



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Lämna in ifylld kursvärdering tillsammans med tentamen!

Lösningarna till tentamensuppgifterna sätts ut på kurssidan på nätet i dag kl 13.

Denna tenta kommer att vara färdigrättad Ti 7/6 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 och då har du ingen möjlighet att klaga på rättningen.

Tentamen i Programmeringsmetodik, 5p, Au2, D1, Fri 050531.

Hjälpmedel : Inga
Tid : 08-13
Ansvarig lärare : Gunnar Joki 303317, 274825(hem), 0705474825(mob)

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 Visual C++.

Detta häfte ska du behålla.

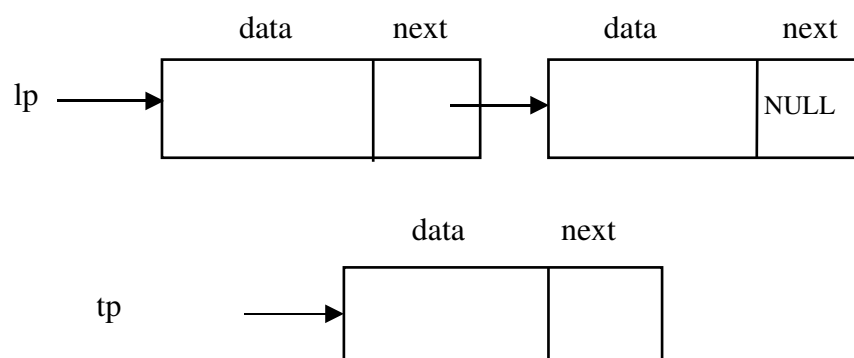
Lycka till!

1) (1p) Anta att du har en pekare given enligt:

```
float *fp = NULL;
```

Visa hur du dynamiskt allokerar minne för ett reellt tal som fp sätts att peka på.
Tilldela slutligen det reella talet värdet 54.5.

2)(1p) Stoppa in länken tp sist i den länkade strukturen nedan. Inga extra pekare får definieras och tp ska efter instoppning peka på samma länk som lp, alltså den första. Den sista structens nextpekare ska alltid vara NULL.



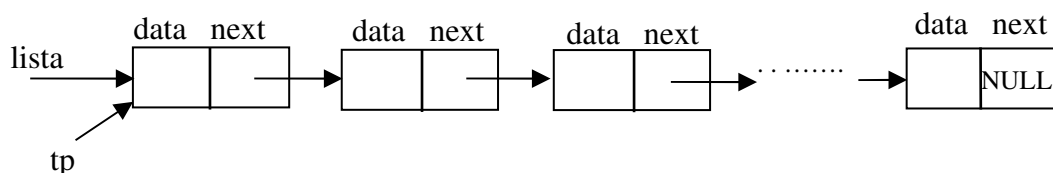
3) (1p) Antag att du har en 8 bitars unsigned char definierad enligt:

```
unsigned char uch = 0x7;
```

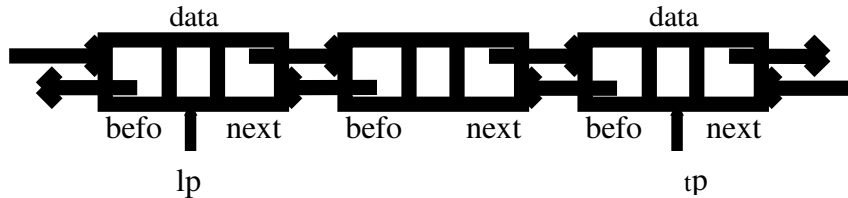
Ange värdet decimalt av uch efter satsen:

```
uch |= 20;
```

4) (1p) Skriv de satser som tar bort och avallokerar den sista länken (längst till höger) nedan. Inga extra pekare får definieras. Efter bortplockandet ska lista och tp peka på samma länk som före avallokeringen



5)(1p) Ange hur du genom att använda de färdiga funktionerna i Twolist (se uppgift 15 nedan) byter plats mellan länkarna lp och tp nedan.



6)(2p) Ett elektriskt instrument kan avbildas som en abstrakt datatyp med typ, visat värde och mätfel enligt:

```

/* Instrument.h */
typedef
struct
{
    char bet[20]; /* Instrumentbeteckning V1, A12 etc. . . . .*/
    float visning; /* Visat värde */
    float fel; /* Mätfel i % av visat värde */
}instrument;

void las_instrument(instrument *ip);
/* Läser in data med ledtexter */

void skriv_instrument(instrument i);
/* Skriver ut data med ledtexter */

float visning_instrument(instrument i);
/* Returnerar det visade värdet */

float absfel_instrument(instrument i);
/* Returnerar det absoluta felet som visning * fel / 100 */

int mindre_instrument(instrument i1, instrument i2);
/* Returnerar sant (1) om i1:s beteckning kommer före i2:s i
alfabetisk ordning */

int samma_instrument(instrument i1, instrument i2);
/* Returnerar sant (1) om i1:s beteckning samma som i2:s annars 0 */

```

Implementera funktionen `las_instrument`.

