



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

**Lämna in ifylld kursvärdering tillsammans med tentamen!**

**Lösningarna till tentamensuppgifterna sätts ut på kurssidan på nätet under eftermiddagen.**

**Tentamen i Programmeringsmetodik, 7,5 hp (DT2006, DT2010), 2007-11-10.**

Hjälpmedel : Inga  
Tid : 08:00-11:00  
Ansvarig lärare : Christer Lindkvist 303393, 070-3273393

Svar till samtliga uppgifter 1-9 ska skrivas på utdelat extra papper. Använd ett papper till uppgifterna 1-3, två papper till uppgifterna 4-6 och ett papper per uppgift till uppgifterna 7-9. Skriv din tentamenskod på varje inlämnat extra papper.

Den maximala poängen för respektive uppgift står angiven i högermarginalen. Totalt kan 24 poäng erhållas. För betyget 3 krävs ca 12, för betyget 4 ca 17 och för betyget 5 ca 21,5 poäng.

Om inget speciellt anges gäller frågorna Visual C++.

Detta häfte ska du behålla.

Lycka till!

- 
- 1) Antag att du har variablerna

```
char *cp, c = '5';
```

Visa hur man sätter **cp** att peka på **c** och hur man med **cp** ändrar värdet på **c** till '3'. (1p)

---

- 2) Antag att du har en 8 bitars unsigned char definierad enligt:

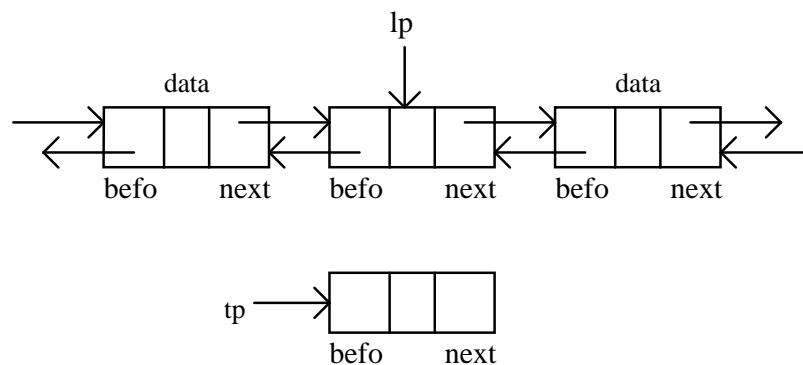
```
unsigned char uch = 0x5a;
```

Ange värdet decimalt av **uch** efter satsen: (1p)

```
uch = uch & (15<<4);
```

---

- 3) Ange hur du med hjälp av funktionerna i Twolist.h (se uppgift 9), byter ut länken som **lp** pekar på mot länken som **tp** pekar på i nedanstående tvåvägslista. När du är färdig ska **lp** peka på den nya länken i listan och **tp** peka på den borttagna länken. Inga andra pekare än **lp** och **tp** får användas. (1p)



- 
- 4) Ett **Fibonaccital** ingår i en sekvens av heltal, där varje tal är summan av de två föregående. De två första talen är 0 och 1. Matematiskt innebär det att Fibonaccitalen är en sekvens  $F(n)$ , definierad rekursivt enligt:

$$F(n) = \begin{cases} 0 & \text{om } n = 0 \\ 1 & \text{om } n = 1 \\ F(n-1) + F(n-2) & \text{om } n > 1 \end{cases}$$

Skriv en rekursiv funktion som returnerar Fibonaccital enl. ovan. (2p)

---

- 5) En medlem i en förening kan avbildas som en abstrakt datatyp enligt:

```

/* Specifikation av medlem -- medlem.h */

#ifndef MEDLEM_H
#define MEDLEM_H

typedef struct {
    int mnr;          /* Medlemsnummer */
    char namn[30];   /* Medlemmens namn */
    float avgift;    /* Medlemsavgift */
    int vip;         /* Vipmedlem eller inte */
} medlem;

void las_medlem(memlem *mp, float vipavgift);
/* Läs mnr, namn och avgift med ledtexter från
   tangentbordet. Sätt vip till 1 om avgift större
   än eller lika med vipavgift, annars 0 */

void visa_medlem(memlem m);
/* Visa data för medlem på skärmen */

float avgift_medlem(memlem m);
/* Returnerar medlemsavgiften */

int vip_medlem(memlem m);
/* Returnerar 1 om m är vipmedlem, annars 0 */

int har_lagre_mnr(memlem m1, memlem m2);
/* Returnerar 1 om m1 har lägre medlemsnummer än m2, annars 0 */

#endif

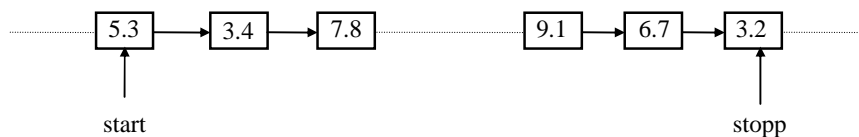
```

