```
- 6) Se uppgift 14
- 7) Se uppgift 14
- 8) Se uppgift 12
- 9) Se uppgift 13
- 10) Se uppgift 11
- 11)

```
void normalisera(float v[], int nr)
{
    int i;
    float max = v[0];

    for (i = 1; i < nr; i++)
    {
        if (v[i] > max)
        {
            max = v[i];
        }
    }
    for (i = 0; i < nr; i++)
    {
        v[i] /= max;
    }
}
```

```

#include <stdio.h>
#include <conio.h>

void main()
{
    float vek[5] = {2.3, 4.5, 2.8, 5.4, 1.3};
    int i;

    normalisera(vek, 5);
    for (i = 0; i < 5; i++)
    {
        printf("%.1f ",vek[i]);
    }
    getch();
}

```

12)

```

struct lamptyp
{
    float effekt, spanning;
};

float lampstrom(struct lamptyp lampa)
{
    return lampa.effekt / lampa.spanning;
}

#include <stdio.h>

void main()
{
    struct lamptyp lampa;

    printf("Ge lampans effekt (avsluta med effekt <= 0.0 ) : ");
    scanf("%f", &lampa.effekt);
    while (lampa.effekt > 0.0)
    {
        printf("Ge lampans spänning : ");
        scanf("%f", &lampa.spanning);
        if (lampa.spanning != 0.0)
        {
            printf("Lampans ström = %.2f\n", lampstrom(lampa));
        }
        else
        {
            printf("Spänningen får ej vara 0.0 !\n");
        }
        printf("Ge lampans effekt (avslut med effekt <= 0.0 ) : ");
        scanf("%f", &lampa.effekt);
    }
}

```

```

13) #include <stdio.h>
#include <math.h>
#include <conio.h>

float summa(int nr)
{
    int i;
    float sum = 0.0;

    for (i = 0; i <= nr; i++)
    {
        sum += 1.0 / pow(2,i);
    }
    return sum;
}

void main()
{
    int i = 0;
    float forsta, andra, diff;

    printf("Ge ett litet reellt tal : ");
    scanf("%f", &diff);
    do
    {
        forsta = summa(i);
        andra = summa(i+1);
        i++;
    } while (diff < andra - forsta);
    printf("%f %f\n", forsta, andra);
    getch();
}

14) int is_oktalsiffra(char siffra)
{
    return (siffra >= '0' && siffra <= '7');
}

int is_oktaltal(char *oktalstr)
{
    int i;

    for (i = 0; oktalstr[i] != '\0'; i++)
    {
        if (! is_oktalsiffra(oktalstr[i]))
            return 0;
    }
    return 1;
}

void main()
{
    char oktaltal[20];
    int i, sum;
    FILE *tsin;

    tsin = fopen("Oktalatal.txt","rt");
    while(fscanf(tsin, "%s", oktaltal) != EOF)
    {

```

```

    if (!is_oktaltal(oktaltal))
    {
        printf("Fel indata!\n");
    }
    else
    {
        i = 0;
        sum = 0;
        while(oktaltal[i] != '\0')
        {
            sum = sum*8 + oktaltal[i] - '0';
            i++;
        }
        printf("Talet decimalt : %d\n", sum);
    }
}
getch();
}

```

15)

```

#include <conio.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

struct kort
{
    int value;
    int color;
};

void skapa(struct kort kortlek[])
{
    int i;

    for (i = 0; i < 52; i++)
    {
        kortlek[i].value = i % 13 + 1;
        kortlek[i].color = i % 4 + 1;
    }
}

void blanda(struct kort kortlek[])
{
    int varv, i, k;
    struct kort temp;

    srand((unsigned)time(NULL));
    for (varv = 1; varv <= 1000; varv++)
    {
        i = rand() % 52;
        k = rand() % 52;
        temp = kortlek[i];
        kortlek[i] = kortlek[k];
        kortlek[k] = temp;
    }
}

void main()
{
    struct kort minlek[52], kort;
    int i, asum=0, bsum = 0, kortnr;

```

```
char svar[20];

skapa(minlek);
blanda(minlek);
kortnr = 51;
printf("Spelare A spelar!\n");
do
{
    asum += minlek[kortnr].value;
    printf("Kortsumma = %d\n", asum);
    kortnr--;
    if (asum < 21)
    {
        printf("Fler kort? ja/nej");
        gets(svar);
    }
} while (asum < 21 && strcmp(svar, "ja") == 0);

if (asum > 21)
{
    printf("Spelare B vinner!\n");
}
else if (asum == 21)
{
    printf("Spelare A vinner!\n");
}
else
{
    printf("Spelare B spelar!\n");
    do
    {
        bsum += minlek[kortnr].value;
        printf("Kortsumma = %d\n", bsum);
        kortnr--;
        if (bsum < 21)
        {
            printf("Fler kort? ja/nej");
            gets(svar);
        }
    } while (bsum < 21 && strcmp(svar, "ja") == 0);

    if (bsum > asum && bsum <= 21)
    {
        printf("Spelare B vinner!\n");
    }
    else
    {
        printf("Spelare A vinner!\n");
    }
}
getch();
}
```