ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

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

Tentamen i Programmeringsmetodik C, 5p, D1 och E1, 000524

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 .

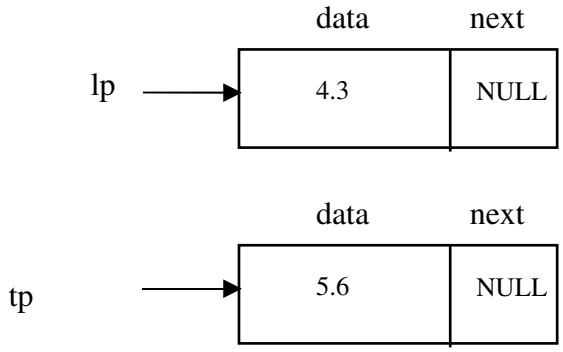
Detta häfte inlämnas.

Lycka till!

1) (1p) Antag att du har en pekare bp som pekar på ett minnesutrymme som innehåller en bokstav. Skriv den sats som skriver ut denna bokstav på skärmen.

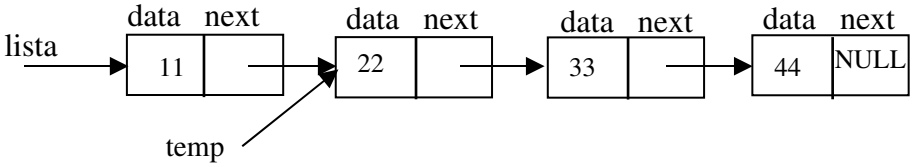
---

2)(1p) Skriv de satser som länkar ihop länkarna, som är av linktyp, nedan så att länken som tp pekar på kommer att ligga efter länken som lp pekar på och både lp och tp kommer att peka på länken med data 4.3.



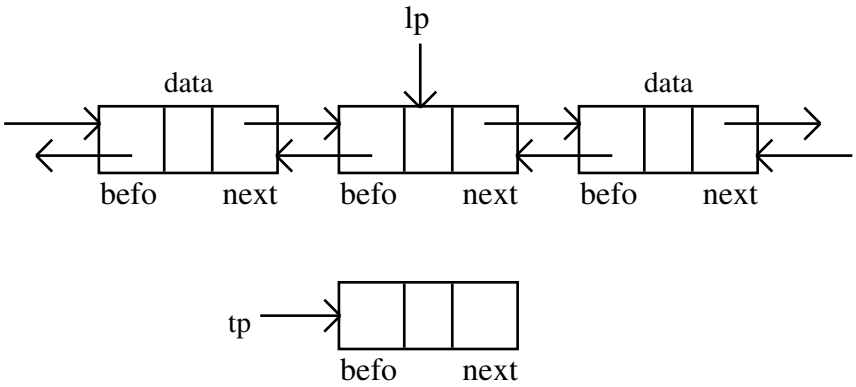
---

3) (1p) Skriv de satser som tar bort och avallokerar länken som lista pekar på i en envägslista enligt figur. Listan ska hänga ihop efteråt och pekaren lista ska peka på samma länk som temp.



---

4) (1p) Skriv de satser som krävs för att stoppa in länken som tp 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 bokstäverna i ordet "program" instoppas bokstav för bokstav i trädet i den angivna ordningen, om instoppningsfunktionen sätter in data som är mindre i högerträdet och lika stora eller större i vänsterträdet.

---

6)(1p) Definiera en hashtabell för hela tal, som använder sig av en länkad lista av LIFO-typ för att hantera kollisioner och hashfunktionen tal % 97.

---

7)(1p) Ange på vilket index i hashtabellen enligt uppgift 6 ovan som talet 120 hamnar.

---

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

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

void main()
{
   rf(3);
}
```

---

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)Sätt sp att peka på första tecknet i strängen str och skriv de satser som använder sp för att skriva strängen str baklänges. Inga färdiga funktioner får användas.

