



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Namn:..... Klass:..... Pnr:.....

Omtentamen i Programmeringsmetodik, 5p, D1 och E1, 010110

Hjälpmedel : Inga

Tid : 8-13

Ansvarig lärare : Gunnar Joki

Tel arb: 303317

Tel hem: 274825

Svaren till uppgifterna 1-15 ska skrivas på tillgängligt utrymme i detta häfte. Behöver du mera utrymme kan du skriva på baksidan eller på extra papper. Lösningarna till uppgifterna 16-19 ska skrivas på utdelat extra papper med maximalt en uppgift per papper. Skriv namn och personnummer på varje inlämnat extra papper.

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

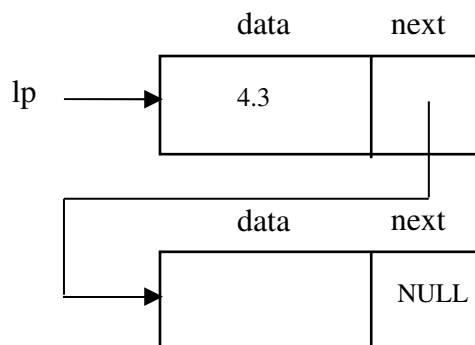
Om inget speciellt anges gäller frågorna Borland C .

Detta häfte inlämnas.

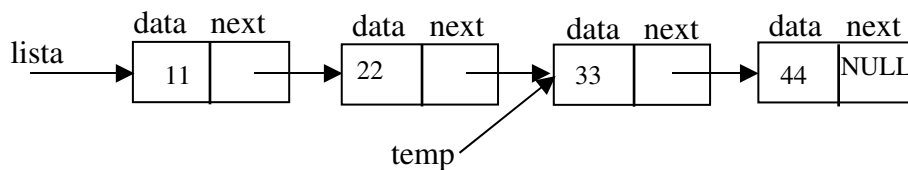
Lycka till!

1) (1p) Antag att du har en pekare `hp` som pekar på ett minnesutrymme som innehåller ett heltal. Skriv den `printf`-sats som skriver ut detta heltal på skärmen.

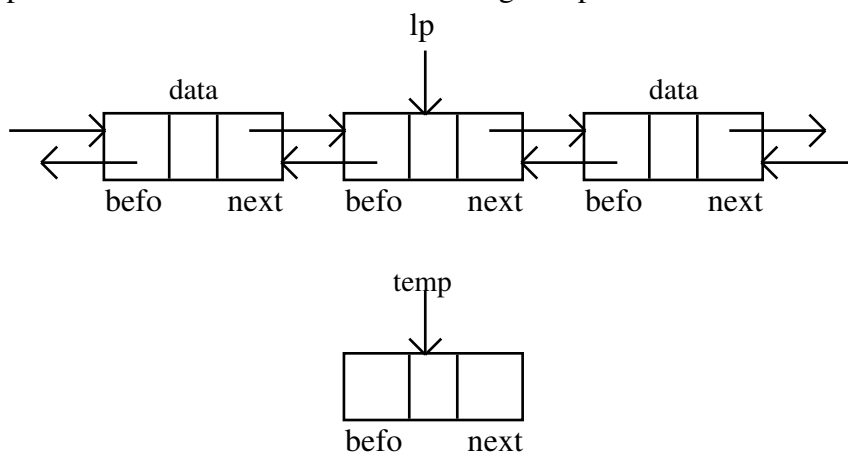
2)(1p) Skriv den `scanf`-sats som från tangentbordet läser in ett reellt tal till datatermen för den länk som saknar data.



3) (1p) Skriv de satser som tar bort och avallokerar den sista länken (länken längst till höger). Inga extrapekare får definieras och den nya listan ska avslutas med `NULL`.



4) (1p) Skriv de satser som krävs för att sätta in länken som `temp` pekar på efter (till höger om) länken som `lp` pekar på i nedanstående tvåvägslista. Inga färdiga rutiner eller extra pekare får användas och listan ska hänga ihop efteråt.



5) (1p) Rita det binära träd som skapas då strängen "ABBA" instoppas bokstav för bokstav i trädet i den angivna ordningen, om instoppningsfunktionen sätter in data som är mindre i vänsterträdet och lika stora eller större i högerträdet.

6)(1p) Som hashfunktion för strängar kan man exempelvis ta summan av ASCII-koderna för strängens tecken % 97. Skriv koden som summerar ASCII-koderna för alla tecken i strängvariabeln `strang`.

