



ÖREBRO UNIVERSITET
INSTITUTIONEN FÖR TEKNIK

Lösningarna till tentamensuppgifterna sätts ut på kurssidan på nätet under eftermiddagen.

**Omtentamen i Programmeringsmetodik, 5p (1ED030, TDD122), 2007-08-30.
[D1, E1, A2, Pr1, SDU2, TDVB]**

Hjälpmedel : Inga
Tid : 08:00-13:00
Ansvarig lärare : Christer Lindkvist 303393, 070-3273393

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 i högermarginalen. Totalt kan 40 poäng erhållas. För betyget 3 krävs ca 20, för betyget 4 ca 28 och för betyget 5 ca 34 poäng.

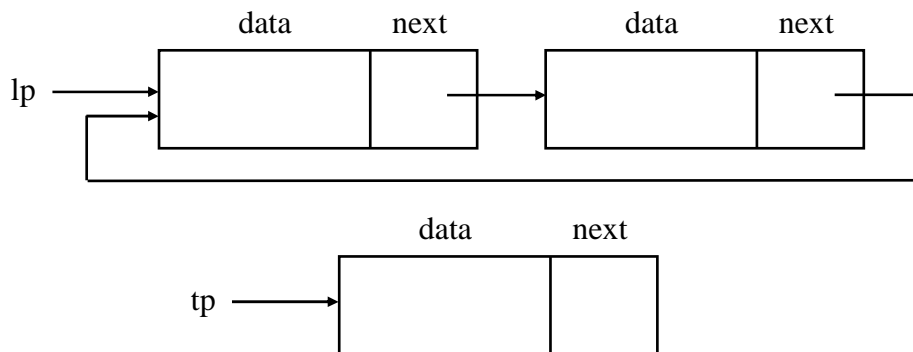
Om inget speciellt anges gäller frågorna Visual C++.

Detta häfte ska du behålla.

Lycka till!

- 1) Antag att du har en pekare **cp** som pekar på ett minnesutrymme som innehåller ett ASCII-tecken. Skriv en sats som byter tecknet mot nästa tecken i ASCII-tabellen. (1p)

- 2) Stoppa in länken som **tp** pekar på efter länken som **lp** pekar på. Stega därefter fram pekaren **lp** så att den pekar på nästa länk i den länkade strukturen nedan. Inga extra pekare får definieras. Den sista structens nextpekare ska alltid peka på den första länken. (1p)



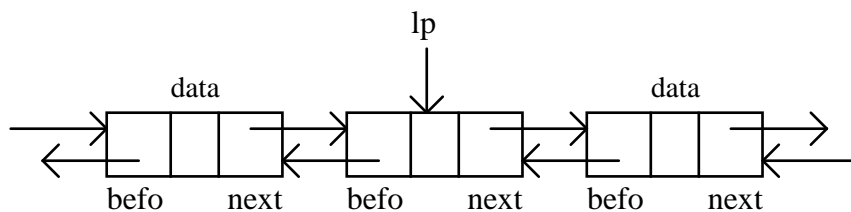
- 3) Antag att du har en 8 bitars unsigned char definierad enligt: (1p)

```
unsigned char uch = 0x45;
```

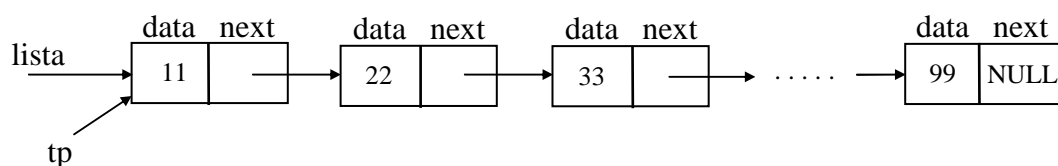
Ange värdet decimalt av **uch** efter satsen:

```
uch &= 45;
```

- 4) 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. (1p)



- 5) Skriv de satser som nollställer alla datavärden i nedanstående länkade struktur. (1p)



6) Ett tentamensresultat kan avbildas enligt:

```

/* Specifikation av tentamensresultat - tentres.h */

typedef struct {
    char namn[20];    /* Namnet på den som tentat */
    int poang;       /* Antal erhållna poäng */
} tentres;

void las_tentres(tentres *trp);
/* Läser in namn och poäng från tangentbordet. */

void skriv_tentres(tentres tr);
/* Skriver ut namn och poäng på skärmen. */

int jfr_tentres(tentres tr1, tentres tr2);
/* Returnerar sant om tr1 har lägre poäng än tr2. */

char satt_betyg(tentres tr, int gr3, int gr4, int gr5);
/* Returnerar betygen U, 3, 4, 5 utgående */
/* ifrån poänggränserna gr3, gr4 och gr5 */