```
char str[] = ”Hej på dig”, *sp;
```

11)(2p)En bil som ska med på en färja kan avbildas enligt:

```
/* Specifikation av bil - pbil.h */

typedef
struct
{
   char reg[7];   /* Registreringsnummer */
   int pers;      /* Antal personer i bilen */
} bil;

int las_bil(bil *bp);
/* Läser in data för en bil från tangentbordet. Returnerar 0 då
endast RETURN ges som reg annars 1 */

void skriv_bil(bil b);
/* Skriver ut reg och pers på skärmen */

int total_vikt(bil b);
/* Returnerar bilens totalvikt i kg*/
```

Implementera funktionen skriv_bil.

---

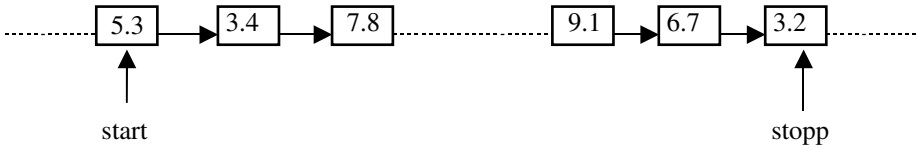
12)(2p)Skriv ett huvudprogram som läser in data för en bil samt visar bilens data och totalvikt på skärmen.

13)(2p)Fullborda funktionen antal_ettor, som ska returnera antalet ettställda bitar i ASCII-koden för uch. Är exempelvis uch ‘A’ med ASCII-koden 01000001 ska funktionen returnera 2 och är uch 'C' med ASCII-koden 01000011 ska funktionen returnera 3.

```
int antal_ettor(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 visa_lista nedan, som tar start och stopp som parametrar och som på skärmen skriver ut alla tal i listan, på en rad åtskilda av en pil (->), från och med start till och med stopp.

```
void visa_lista(linktyp *start, linktyp *stopp)
{
```

---

15)(2p)Skriv en rekursiv funktion som räknar antalet förekomster av en bokstav i ett binärt träd av tecken enligt uppgift 5 ovan. Funktionshuvud enligt nedan där nodtyp är definierad enligt:

```
typedef struct nod
{
   char data;
   struct nod *left, *right;
} nodtyp;

int antal(nodtyp *np, char bokstav)
{
```

16)(5p)Skriv funktionen reverse, som tar en reell vektor och antalet element i vektorn som parametrar och som med hjälp av en LIFO-lista vänder på vektorn. Då funktionen anropas enligt programmet nedan ska utskriften bli:

```
5.5 4.4 3.3 2.2 1.1
```

```
void main()
{
   float vek[5] = {1.1, 2.2, 3.3, 4.4, 5.5};
   int i;

   /* Vänd vektorn */
   reverse(vek, 5);

   /* Skriv ut vektorn */
   for (i = 0; i < 5; i++)
      printf("%.1f ", vek[i]);
}
```

För hantering av LIFO-listan ska du använda

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

typedef float 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 */
```

---

17)(5p)I textfilen postal.txt finns ett antal positiva heltal mellan 0 och 1000, ett tal per rad. Skriv ett program som läser alla tal från filen och skriver ut talen och deras binärkod enligt exempelvis:

```
5 = 00000101
65 = 01000001
300 = 0000000100101100
7 = 00000111
512 = 0000001000000000
1000 = 0000001111101000
96 = 01100000
```

För tal som är mindre än 256 ska binärkoden bestå av 8 bitar och för alla andra tal 16 bitar . För omvandlingen av tal till binära strängar ska du dynamiskt allokera rätt längd på strängen och sedan anropa funktionen:

```
void tal_str(int tal, char *binstr, int nrbits);
```

som du kan anta vara given. Glöm ej att avallokera strängen efter att den använts.

18)(5p)Skriv funktionen total_vikt för den abstrakta datatypen bil enligt uppgift 11 ovan, om vikten för tom bil kan fås genom sökning i binärfilen bilreg.dat, som innehåller bilars registreringsnummer av typen char regnr[7] följt av bilens vikt i hela kg av typen int. Du kan anta att varje person i bilen med bagage väger 100 kg.

---

19)(5p)Skriv ett program som upprepat (avslutas med RETURN) läser in bilar enligt uppgift 11 och lastar in dessa i en kö i form av en tvåvägslista. När ilastningen är klar ska programmet löpa igenom listan från början och räkna antalet bilar samt summera deras totala vikt. Avslutningsvis ska antalet bilar och deras totalvikt skrivas ut på skärmen. För hantering av tvåvägslistan ska du använda:

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

#include ”pbil.h”
typedef bil 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 C, 5p, D1 och E1, 000524

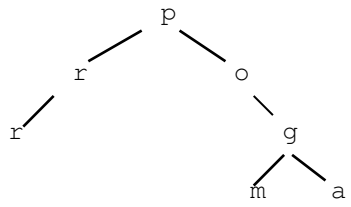
1)
```
printf("%c", *bp);
```

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

3)
```
free(lista);
lista = temp;
```

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

5)



6)
```
struct link
{
   int data;
   struct link *next;
} *hashtabell[97];
```

7)
```
index = 120 % 97 = 23
```

8)
```
3
2
1
0
```

9)
```
3
```

10)
```
sp = str;
while (*sp != '\0')
   sp++;
while ( sp != str)
{
   sp--;
   putchar(*sp);
}
```

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

void skriv_bil(bil b)
{
   printf("Registreringsnummer : %s\n", b.reg);
   printf("Antal personer i bilen : %d\n, b.pers);
}
```

