



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Kod:.....

Tentamen i Programmeringsmetodik, 5p, Au2, D1 och E1, 020604

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 din kod 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.

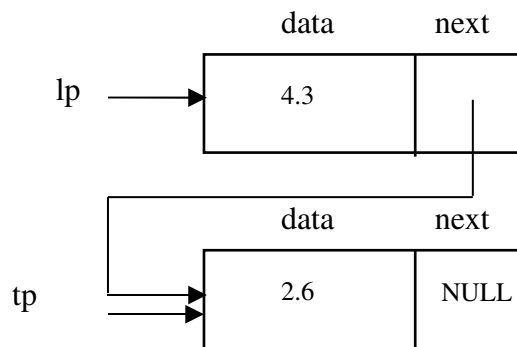
Om inget speciellt anges gäller frågorna Borland C .

Detta häfte inlämnas.

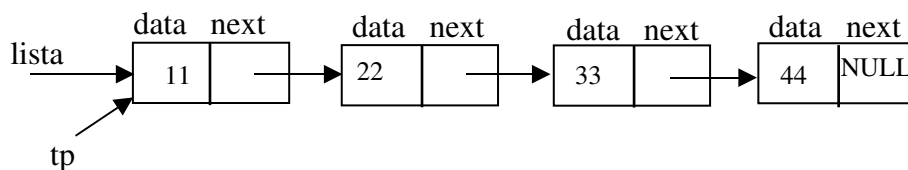
Lycka till!

1) (1p) Antag att du har en pekare rp som pekar på ett minnesutrymme som innehåller ett reellt tal. Skriv de satser som skriver ut värdet av detta tal på skärmen med en decimal.

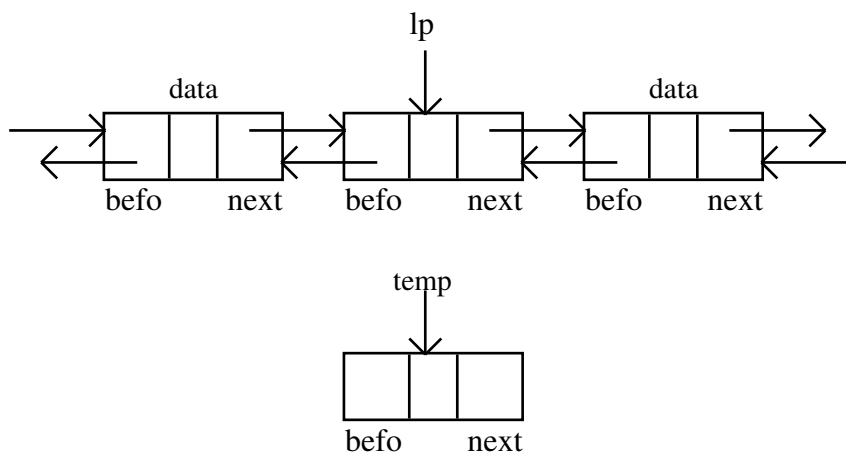
2)(1p) Skriv de satser som sätter lp att peka på den nedersta länken och tp att peka på den översta länken. Du får ej definiera några nya pekare.



3) (1p) Skriv de satser som avallokerar den första länken (längst till vänster) i nedanstående lista. Listan ska hänga ihop efteråt och lista och tp ska peka på länken vars data är 22. Inga extra pekare får användas.



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



5) (1p) Rita det binära träd som skapas då talen 22, 25, 34, 12 och 14 instoppas tal för tal i trädet i den angivna ordningen, om instoppningsfunktionen sätter in data som är mindre i vänsterträdet och som är lika eller större i högerträdet.

6)(1p) Som index i en hashtabell för tal kan man exempelvis ta tal % 10. Skissa på hur en sådan hashtabell ser ut då talen ovan i uppgift 5 instoppas i angiven ordning och om kollisioner hanteras med öppen adressering med hoppfunktionen hopp = 1 första gången och sedan hopp = hopp + 1.

7)(1p) Hur kommer hashtabellen, enligt uppgift 6 ovan att se ut om man istället använder stackar för att hantera kollisioner?

