ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Lämna ifylld kursvärdering till tentavakten samtidigt som du lämnar in tentan.

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

Denna tenta kommer att vara färdigrättad Må 7/6 -04 och kan då hämtas på mitt tjänsterum T2221 mellan 13 och 15. Vid detta tillfälle har du också möjlighet att diskutera rättningen. Tentor som ej hämtats då placeras på studentexpedition.

Tentamen i Programmeringsmetodik, 5p, Au2, D1, E1, Pr1, 040601.

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!

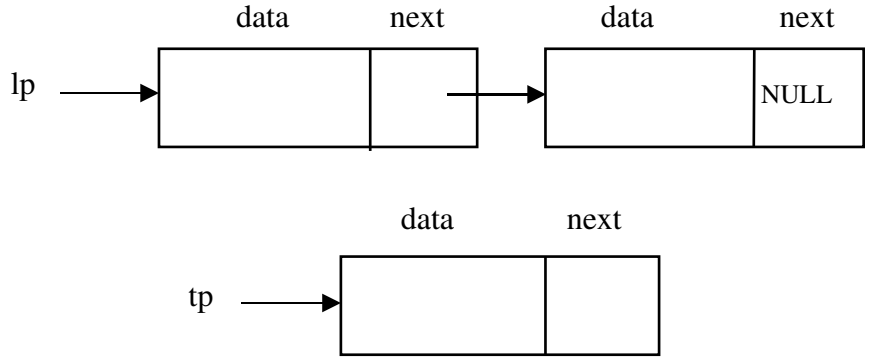
1) (1p) Anta att du har variablerna

```
float *fp, f = 2.3;
```

Visa hur man sätter fp att peka på f och hur man med fp ändrar värdet på f till 4.5.

---

2)(1p) Stoppa in länken tp först (till vänster om länken lp) i den länkade strukturen nedan. Inga extra pekare får definieras och både lp och tp ska peka på den nya första länken.



---

3) (1p) Antag att du har en 8 bitars unsigned char definierad enligt:

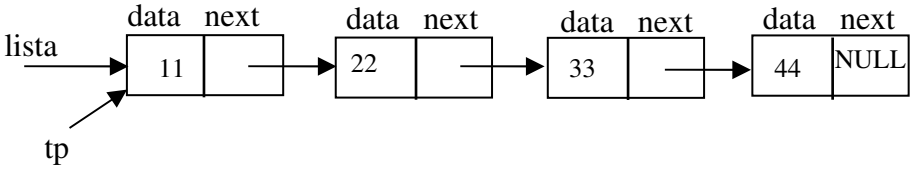
```
unsigned char uch = 0xc;
```

Ange värdet decimalt av uch efter satsen:

```
uch &= 4;
```

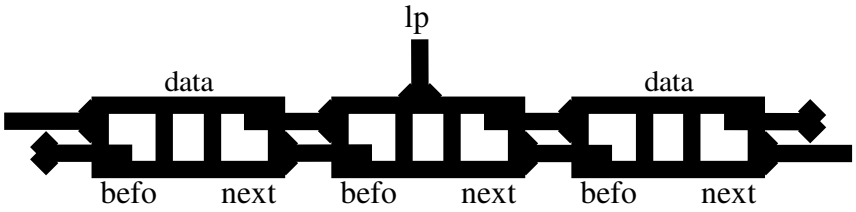
---

4) (1p) Skriv de satser som börjar med att leta upp den länk som har värdet 33 i nedanstående länkade struktur och som sedan ökar denna länks datavärde med 5.



---

5)(1p) Ange hur du, med hjälp av funktionerna i Twolist.h (se uppgift 15), tar bort och avallokerar länken före (till vänster om) lp i nedanstående tvåvägslista.



6)(2p) En rät linje i ett rätvinkligt xy-koordinatsystem kan avbildas som en abstrakt datatyp med riktningskoefficient och skärning med y-axeln enligt:

```
/* Linje.h */

typedef
struct
{
   float k;       /* Riktningskoefficient */
   float m;       /* Skärning med y-axeln */
} linje;

void las_linje(linje *lp);
/* Läser in k och m med ledtexter*/

void skriv_linje(linje l);
/* Skriver ut linjens ekvation med aktuella värden på k och m insatta
enligt: Linjen y = k*x + m */

float x_linje(linje l);
/* Returnerar skärning med x-axeln som är -m/k */

int parallell_linje(linje l1, linje l2);
/* Returnerar sant om samma riktningskoefficient för l1 och l2 */
```

Implementera funktionen las_linje.

---

7)(2p) Implementera funktionen skriv_linje, i uppgift 6 ovan.

---

8)(2p) Skriv ett huvudprogram som skapar två linjer, enligt uppgift 6 ovan, läser in värden till dessa och skriver ut linjernas skärning med x-axeln exempelvis som:

```
Linjen y = 5.0*x + 9.0 skär x-axeln i x = -1.8
Linjen y = 5.0*x - 2.0 skär x-axeln i x = 0.4
```

Använd funktioner för den abstrakta datatypen linje ovan för de understrukna delarna.