Implementera funktionerna **las\_medlem** enl. ovan. Namnet kan innehålla flera ord. För att rensa bufferten till och med nästa radslut kan funktionen **clr\_line** användas:

(2p)

```
void clr_line(void) // Rensar resten av raden
```

- 6) Antag att du har en envägslista, där länkarna är av typen **linktyp** och innehåller **data** i form av reella tal och en **next**-pekare enligt:



Fullborda funktionen **maxpek** nedan, som tar **start** och **stopp** som parametrar och som returnerar en pekare till den länk (från och med start och till och med stopp) som har det största **data**-värdet..

(2p)

```
linktyp *maxpek(linktyp *start, linktyp *stopp)
```

- 
- 7) Skriv ett program som dynamiskt allokerar en vektor som ska innehålla ett inläst antal medlemmar enligt uppgift 5 ovan. Läs först in antalet medlemmar vektorn ska innehålla. Läs därefter in medlemmarna till vektorn. Avgiftsgränsen för vip-medlemmar ska vara 500.00 kr. Skriv slutligen ut medelvärdet av medlemsavgifterna samt de medlemmar som betalt en lägre avgift än medelvärdet. Använd funktionen **avgift\_medlem** vid summering och utskriftstest. (5p)
- 

- 8) Skriv ett program som upprepat (avslutas med bara RETURN) läser in ett hexadecimalt tal som en sträng och stoppar in de hexadecimala siffrorna på en stack. När hela strängen är stackad plockas tecknen bort från stacken samtidigt som talets decimala värde summeras. Avslutningsvis skrivs talet ut decimalt. Omvandlingen av en hexadecimal siffra till dess decimala värde ska ske i en funktion. För hantering av stacken ska du använda: (5p)

```

/* Specifikation av LIFO-lista -- lifo.h */

#ifndef LIFOH
#define LIFOH

typedef char datatyp;      /* Exempelvis */

typedef struct link {
    datatyp data;
    struct link *next;
} linktyp;

void push(linktyp **lpp, datatyp d);
/* Stoppar in d i LIFO-listan */

datatyp pop(linktyp **lpp);
/* Tar bort data från LIFO-listan */

#endif

```

---

- 9) I binärfilen **Medlemmar.dat** finns en förenings medlemmar. Skriv ett program som läser hela filen och stoppar in medlemmarna i två olika tvåvägslistor, en för vip-medlemmar och en för övriga. Listorna ska vara sorterade efter medlemsnummer. Avslutningsvis ska programmet skriva ut innehållet i de två listorna på skärmen. För hantering av medlemmar ska du använda den abstrakta datatypen **medlem** i uppgift 5 och för hantering av tvåvägslistorna ska du använda den abstrakta datatypen **twolist** nedan. (5p)

```

/* Specifikation av tvåvägslista -- twolist.h */

#ifndef TWOLISTH
#define TWOLISTH

#include "medlem.h"
typedef medlem datatype;

typedef struct twolink {
    enum {head, link} kind;
    struct twolink *befo, *next;
    datatype data;
} headtyp, linktyp;

void newhead(headtyp **hpp);
/* Skapar en ny tom lista */

void newlink(linktyp **lpp);
/* Skapar en ny tom länk */

void putlink(datatype d, linktyp *lp);
/* Sätter in data i en länk */

datatype getlink(linktyp *lp);
/* Returnerar data från länk */

void inlast(linktyp *lp, headtyp *hp);
/* Sätter in länken sist i listan */

void infirst(linktyp *lp, headtyp *hp);
/* Sätter in länken först i listan */

void inpred(linktyp *lp, linktyp *ep);
/* Sätter in första länken före den andra */

void insucc(linktyp *lp, linktyp *ep);
/* Sätter in första länken efter den andra */

void insort(linktyp *lp, headtyp *hp,
            int (*is_less)(datatype d1, datatype d2));
/* Sätter in länken sorterad enligt is_less */

linktyp *firstlink(headtyp *hp);
/* Returnerar pekare till första länken i listan */

linktyp *lastlink(headtyp *hp);
/* Returnerar pekare till sista länken i listan */

linktyp *predlink(linktyp *lp);
/* Returnerar pekare till länken före */

linktyp *succlink(linktyp *lp);
/* Returnerar pekare till länken efter */

int is_link(linktyp *lp);
/* Returnerar 1 om länk annars 0 */

int empty(headtyp *hp);
/* Returnerar 1 om listan tom annars 0 */

int nrlinks(headtyp *hp);
/* Returnerar antalet länkar i listan */