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(3);
}
```

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

```
unsigned char uch = 0xFF;
```

Ange värdet för uch efter satsen:

```
uch &= (1 << 3);
```

10)(1p)Antag att du har följande vektor av strängar:

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

Vad är *str[2]?

11)(2p)En lagervara kan avbildas som en abstrakt datatyp enligt:

```
/* Specifikation -- Vara.h */

typedef struct
{
    char namn[20]; /* Varans namn */
    int antal;     /* Antal i lager */
    int platser;  /* Antal lagerplatser för varan */
} vara;

void las_vara(vara *vp);
/* Frågar efter och läser in vara */

void skriv_vara(vara v);
/* Skriver ut vara på skärmen */

void salj_vara(vara *vp, int nr);
/* Sälj nr st av varan */

void fyll_vara(vara *vp);
/* Fyller på lagret så att alla platser blir fyllda */
```

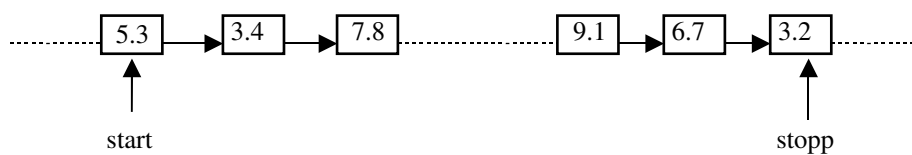
Implementera funktionen las_vara.

12)(2p)Implementera funktionen skriv_vara, enligt uppgift 11 ovan.

- 13)(2p) Fullborda funktionen `positiv`, som ska returnera sant om tal är positivt annars falskt. Inga jämförelseoperatorer finns tillgängliga för tal, så du måste kontrollera om den mest signifikanta biten är nollställd, då är talet positivt. Du kan anta att `int` är 32 bitar. Funktionshuvud enligt:

```
int positiv(int tal)
{
```

-
- 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 `summa` nedan, så att den summerar data för alla länkar från och med `start` till och med `stopp`.

```
float summa(linktyp *start, linktyp *stopp)
{
```

-
- 15)(2p) Skriv en rekursiv funktion som åstadkommer samma sak som funktionen `summa` i uppgift 14 ovan. Funktionshuvud enligt:

```
float rek_summa(linktyp *start, linktyp *stopp)
{
```

16)(5p)Skriv ett fullständigt program som börjar med att fråga efter antalet värden, som en heltalsvektor ska ha och som dynamiskt skapar en sådan vektor. Programmet ska sedan slumpa tresiffriga positiva heltal till vektorn, skriva ut det mittersta värdet i vektorn, om antalet är udda, annars de två mittersta. Avslutningsvis ska det dynamiskt allokerade minnet frigöras. Två körexempel där du matar in det understrukna.

```
Nr? 9
213 355 657 651 119 496 879 368 788
Mittersta är 119
```

```
Nr? 10
755 980 543 534 140 252 315 575 645 143
Två mittersta är 140 och 252
```

17)(5p)Skriv ett fullständigt program som upprepat, avslutas med 0.0, läser reella tal och lägger dessa på en stack. Programmet ska sedan tömma stacken och skriva ut alla tal som är större än det sist inlagda talet (0.0 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 funktionerna fyll_vara och salj_vara, enligt uppgift 11 ovan och skriv ett huvudprogram som testar dessa funktioner. Funktionen salj_vara ska anropa fyll_vara automatiskt då varan är slut i lagret och också restnotera varan, d.v.s. anropa salj_vara med det restnoterade antalet. Funktionen fyll_vara ska alltid fylla på så att lagret blir fullt.

19)(5p) I textfilen lager.txt finns ett antal varor av den abstrakta datatypen vara, enligt uppgift 11 ovan, med namn, antal och antal lagerplatser radvis enligt:

```
ugf123 5 12
hky314 8 14
der252 3 10
gty257 5 18
klo152 7 13
. . . . .
. . . . .
```

Skriv ett fullständigt program som läser alla varor från filen och stoppar in dessa i en tvåvägslista och därefter utför ett antal försäljningar genom att gå igenom hela listan, visa varje vara och fråga hur många som säljs. Avslutningsvis ska programmet skriva ut det aktuella lagret på skärmen.

För hantering av tvåvägslistan ska du använda:

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

#include "Vara.h"
typedef vara 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 */
```