7)(1p) På vilket index hamnar strängen "ABBA" i en tom hashtabell, om man använder en hashfunktion enligt uppgift 6 ovan? Bokstaven 'A' har ASCII-koden 65.

8)(1p) Vad blir utskriften från följande program?

```
#include <stdio.h>
void rf(int n)
{
    if (n > 0)
    {
        rf(n-1);
    }
    printf("%d\n", n);
}

void main()
{
    rf(2);
}
```

9)(1p) Antag att du har en 8 bitars unsigned char definierad enligt:

```
unsigned char uch = 3;
```

Ange värdet för uch efter satsen:

```
uch ^= 2;
```

10)(1p) Visa hur du kommer åt bokstaven 'å' i följande vektor av strängar.

```
char *str[] = {"Hej", "på" , "dig"};
```

11)(2p)En medlem i en förening kan avbildas som en abstrakt datatyp enligt:

```
/* Specifikation av medlem -- 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 */
```

Implementera funktionen `visa_medlem` så att utskriften exempelvis blir:

```
Mnr : 123
Namn : Eva Karlsson
Avgift : 200.00
Vip : Nej
```

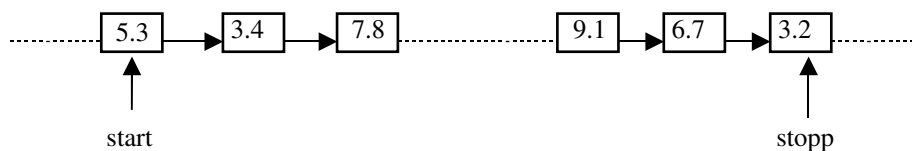
12)(2p)Implementera funktionen `las_medlem` i uppgift 11 ovan.

13)(2p) Fullborda funktionen `enkomp`, som ska returnera enkomplementet för `uch`.

Enkomplementet fås genom att man byter alla bitar som är 1 till 0 och alla bitar som är 0 till 1. Exempelvis om `uch` är 00001000 ska funktionen returnera 11110111 och är `uch` 00100001 ska funktionen returnera 11011110 osv.

```
unsigned char enkomp(unsigned char uch)
{
```

14)(2p) Antag att du har en envägslista, där länkarna är av samma typ som i uppgift 2 ovan med data i form av reella tal, 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.

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

15)(2p) Skriv en rekursiv funktion som beräknar längden av en sträng. Inga färdiga funktioner får användas. Funktionshuvud enligt:

```
int strlangd(char *str)
{
```

16)(5p)Skriv ett program som upprepat (avslutas med 0) läser in avgiftsgränsen för att bli vipmedlem och dynamiskt allokerar en variabel av typen medlem, enligt uppgift 11 ovan. Programmet ska sedan läsa in värden till medlemmen och visa medlemmen på skärmen. Glöm ej att frigöra minne inför nästa inläsning.

17)(5p)Skriv ett program som frågar efter antalet medlemmar, enligt uppgift 11 ovan, som ska läsas in och som skriver dessa till en textfil, vars namn inläses. Avgiftsgränsen för att bli vipmedlem ska vara 1000 kr. Textfilen ska innehålla medlemsnummer, namn, medlemsavgift och vipvärde radvis enligt exempelvis:

```
1
a a
200.00
0

2
b b
1200.00
1

3
c c
500.00
0

4
d d
1500.00
1

5
e e
700.00
0
```

18)(5p)Skriv ett program som upprepat (avslutas med bara RETURN) läser in ett hexadecimalt tal som en sträng, 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:

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

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 */

```

- 19)(5p) I en textfil enligt uppgift 17 ovan finns en förenings medlemmar. Skriv ett program som läser hela filen och stoppar in medlemmarna i två olika tvåvägslistor, en för vipmedlemmar och en för övriga. Avslutningsvis ska programmet skriva ut innehållet i de två listorna på skärmen. För hantering av tvåvägslistan ska du använda:

```

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

#include "medlem.h"
typedef medlem datatyp;

