ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

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

Tentamen i Programmeringsmetodik, 5p, Au2, Au3, D1 och E1, 010605

Hjälpmedel	: Inga	
Tid	: 14-19	
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 .

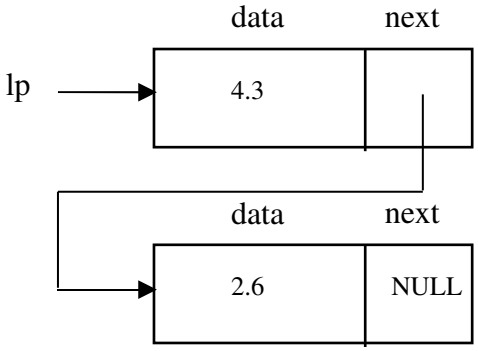
Detta häfte inlämnas.

Lycka till!

1) (1p) Antag att du har en pekare hp som pekar på ett minnesutrymme som innehåller ett heltal. Skriv den scanf-sats som läser in ett nytt heltal till detta minnesutrymme.

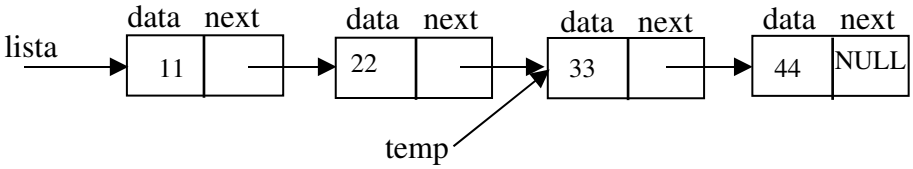
---

2)(1p) Skriv den printf-sats som skriver ut summan av datatermerna för två länkar enligt:



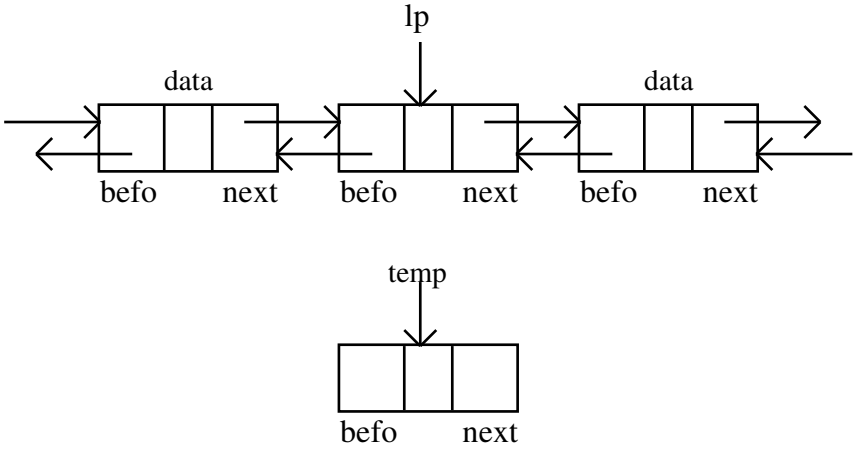
---

3) (1p) Skriv de satser som med hjälp av de pekare som finns i figuren tar bort och avallokerar länken som har 22 som data. Inga extra pekare får definieras och den nya listan ska hänga ihop efteråt.



---

4) (1p) Skriv de satser som krävs för att sätta in länken som temp pekar på före (till vänster) 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 orden i "hej på dig du glada student" instoppas ord för ord i trädet, i den angivna ordningen, om instoppningsfunktionen sätter in data som kommer före i alfabetisk ordning i vänsterträdet och som är lika eller kommer efter i högerträdet.

---

6)(1p) Som index i en hashtabell för ord kan man exempelvis ta första bokstaven (egentligen ASCII för första bokstaven - 'a') . Skissa på hur den aktuella delen av en sådan hashtabell ser ut, då orden ovan i uppgift 5 instoppats. Kollisioner hanteras med stackar.

---

7)(1p) Hur kommer hashtabellen, enligt uppgift 6 ovan att se ut, om man istället använder öppen adressering med hoppfunktionen hopp = 1 och sedan hopp += 2, för att hantera kollisioner?

---

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

```
#include <stdio.h>
void rf(int n)
{
   if (n > 0)
   {
      printf("%d\n", n);
      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 = 7;
```

Ange värdet för uch efter satsen:

```
uch = (uch << 2) | 2;
```

---

10)(1p)Visa hur du skriver ut den mittersta strängen i följande vektor av strängar.