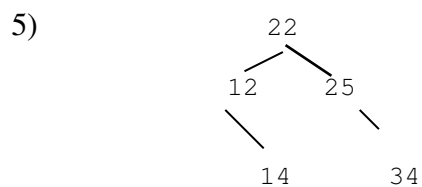

Lösningar till tentamen i Programmeringsmetodik, 5p, 020604

1) `printf("%.1f", *rp);`

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

3) `lista = lista->next;`
`free(tp);`
`tp = lista;`

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



6)

```

2 -> 22
3 -> 12
4 -> 34
5 -> 25
6 -> NULL
7 -> 14
  
```

7)

```

2 -> 12 -> 22
3 -> NULL
4 -> 14 -> 34
5 -> 25
  
```

8) 1
 2
 3

9) 8

10) Tecknet 'd'

11)

```
void las_vara(vara *vp)
{
    printf("Namn? ");
    scanf("%s", vp->namn);
    printf("Antal? ");
    scanf("%d", &vp->antal);
    printf("Platser? ");
    scanf("%d", &vp->platser);
}
```

12)

```
void skriv_vara(vara v)
{
    printf("\nNamn: %s\nAntal: %d\nPlatser: %d\n", v.namn, v.antal,
        v.platser);
}
```

13)

```
int positiv(int tal)
{
    if ((uch & (1 << 31)) == 0)
        return 1;
    return 0;
}
```

14)

```
float summa(linktyp *start, linktyp *stopp)
{
    linktyp *lp = start;
    float sum = lp->data;

    while (lp != stopp)
    {
        lp = lp->next;
        sum += lp->data;
    }
    return sum;
}
```

15)

```
float rek_summa(linktyp *start, linktyp *stopp)
{
    if (start == stopp)
        return start->data;
    else
        return start->data + rek_summa(start->next, stopp);
}
```

16)

```

#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <conio.h>

void main()
{
    int *vp, nr, i;

    printf("Nr? ");
    scanf("%d", &nr);

    vp = calloc(nr, sizeof(int));

    srand((unsigned)time(NULL));
    for (i = 0; i < nr; i++)
    {
        vp[i] = rand() % 900 + 100;
        printf("%d ", vp[i]);
    }

    if (nr % 2)
        printf("\nMittersta är %d\n", vp[nr/2]);
    else
        printf("\nTvå mittersta är %d och %d\n", vp[nr/2-1], vp[nr/2]);

    free(vp);

    getch();
}

```

17)

```

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

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

    printf("Ge x (avslut 0.0): ");
    scanf("%f", &x);
    while (x != 0.0)
    {
        push(&lp, x);
        printf("Ge x (avslut 0.0): ");
        scanf("%f", &x);
    }

    if (lp != NULL)
        sista = pop(&lp);
    while (lp != NULL)
    {
        x = pop(&lp);
        if (x > sista)

```

```
        printf("%f\n", x);
    }

    getch();
}
```

18)

```
#include "Vara.h"

void salj_vara(vara *vp, int nr)
{
    int rest;

    vp->antal -= nr;
    if (vp->antal <= 0)
    {
        rest = -vp->antal;
        fyll_vara(vp);
        salj_vara(vp, rest);
    }
}

void fyll_vara(vara *vp)
{
    vp->antal = vp->platser;
}

/* Huvudprogram -- Varamain.c */

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

void main()
{
    vara v;

    las_vara(&v);
    salj_vara(&v, 3);
    skriv_vara(v);

    salj_vara(&v, 8);
    skriv_vara(v);

    getch();
}
```

19)

```
/* Varulist.c */

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

void main()
{
    FILE *tsin;
    headtyp *hp;
    linktyp *lp;
    vara v;
    int nr;

    tsin = fopen("Lager.txt", "r");
    newhead(&hp);
    while (fscanf(tsin, "%s%d%d", v.namn, &v.antal, &v.platser) != EOF)
    {
        newlink(&lp);
        putlink(v, lp);
        inlast(lp, hp);
    }
    fclose(tsin);

    lp = firstlink(hp);
    while (lp != NULL)
    {
        v = getlink(lp);
        skriv_vara(v);
        printf(" Sälj? ");
        scanf("%d", &nr);
        salj_vara(&v, nr);
        putlink(v, lp);
        lp = succlink(lp);
    }

    lp = firstlink(hp);
    while (lp != NULL)
    {
        v = getlink(lp);
        skriv_vara(v);
        lp = succlink(lp);
    }

    getch();
}
```