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

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

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

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

datatyp 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)(datatyp d1, datatyp 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);
/* Eliminerar och NULL-ställer listan */
```


Lösningar till omtentamen i Programmeringsmetodik, 5p, D1 och E1, 010110

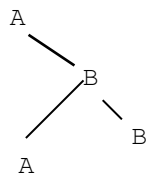
1) `printf("%d", *hp);`

2) `scanf("%f", &lp->next->data);`

3) `free(temp->next);`
`temp->next = NULL;`

4) `temp->befo = lp;`
`temp->next = lp->next;`
`lp->next->befo = temp;`
`lp->next = temp;`

5)



6) `int i, sum = 0;`
`for (i = 0; strang[i] != '\0'; i++)`
`sum += strang[i];`

7) $(65 + 66 + 66 + 65) \% 97 = 262 \% 97 = 68$

8) 0
1
2

9) 1

10) `str[1][1];`

- 11) `#include "medlem.h"`
`#include <stdio.h>`
- ```
void visa_medlem(medlem m)
{
 printf("Mnr : %d\n", m.mnr);
 printf("Namn : %s\n", m.namn);
 printf("Avgift : %.2f\n", m.avgift);
 printf("Vip : ");
 if (m.vip)
 printf("Ja\n");
 else
 printf("Nej\n");
}
```
- 12) `#include "medlem.h"`  
`#include <stdio.h>`
- ```
void las_medlem(medlem *mp, float vipavgift)
{
    printf("Mnr? ");
    scanf("%d", &mp->mnr);
    getchar();
    printf("Namn? ");
    gets(mp->namn);
    printf("Avgift? ");
    scanf("%f", &mp->avgift);
    if (mp->avgift >= vipavgift)
        mp->vip = 1;
    else
        mp->vip = 0;
}
```
- 13) `unsigned char enkomp(unsigned char uch)`
- ```
{
 int i;

 for (i = 0; i <= 7; i++)
 {
 uch = uch ^ (1 << i);
 }
 return uch;
}
```
- 14) `linktyp *maxpek(linktyp *start, linktyp *stopp)`
- ```
{
    linktyp *temp, *max;

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

- 15)

```
int strlangd(char *str)
{
    if (*str == '\0')
        return 0;
    return 1 + strlangd(++str);
}
```
- 16)

```
/* Huvudprogram -- medmain.c */

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

void main()
{
    medlem *mp = NULL;
    float vipavgift;

    /* Läs vipavgift */
    printf("Vipavgift? ");
    scanf("%f", &vipavgift);
    while (vipavgift != 0.0)
    {
        /* Skapa en medlem */
        mp = malloc(sizeof(memlem));
        las_medlem(mp, vipavgift);

        /* Visa medlem */
        visa_medlem(*mp);

        /* Frigör minne */
        free(mp);
        mp = NULL;

        /* Läs vipavgift */
        printf("Vipavgift? ");
        scanf("%f", &vipavgift);
    }
}
```
- 17)

```
/* Huvudprogram -- forening.c */

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

void main()
{
    FILE *tsut;
    char fnamn[20];
    medlem m;
    const float vipavg = 1000;
    int nr, med;

    /* Öppna fil för utskrift */
    printf("Filnamn? ");
    gets(fnamn);
```

```

tsut = fopen(fnamn, "w");

/* Läs medlemmar och skriv på fil */
printf("Hur många medlemmar ska läsas? ");
scanf("%d", &nr);
for (med = 1; med <= nr; med++)
{
    las_medlem(&m, vipavg);
    fprintf(tsut, "%d\n%s\n%.2f\n%d\n\n", m.mnr, m.namn, m.avgift,
                                                    m.vip);
}
fclose(tsut);
}

```

18)

```

/* Hextal.c */

#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("Hextal? ");
    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);
    }
}

```

19)

```

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

/* Medtwo.c -- huvudprogram */

#include "Twolist.h"
#include <stdio.h>
#include <string.h>
void main()
{
    FILE *tsin;
    headtyp *lista, *viplista;
    linktyp *link;
    medlem m;
    char filnamn[20];

    /* Läs data från fil och stoppa in i rätt lista */
    printf("Filnamn? ");
    gets(filnamn);
    tsin = fopen(filnamn, "r");
    newhead(&lista);
    newhead(&viplista);
    while (fscanf(tsin, "%d", &m.mnr) != EOF)
    {
        fgetc(tsin);      /* Läs bort radslut */
        fgets(m.namn, sizeof(m.namn), tsin);
        m.namn[strlen(m.namn)-1] = '\0';    /* Stryk radslut */
        fscanf(tsin, "%f%d", &m.avgift, &m.vip);
        newlink(&link);
        putlink(m, link);
        if (m.vip)
            inlast(link, viplista);
        else
            inlast(link, lista);
    }
    fclose(tsin);

    /* Skriv ut resp. lista */

    printf("Viplista\n");
    link = firstlink(viplista);
    while (link != NULL)
    {
        visa_medlem(getlink(link));
        link = succlink(link);
    }

    printf("\n\nLista\n");
    link = firstlink(lista);
    while (link != NULL)
    {
        visa_medlem(getlink(link));
        link = succlink(link);
    }
}

```