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

8)(2p) Skriv ett huvudprogram som skapar ett instrument, enligt uppgift 6 ovan, läser in all data och skriver ut instrumentets visning med angivet absolut fel.

- 9)(2p) Du ska skicka bytes där de 7 minst signifikanta bitarna i varje byte utgör ASCII-koden för det tecken som skickas och den åttonde biten den mest signifikanta är en paritetskontrollbit. Skriv en funktion som sätter denna åttonde bit så att pariteten blir udda dvs. antalet satta bitar totalt i varje skickad byte blir ett udda tal.
Funktionsprototyp enligt:

```
void set_udda_paritet(unsigned char *uchp );
```

- 10)(2p) Antag att du har en lista enligt uppgift 4 ovan där data är av typen instrument enligt uppgift 6 ovan. Skriv den funktion som skriver ut en tabell, med alla instrument i listan, radvis med beteckning, visning och fel i %, enligt:

```
A12 4.5 1.5
V23 1.2 1.0
. . . . .
```

Funktionshuvud enligt:

```
void instr_tab(struct link *lista);
```

- 11)(5p) Skriv ett program som dynamiskt allokerar en vektor som ska innehålla ett inläst antal instrument, enligt uppgift 6 ovan. Läs in instrumenten till vektorn och skriv ut medelvärdet av visningarna för instrumenten i vektorn.
-

- 12)(5p) Implementera funktionerna visning_instrument, absfel_instrument, mindre_instrument och samma_instrument för den abstrakta datatypen instrument i uppgift 6 ovan.
-

- 13)(5p) Skriv ett fullständigt program som läser in alla reella mätvärden från textfilen Retal.txt till en stack och som frågar efter en reell nyckel, tömmer stacken och skriver ut alla tal som är större än nyckeln. För hantering av stacken 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 */

```

- 14)(5p)Skriv ett fullständigt program som slumpar ett inläst antal tärningskast, stoppar in alla tärningsutfall i ett binärt träd, skriver ut trädet, frågar efter en nyckel (1 – 6) och räknar och skriver ut antalet utfall som är lika med nyckeln. För det binära trädet ska du använda modulen Tree nedan där alla funktioner är givna utom antal_lika_utfall som du ska skriva själv. Funktionen intotree är skriven så att den stoppar in lika stora tal i högerträdet.

```

/* Specifikation av binärt heltalsträd -- Tree.h */

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

void intotree(nodtyp **rpp, int d);
/* Stoppar in d i trädet */

void showtree(nodtyp *rp);
/* Skriver ut trädet */

int antal_lika_utfall(nodtyp *rp, int key);
/* Beräknar antalet likadana utfall av typen key i trädet */

```

- 15)(5p)Binärfilen Instrument.dat innehåller ett antal instrument enligt uppgift 6 ovan. Skriv ett program som läser alla instrument från binärfilen och placerar dessa i en tvåvägslista sorterad efter beteckning. Använd funktionerna för den abstrakta datatypen instrument i uppgift 6 ovan. Sök sedan efter inläst instrumentbeteckning och skriv ut dess data. För hantering av tvåvägslistan ska du använda:

```

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

#include "Instrument.h"
typedef instrument 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 */
1
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 050531