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

11)(2p)Ett positivt allmänt bråk som exempelvis 5 2/3 kan avbildas som en abstrakt datatyp enligt:

```
/* Specifikation av allbrak -- allbrak.h */

typedef struct
{
   int hel;            /* Heltalsdelen, 5 i ex ovan */
   int taljare;        /* Täljaren, 2 i ex ovan */
   int namnare;        /* Nämnaren, 3 i ex ovan */
} allbrak;

void las_allbrak(allbrak *abp);
/* Läs hel, taljare, namnare med ledtext från tangentbordet. */

void skriv_allbrak(allbrak ab);
/* Skriv bråket på skärmen. Om hel är 0 skrivs den inte ut */

void omvandla_allbrak(allbrak *abp);
/* Omvandla bråket till största möjliga hel ex 2 4/3 blir 3 1/3 */

allbrak add_allbrak(allbrak ab1, allbrak ab2);
/* Returnerar summan av ab1 och ab2 */
```

Implementera funktionen skriv_allbrak så att utskriften blir ex. 5 2/3.

---

12)(2p)Implementera funktionen las_allbrak, enligt uppgift 11 ovan, så att inmatningen ser ut som nedan, där du matar in det understrukna.

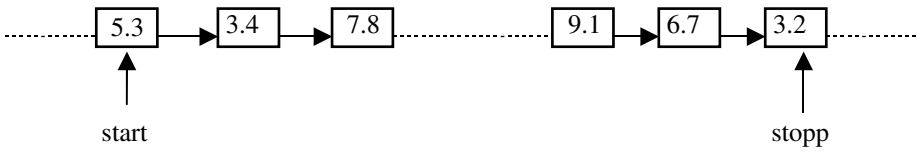
```
Allmänt bråk på formen a b/c? 0 2/3
```

13)(2p)Fullborda funktionen max_15, som ska returnera sant (1) om uch ligger i intervallet 0 till 15, annars falskt (0). Antag att inga jämförelseoperatorer finns tillgängliga för uch och att funktionen därför måste kontrollera att alla de 4 mest signifikanta bitarna är nollställda. Funktionshuvud enligt:

```
int max_15(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 antal nedan, som tar start och stopp som parametrar och som returnerar antalet länkar från och med start och till och med stopp som har data större än x.

```
int antal(linktyp *start, linktyp *stopp, float x)
{
```

---

15)(2p)Skriv en rekursiv funktion som skriver ut en sträng baklänges. Inga färdiga funktioner får användas. Funktionshuvud enligt:

```
void strbak(char *str)
{
```

16)(5p)Skriv ett fullständigt program som läser in värden till två variabler av den abstrakta datatypen allbrak, enligt uppgift 11 ovan, adderar dessa och skriver ut summan i omvandlad form. För full poäng på uppgiften krävs också att du implementerar en av funktionerna add_allbrak eller omvandla_allbrak. Du får själv välja vilken av dessa funktioner du vill implementera.

---

17)(5p)Skriv ett fullständigt program som upprepat (avslutas med 0) läser in reella tal och lägger dessa på en stack. Efter avslutad inläsning ska programmet plocka ut alla tal från stacken och skriva ut de tal som är större än det sist inlästa talet. Talet 0 räknas ej. För hantering av stacken ska du använda:

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

typedef float 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 */
```

---

18)(5p)Implementera alla funktioner för den abstrakta datatypen matpost med specifikation enligt:

```
/* Specifikation av matpost -- Matpost.h */

typedef
struct
{
   int nr;     /* Antal mätvärden */
   float *vek; /* Pekare till första mätvärdet */
} matpost;

void skapa_matpost(matpost *mp, int nr);
/* Sätter antalet mätvärden och allokerar minne för dessa */

void visa_matpost(matpost m);
/* Visar alla mätvärden på skärmen */

float medel_matpost(matpost m);
/* Returnerar medelvärdet av alla mätvärden */

int mindre_matpost(matpost m1, matpost m2);
/* Returnerar sant om m1:s medelvärde < m2:s, annars falskt */
```

19)(5p)I textfilen matning.txt finns ett antal mätposter, enligt uppgift 18 ovan, med först nr och sedan mätvärdena radvis enligt:

```
3  5.4 5.2 5.3
4  2.1 2.5 2.2 2.3
2  4.5 4.6
5  1.2 1.3 1.2 1.4 1.2
1  3.2
.  ....
.  ....
```

Skriv ett fullständigt program som läser all data från filen och stoppar in mätposterna i en tvåvägslista sorterad enligt mindre_matpost-funktionen Avslutningsvis ska programmet skriva ut de sorterade mätvärden radvis på skärmen. För hantering av tvåvägslistan ska du använda:

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

#include ”matpost.h”
typedef matpost 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, 010605