---

9)(2p) Skriv en funktion, set_bit, som tar en pekare till ett heltal och ett bitnummer (0 – 31) som parametrar och som sätter den angivna biten i detta heltal . Funktionsprototyp enligt:

```
void set_bit(int *talpek, int bit_nr);
```

---

10)(2p)Skriv en rekursiv funktion, som i en lista av typen enligt uppgift 4 ovan med länkar av linktyp, letar efter en nyckel. Finns nyckel returnerar funktionen sant (1) annars falskt(0). Funktionshuvud enligt:

```
int reksok(linktyp *lista, int key);
```

11)(5p)Skriv ett fullständigt program som bestämmer det minsta och största värdet av de reella elementen i en vektor. Största och minsta värdet ska bestämmas i en funktion med prototypen :

```
void min_max(float rvek[], int nr, float *minpek, float *maxpek);
```

där nr är antalet element i vektorn rvek och minpek och maxpek är pekare till det minsta resp största elementet. Du ska skriva både funktionen och huvudprogrammet. Vektorn ska initieras i huvudprogrammet med 5 valfria reella tal.

---

12)(5p)Skriv ett fullständigt program som upprepat (avslut om längden 0) läser in aktuell längd för en sträng och allokerar minne dynamiskt för denna sträng. Efter allokering läser programmet in strängen och skriver ut den baklänges. Glöm ej att avallokera dynamiskt minne så att ditt program inte äter minne.

---

13)(5p)Fullborda nedanstående huvudprogram genom att skriva koden som utför det som står i kommentarerna. Programmet använder ett binärt träd för att leta upp en rät linje som är parallell med nyckeln. Du ska anropa de givna funktionerna :

```
#include ”Linje.h”

struct nod
{
   linje l;
   struct nod *left, *right;
};

void into_bintree(struct nod **tree, linje l);
/* Stoppar in linjen l i trädet */

struct nod *search_bintree(struct nod *tree, linje l);
/* Om det finns en med linjen l parallell linje i trädet returneras
en pekare till denna nod annars returneras NULL */


void main()
{
   struct nod  *linje_tree = NULL, *keypek;
   linje lv[100] = {{1.0, 5.6}, {2.5, 1.3},...}, key;
   int i;

   /* Stoppa in alla linjerna från vektorn lv i trädet */
   /* Läs in sökt linje */
   /* Sök i trädet efter parallell linje */
   /* Skriv sökresultat */
}
```

---

14)(5p)En tvåpol med beteckning (nr), tomgångsspänning (Ut) och inre resistans (Ri) kan avbildas som en abstrakt datatyp enligt:

```
typedef
struct
{
   int nr;
   float Ut;
   float Ri;
} tvapol;

void las_tvapol(tvapol *tpp); // Läser in nr, Ut och Ri
void skriv_tvapol(tvapol tp); // Skriver ut nr, Ut och Ri
float max_uteffekt_tvapol(tvapol tp); // Beräknar max uteffekt
int mindre_nr_tvapol(tvapol tp1, tvapol tp2);  // Sant om tp1 har
                                               // mindre nr än tp2
```

I textfilen Tvapol.txt finns ett antal tvåpoler med tvåpolens beteckning, tomgångsspänning och inre resistans radvis enligt:

```
123   2.1   0.3
234   1.6   0.7
156   1.8   0.2
. . . . . . . .
```

Skriv ett fullständigt program som läser alla tvåpoler från filen och stoppar in dessa på en stack. Töm sedan stacken och skriv ut alla tvåpoler som har större maximal uteffekt än den sista tvåpolen i filen. För hantering av stacken ska du använda:

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

#include ”tvapol.h”
typedef tvapol datatyp;

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 */
```

---

15)(5p)Binärfilen Tvapol.dat innehåller ett antal tvåpoler enligt uppgift 14 ovan. Skriv ett fullständigt program som läser alla tvapoler från binärfilen till en efter nr sorterad tvåvägslista. Programmet ska sedan fortsätta med att upprepat (avslut 0) fråga efter nr och i tvåvägslistan söka upp den tvåpol som har detta nr och skriva ut tvåpolens övriga data. Sökningen ska avslutas direkt då tvåpolen hittats eller att man sökt igenom hela listan. För hantering av tvåvägslistan ska du använda:

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

#include ”Tvapol.h”
typedef tvapol 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 tentamen i Programmeringsmetodik, 5p, 040601

1)
```
fp = &f;
*fp = 4.5;
```

2)
```
tp->next = lp;
lp = tp;
```

3)
```
4
```

4)
```
while (tp->data != 33)
   tp = tp->next;
tp->data += 5;
```

5)
```
lp = predlink(lp);
elimlink(&lp);
```

6)
```
#include ”Linje.h”
#include <stdio.h>

void las_linje(linje *lp)
{
   printf("Ge riktningskoefficient : ");
   scanf("%f", &lp->k);
   printf("Ge skärning med y-axeln : ");
   scanf("%f", &lp->m);
}
```

7)
```
#include "Linje.h"
#include <stdio.h>