- 1)

```
fp = malloc(sizeof(float));
*fp = 54.5;
```
- 2)

```
lp->next->next = tp;
tp->next = NULL;
tp = lp;
```
- 3)

```
00000111
00010100
-----
or 00010111 = 23
```
- 4)

```
while(tp->next->next != NULL)
{
    tp = tp->next;
}
free(tp->next);
tp->next = NULL;
tp = lista;
```
- 5)

```
outlist(tp);
inpred(tp, lp);
outlist(lp);
insucc(succlink(tp), lp);
```
- 6)

```
#include "Instrument.h"
#include <stdio.h>

void las_instrument(instrument *ip)
{
    printf("Ge instrumentets beteckning : ");
    gets(ip->bet);
    printf("Ge instruments visning : ");
    scanf("%d", &ip->visning);
    printf("Ge instrumentens fel : ");
    scanf("%d", &ip->fel);
}
```
- 7)

```
#include "Instrument.h"
#include <stdio.h>
void skriv_instrument(instrument i)
{
    printf("Beteckning : %s\n", i.bet);
    printf("Visning : %.2f\n", i.visning);
    printf("Fel : %.1f\n", i.fel);
}
```
- 8)

```
#include "Instrument.h"
#include <conio.h>
void main()
{
    instrument a;

    las_instrument(&a);
    printf("Mätresultat : %.2f +- %.1f\n", visning_instrument(a),
        absfel_instrument(a));
}
```

```

    getch();
}

9) void set_udda_paritet(unsigned char *uchp )
{
    int i, sum = 0;

    for (i = 0; i < 7; i++)
    {
        if (*uchp & (1 << i) != 0)
            sum++;
    }
    if (sum % 2 == 0)
        *uchp |= (1 << 7)
}

10) #include <conio.h>
#include <stdio.h>
#include "Instrument.h"

void instr_tab(struct link *lista)
{
    struct link *tp = lista;

    while(tp != NULL)
    {
        skriv_instrument(tp->data);
        tp = tp->next;
    }
    getch();
}

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

void main()
{
    int nr ;
    float sum = 0.0;
    instrument *ip;

    printf("Ge antal instrument (avslut 0) : ");
    scanf("%d", &nr)
    ip = calloc(nr, sizeof(instrument));
    for (i = 0; i < nr; i++)
    {
        las_instrument(&ip[i]);
        sum += visning_instrument(ip[i])
    }
    printf("Medelvisning : %.1f\n", sum / nr);
    free(ip);
    getch();
}

```



```

12) #include "Instrument.h"
#include <string.h>

float visning_instrument(instrument i)
{
    return i.visning;
}

float absfel_instrument(instrument i)
{
    return i.fel * i.visning /100;
}

int mindre_instrument(instrument i1, instrument i2)
{
    return strcmp(i1.bet, i2.bet) < 0;
}

int samma_instrument(instrument i1, instrument i2)
{
    return strcmp(i1.bet, i2.bet) == 0;
}

13) #include <stdio.h>
#include <conio.h>
#include "Lifo.h"

void main()
{
    linktyp *lp = NULL;
    FILE *tsin;
    float key, x;

    tsin = fopen("Retal.txt", "rt");
    while( fscanf(tsin, "%f", &x) != EOF)
    {
        push(&lp, x);
    }
    printf("Ge nyckel : ");
    scanf("%f", &key);

    while (lp != NULL)
    {
        x = pop(&lp);
        if (x > key)
            printf("%.1f\n", x);
    }
    getch();
}

```

```
14) #include <stdio.h>
#include <stdlib.h>
#include <conio.h>

int antal_lika_utfall(nodtyp *rp, int key)
{
    if (rp == NULL)
    {
        return 0;
    }
    else if(rp->data == key)
    {
        return 1 + antal_lika_utfall(rp->right, key);
    }
    else if (key < rp->data)
    {
        return antal_lika_utfall(rp->left, key);
    }
    else
    {
        return antal_lika_utfall(rp->right, key);
    }
}

void main()
{
    nodtyp *rot = NULL;
    int slump, antal, key;

    srand((unsigned), time(NULL));
    printf("Ge antalet kast : ");
    scanf("%d", &antal);

    for (i = 1; i <= antal; i++)
    {
        slump = rand() % 6 + 1;
        intotree(&rot, slump);
    }
    showtree(rot);

    printf("Ge sökt utfall : ");
    scanf("%d", &key);

    printf("Antal %d = %d\n", key, antal_lika_utfall(rot, key));

    getch();
}
```

```
15) #include <stdio.h>
#include <conio.h>
#include "Instrument.h"
#include "Twolist.h"

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    instrument instr, sokinstr ;

    newhead(&hp);
    bsin = fopen("Instrument.dat", "rb");
    fread(&instr, sizeof(instrument), 1, bsin);
    while(!feof(bsin))
    {
        newlink(&lp);
        putlink(instr, lp);
        insert(lp, hp, mindre_instrument);
        fread(&instr, sizeof(instrument), 1, bsin);
    }
    fclose(bsin);
    printf("Ge det sökta instrumentets beteckning : ");
    scanf("%s", sokinstr.bet);
    lp = firstlink(hp);
    while (lp != NULL)
    {
        instr = getlink(lp);
        if (samma_instrument(instr, sokinstr))
            skriv_instrument(instr);
        lp = succlink(lp);
    }
    elimhead(&hp);
    getch();
}
```