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

void main()
{
   bil b;

   las_bil(&b);
   skriv_bil(b);
   printf("Totalvikt : %d\n", total_vikt(b));
}
```

13)
```
int antal_ettor(unsigned char uch)
{
   int i, sum = 0;

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

14)
```
#include <stdio.h>
void visa_lista(linktyp *start, linktyp *stopp)
{
   linktyp *temp;

   temp = start;
   printf("%.1f", temp->data);
   while (temp != stopp)
   {
      temp = temp->next;
      printf(" -> %.1f", temp->data);
   }
}
```

15)
```
int antal(nodtyp *np, char bokstav)
{
   if (np == NULL)
      return 0;
   else if (np->data == bokstav)
      return 1 + antal(np->left, bokstav);
   return antal(np->left, bokstav) + antal(np->right, bokstav);
}
```

16)

```
/* Re_lifo.c */
#include <stdio.h>
#include "lifo.h"

void reverse(float v[], int nr)
{
   linktyp *lp = NULL;
   int i;

   /* Stoppa vektorelementen på stacken */
   for (i = 0; i < nr; i++)
      push(&lp, v[i]);

   /* Hämta från stacken tillbaka till vektorn */
   for (i = 0; lp != NULL; i++)
      v[i] = pop(&lp);
}
```

17)

```
/* Talbinst.c */
void tal_str(int tal, char *binstr, int nrbits)
{
   int i;

   binstr[nrbits] = '\0';
   for (i = nrbits - 1; i >= 0; i--)
   {
      binstr[i] = tal % 2 + '0';
      tal /= 2;
   }
}

#include <stdio.h>
#include <stdlib.h>
void main()
{
   FILE *tsin;
   int nr;
   char *bstr;

   tsin = fopen("Postal.txt", "r");
   while (fscanf(tsin, "%d", &nr) != EOF)
   {
      if (nr <= 255)
      {
         bstr = calloc(9, sizeof(char));
         tal_str(nr, bstr, 8);
      }
      else
      {
         bstr = calloc(17, sizeof(char));
         tal_str(nr, bstr, 16);
      }
      printf("%d = %s\n", nr, bstr);
      free(bstr);
      bstr = NULL;
   }
   fclose(tsin);
}
```

18)

```
/* Implementation av bil -- pbil.c */

#include "pbil.h"
#include <stdio.h>
#include <string.h>

int las_bil(bil *bp)

{
   printf("Reg : ");
   gets(bp->reg);
   if (bp->reg[0] == '\0')
      return 0;
   printf("Pers : ");
   scanf("%d", &bp->pers);
   getchar();
   return 1;
}

void skriv_bil(bil b)
{
   printf("Reg : %s\n", b.reg);
   printf("Pers : %d\n", b.pers);
}

int total_vikt(bil b)
{
   FILE *bsin;
   int vikt;
   char regnr[7];

   bsin = fopen("bilreg.dat", "rb");
   fread(regnr, sizeof(regnr), 1, bsin);
   fread(&vikt, sizeof(int), 1, bsin);
   while(!feof(bsin) && stricmp(regnr, b.reg) != 0)
   {
      fread(regnr, sizeof(regnr), 1, bsin);
      fread(&vikt, sizeof(int), 1, bsin);
   }
   fclose(bsin);
   return vikt + b.pers*100;
}
```

19)

```
/* Huvudprogram -- bilfarja.c */
#include "twolist.h"
#include <stdio.h>

void main()
{
   headtyp *hp;
   linktyp *lp;
   bil b;
   long totvikt = 0;
   int nr = 0;

   /* läs in bilar till listan */
   newhead(&hp);
   while (las_bil(&b))
   {
      newlink(&lp);
      putlink(b, lp);
      inlast(lp, hp);
   }

   /* Löp igenom listan och summera bilarnas totalvikter */
   lp = firstlink(hp);
   while (lp != NULL)
   {
      b = getlink(lp);
      totvikt += total_vikt(b);
      nr++;
      lp = succlink(lp);
   }

   /* Skriv ut antalet bilar och deras totala vikt */
   printf("Antal bilar: %d\n", nr);
   printf("Totalvikt : %ld kg\n", totvikt);
}
```