void skriv_linje(linje l)
{
   printf(”Linjen y = %.1f*x”, l.k);
   if (l.m > 0)
      printf(” + %.1f”, l.m)
   else if (l.m < 0)
      printf(”%.1f”, l.m);
}
```

8)
```
#include ”Linje.h”
void main()
{
   linje a, b;

   las_linje(&a);
   las_linje(&b);
   skriv_linje(a);
   printf(” skär x-axeln i x = %.1f\n”, x_linje(a));
   skriv_linje(b);
   printf(” skär x-axeln i x = %.1f\n”, x_linje(b));
}
```

9)
```
void set_bit(int *talpek, int bit_nr)
{
    *talpek = *talpek | (1 << bit_nr);
}
```

10)
```
int rek_sok(linktyp *lista, int key)
{
   linktyp *tp = lista;

   if (tp == NULL)
      return  0;
   else if (key == tp->data)
      return 1;
   else
      return rek_sok(tp->next, key);
}
```

11)
```
#include <stdio.h>
#include <conio.h>
void min_max(float rvek[], int nr, float *minpek, float *maxpek)
{
   int i;

   *minpek = *maxpek = rvek[0];
   for (i = 1; i < nr; i++)
   {
      if (rvek[i] < *minpek)
      {
         *minpek = rvek[i];
      }
      else if (rvek[i] > *maxpek)
      {
         *maxpek = rvek[i];
      }
   }
}

void main()
{
   float rv[5] = {1.2, 3.4, 2.3, 0.9, 2.0}, min, max;

   min_max(rv, 5, &min, &max);
   printf("Min = %.1f\nMax = %.1f\n", min, max);
   getch();
}
```

12)
```
#include <stdio.h>
#include <stdlib.h>
void main()
{
   char *str;
   int i, langd;

   printf("Ge strängens längd (avslut 0) : ");
   scanf("%d", &langd);
   while (langd > 0)
   {
      str = calloc(langd + 1, sizeof(char));
      getchar();
      printf("Ge sträng : ");
      gets(str);
      for(i = langd - 1; i >= 0; i--)
      {
         printf("%c", str[i]);
```

```
      }
free(str);
      printf("Ge strängens längd (avslut 0) : ");
      scanf("%d", &langd);
   }
}
```

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

void main()
{
   struct nod  *linje_tree = NULL, *keypek;
   linje lv[100] = {{1.0, 5.6}, {2.5, 1.3},...}, key;
   int i;

   /* Stoppa in alla linjerna från vektorn lv i trädet */
   for (i =0; i < 100; i++)
   {
      into_bintree(&linje_tree, lv[i]);
   }

   /* Läs in sökt nyckel */
   printf(”Vilken linje söks? ”);
   las_linje(&key);

   /* Sök i trädet */
   keypek = search_bintree(linje_tree, key);

   /* Skriv sökresultat */
   if (keypek != NULL)
   {
      skriv_linje(keypek->l);
   }
   else
   {
      printf(”Finns ingen sådan linje”);
   }
   getch();
}
```

14)
```
#include <stdio.h>
#include <conio.h>
#include "Lifo.h"


void main()
{
   FILE *tsin;
   linktyp *lp = NULL;
   tvapol sista, tv;

   tsin = fopen(”Tvapol.txt”,”rt”);
   while (fscanf(tsin, ”%d%f%f”, &tv.nr, &tv.Ut, &tv.Ri) != EOF)
   {
      push(&lp, tv);
   }
   fclose(tsin);
   sista = tv;
   while (lp != NULL)
   {
      tv = pop(&lp);
      if (max_uteffekt_tvapol(tv) > max_uteffekt_tvapol(sista))
```

```
        {
            skriv_tvapol(tv);
        }
    }
    getch();
}
```

15)

```
#include <stdio.h>
#include <conio.h>
#include "Twolist.h"
#include "Tvapol.h"

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    tvapol tv;
    int found, soknr;

    newhead(&hp);
    bsin = fopen("Tvapol.dat", "rb");
    fread(&tv, sizeof(tvapol), 1, bsin);
    while(!feof(bsin))
    {
        newlink(&lp);
        putlink(tv, lp);
        insort(lp, hp, mindre_nr_tvapol);
        fread(&tv, sizeof(tvapol), 1, bsin);
    }
    fclose(bsin);
    printf("Ge den sökta tvåpolens nr (avslut 0) : ");
    scanf("%d", &soknr);
    while (soknr > 0)
    {
        found = 0;
        lp = firstlink(hp);
        while (lp != NULL && !found)
        {
            tv = getlink(lp);
            if (tv.nr == soknr)
                found = 1;
            else
                lp = succlink(lp);
        }
        if (found)
            skriv_tvapol(tv);
        else
            printf("Hittar ingen sådan tvåpol!");
        printf("Ge den sökta tvåpolens nr (avslut 0) : ");
        scanf("%d", &soknr);
    }
    elimhead(&hp);
    getch();
}
```