```

```
void outlist(linktyp *lp);  
/* Tar bort länken från listan */  
  
void elimlink(linktyp **lpp);  
/* Tar bort, avallokerar och NULL-ställer länken */  
  
void clearhead(headtyp *hp);  
/* Tar bort alla länkar från listan */  
  
void elimhead(headtyp **hpp);  
/* Elimineras och NULL-ställer listan */  
  
#endif
```

---

## Lösningar till tentamen i Programmeringsmetodik, 5p 2007-11-10

- 1) 

```
cp = &c;
*cp = '3';
```
- 2) 
$$\frac{01011010 \text{ \& } 11110000}{01010000} = 0x50 = 5 \cdot 16 = 80$$
- 3) 

```
insucc(tp, lp);
lp = tp;
tp = predlink(tp);
outlist(tp);
```
- 4) 

```
unsigned int F(unsigned int n)
{
    if (n == 0)
        return 0;
    else if (n == 1)
        return 1;
    else
        return F(n-1) + F(n-2);
}
```
- 5) 

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

void las_medlem(medlem *mp, float vipavgift)
{
    printf("\nGe medlemsnummer: ");
    scanf("%d", &mp->mnr); clr_line();
    printf("Ge medlemmens namn: ");
    gets(mp->namn);
    printf("Ge medlemsavgift: ");
    scanf("%f", &mp->avgift); clr_line();
    mp->vip = mp->avgift >= vipavgift;
}
```
- 6) 

```
linktyp *maxpek(linktyp *start, linktyp *stopp)
{
    linktyp *temp = start, *max = stopp;

    while (temp != stopp) {
        if (temp->data > max->data)
            max = temp;
        temp = temp->next;
    }
    return max;
}
```

```

7) #include <stdio.h>
#include <stdlib.h>
#include "medlem.h"

void main()
{
    int i, nr;
    float sum = 0.0, medel;
    medlem *mp;

    printf("Ge antal medlemmar: ");
    scanf("%d", &nr);
    mp = calloc(nr, sizeof(memlem));
    for (i = 0; i < nr; i++) {
        las_medlem(&mp[i], 500.00);
        sum += avgift_medlem(mp[i]);
    }

    medel = sum/nr;
    printf("\nMedel = %.2f\n", medel);

    for (i = 0; i < nr; i++)
        if (avgift_medlem(mp[i]) < medel)
            visa_medlem(mp[i]);

    free(mp);
}

```

```

8) #include <stdio.h>
#include "lifo.h"

int value(char hex)
{
    if (hex >= '0' && hex <= '9')
        return hex - '0';
    if (hex >= 'A' && hex <= 'F')
        return hex + 10 - 'A';
    if (hex >= 'a' && hex <= 'f')
        return hex + 10 - 'a';
    return -1;
}

void main()
{
    linktyp *lp = NULL;
    char bch, hexstr[6];
    int i, fak, sum;

    /* Läs in en hexsträng */
    printf("Hexstal? ");
    gets(hexstr);

    while (hexstr[0] != '\0') {
        /* Stacka tecknen i hexsträngen */
        for (i = 0; hexstr[i] != '\0'; i++) {
            push(&lp, hexstr[i]);
        }

        /* Plocka ut tecknen och summera värdet */
        sum = 0;
        fak = 1;
    }
}

```



```

while (lp != NULL) {
    bch = pop(&lp);
    sum += value(bch) * fak;
    fak *= 16;
}

/* Skriv resultat */
printf("Värde : %d\n", sum);

/* Läs in en hexsträng */
printf("Hexstal? ");
gets(hexstr);
}
}

```

```

9) #include <stdio.h>
#include "twolist.h"
#include "medlem.h"

void main()
{
    FILE *bsin;
    headtyp *lista, *viplista;
    linktyp *link;
    medlem m;

    newhead(&lista);
    newhead(&viplista);

    /* Läs data från fil och stoppa in i rätt lista */
    bsin = fopen("Medlemmar.dat", "rb");
    fread(&m, sizeof(medlem), 1, bsin);
    while (!feof(bsin)) {
        newlink(&link);
        putlink(m, link);
        if (vip_medlem(m))
            insort(link, viplista, har_lagre_mnr);
        else
            insort(link, lista, har_lagre_mnr);
        fread(&m, sizeof(medlem), 1, bsin);
    }
    fclose(bsin);

    /* Skriv ut resp. lista */
    printf("Viplista\n");
    for (link = firstlink(viplista); link; link = succlink(link))
        visa_medlem(getlink(link));

    printf("\n\nLista\n");
    for (link = firstlink(lista); link; link = succlink(link))
        visa_medlem(getlink(link));
}

```