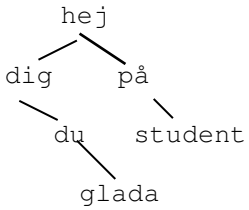
1)
```
scanf("%d", hp);
```

2)
```
printf("%f", lp->data + lp->next->data);
```

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

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

5)



6)
```
d -> du -> dig
h -> hej
g -> glada
p -> på
s -> student
```

7)
```
d -> dig
e -> du
h -> hej
g -> glada
p -> på
s -> student
```

8)
```
2
1
1
2
```

9)
```
30
```

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

11)
```
#include ”allbrak.h”
#include <stdio.h>

void skriv_allbrak(allbrak ab)
{
   if (ab.hel != 0)
      printf("%d %d/%d\n", ab.hel, ab.taljare, ab.namnare);
   else
      printf("%d/%d\n", ab.taljare, ab.namnare);
}
```

12)
```
#include ”allbrak.h”
#include <stdio.h>

void las_allbrak(allbrak *abp)
{
   printf("Allmänt bråk på formen a b/c? ");
   scanf("%d %d/%d", &abp->hel, &abp->taljare, &abp->namnare);
}
```

13)
```
int max_15(unsigned char uch)
{
   int i;

   for (i = 4; i <= 7; i++)
   {
      if (uch & (1 << i) != 0)
         return 0;
   }
   return 1;
}
```

14)
```
int antal(linktyp *start, linktyp *stopp, float x)
{
   int sum = 0;
   linktyp *temp;

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

15)
```
#include <stdio.h>

void strbak(char *str)
{
   if (*str != '\0')
   {
      strbak(str+1);
      putchar(*str);
   }
}
```

16)

```
#include "Allbrak.h"
#include <conio.h>

void main()
{
   allbrak a1, a2, sum;

   las_allbrak(&a1);
   las_allbrak(&a2);
   sum = add_allbrak(a1, a2);
   omvandla_allbrak(&sum);
   skriv_allbrak(sum);
   getch();
}

void omvandla_allbrak(allbrak *abp)
{
   abp->hel += abp->taljare / abp->namnare;
   abp->taljare = abp->taljare % abp->namnare;
}

allbrak add_allbrak(allbrak ab1, allbrak ab2)
{
   allbrak temp;

   temp.hel = ab1.hel + ab2.hel;
   temp.taljare = ab1.taljare*ab2.namnare + ab1.namnare*ab2.taljare;
   temp.namnare = ab1.namnare*ab2.namnare;
   return temp;
}
```

17)

```
#include <stdio.h>
#include <conio.h>
#include "lifo.h"

void main()
{
   linktyp *lp = NULL;
   float x, sist;

   printf("Ge reellt tal (avsluta med 0) : ");
   scanf("%f", &x);
   while (x != 0)
   {
      push(&lp, x);
      printf("Ge reellt tal (avsluta med 0) : ");
      scanf("%f", &x);
   }

   if (lp != NULL)
      sist = pop(&lp);
   while (lp != NULL)
   {
```

```
      x = pop(&lp);
      if (x > sist)
         printf("%.1f\n", x);
   }
   getch();
}
```

18)

```
/* Implementation av matpost -- Matpost.c */

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

void skapa_matpost(matpost *mp, int nr)
{
   mp->nr = nr;
   mp->vek = calloc(nr, sizeof(float));
}

void visa_matpost(matpost m)
{
   int i;

   for (i = 0; i < m.nr; i++)
      printf("%.1f ", m.vek[i]);
   printf("\n");
}

float medel_matpost(matpost m)
{
   int i;
   float sum = 0;

   for (i = 0; i < m.nr; i++)
      sum += m.vek[i];
   return sum / m.nr;
}

int mindre_matpost(matpost m1, matpost m2)
{
   return medel_matpost(m1) < medel_matpost(m2);
}
```

19)

```
/* Huvudprogram -- Matning.c */

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

void main()
{
   headtyp *hp;
   linktyp *lp;
   matpost m;
   int i, antal;
   FILE *tsin;

   tsin = fopen("Matning.txt", "r");
   newhead(&hp);
   while (fscanf(tsin, "%d", &antal) != EOF)
   {
      skapa_matpost(&m, antal);
      for (i = 0; i < antal; i++)
         fscanf(tsin, "%f", &m.vek[i]);
      newlink(&lp);
      putlink(m, lp);
      insort(lp, hp, mindre_matpost);
   }
   fclose(tsin);

   lp = firstlink(hp);
   while (lp)
   {
      m = getlink(lp);
      visa_matpost(m);
      lp = succlink(lp);
   }
   getch();
}
```