```

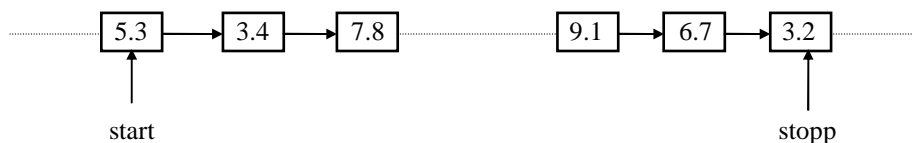
Implementera funktionerna **las_tentres** och **skriv_tentres** enl. ovan. (2p)

7) Implementera funktionerna **jfr_tentres** och **satt_betyg** i uppgift 6 ovan. (2p)

8) Fullborda funktionen **antal_noll** som ska returnera antalet nollställda bitar i uch. Är exempelvis uch 'A' med ASCII-koden 01000001 ska funktionen returnera 6. (2p)

```
int antal_noll(unsigned char uch)
```

9) 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 den *iterativa* funktionen **antal_lankar_iter** nedan, så att den returnerar antalet länkar mellan **start** och **stopp**. (2p)

```
int antal_lankar_iter(linktyp *start, linktyp *stopp)
```

10) Fullborda den *rekursiva* funktionen **antal_lankar_rek** nedan, så att den returnerar antalet länkar mellan **start** och **stopp** i uppgift 9 ovan. Funktionshuvud enligt: (2p)

```
int antal_lankar_rek(linktyp *start, linktyp *stopp)
```

-
- 11) Skriv ett program som först läser in antalet element och dynamiskt skapar en vektor med detta antal element av den abstrakta datatypen **tentres** enligt uppgift 6 ovan. Programmet ska fortsätta med att läsa in alla tentamensresultat och sedan skriva ut dessa på skärmen. Avslutningsvis ska vektorn avallokeras. (5p)
-

- 12) 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: (5p)

```
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 anropa den givna funktionen:

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

- 13) 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. Skriv därefter ett litet huvudprogram som skapar en vektor och skriver ut den både före och efter anrop av **reverse**. (5p)

För hantering av LIFO-listan ska du använda

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

- 14) Binärfilen **Bilar.dat** innehåller ett antal bilar av den abstrakta datatypen **bil** nedan. Skriv ett program som läser alla bilar från binärfilen och placerar dessa i en hashtabell med länkade listor. Därefter ska en söknyckel i form av en bil läsas in. Efter sökning i hashtabellen ska sökresultatet skrivas ut. För att hantera hashtabellen skall du endast använda den abstrakta datatypen **linkhash** nedan och för att hantera bilar ska du endast använda den abstrakta datatypen **bil**.

(5p)

```

/* Specifikation av bilar - bil.h */

typedef struct bilpost {
    char regnr[7];
    char agare[40];
} biltyp;

void skapa_bil(biltyp *bp, char *regnr, char *agare);
/* Skapar en bil utifrån medskickad data (se las_bil) */

int las_bil(biltyp *bp);
/* Läser in data för en bil från tangentbordet.          */
/* Returnerar 0 då endast RETURN ges som regnr, annars 1 */
/* Om endast RETURN anges som ägare blir ägaren "UNKNOWN" */

void skriv_bil(biltyp b);
/* Skriver ut regnr och agare på skärmen */

int hashfunk_bil(biltyp b);
/* Returnerar summan av ASCII-koderna för b.regnr */

int samma_bil(biltyp b1, biltyp b2);
/* Returnerar 1 om b1 och b2 har samma regnr, annars 0 */

/* Specifikation av länkad hashtabell -- linkhash.h */

#include "bil.h"
typedef biltyp datatyp; /* Exempelvis */

#define HASHTABSIZE 100

typedef struct link {
    datatyp data;
    struct link *next;
} linktyp;

void init_linkhash(linktyp *htab[]);
/* NULL-ställer hashtabellen */

void into_linkhash(linktyp *htab[], datatyp d,
                  int (*hashfunk)(datatyp));
/* Stoppas in d i hashtabell enligt hashfunk */

datatyp *search_linkhash(linktyp *htab[], datatyp nyckel,
                        int (*hashfunk)(datatyp),
                        int (*is_equal)(datatyp, datatyp));
/* Söker efter nyckel i hashtabell */

```

- 15) I binärfilen **tentres.dat** finns ett antal tentamensresultat av typen **tentres** enligt uppgift 6 ovan. Skriv ett program som läser hela filen och stoppar in resultaten sorterade efter poäng i en tvåvägslista.

Efter instoppning ska listan genomlöpas och resultatet i form av **tentres** och betyg skrivs ut. Betyg sätts genom anrop av funktionen **satt_betyg**, där gränserna är 20, 28 och 34 poäng precis som på tentan som du skriver just nu.

För hantering av tentamensresultat ska du endast använda den abstrakta datatypen **tentres** i uppgift 6 ovan och för hantering av tvåvägslistan ska du endast använda den abstrakta datatypen **twolist** nedan.

```

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

#ifndef TWOLISTH
#define TWOLISTH

#include "tentres.h"
typedef tentres 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 */

#endif
```

Lösningar till tentamen i Programmeringsmetodik, 5p 2007-08-30

- 1) `*cp = *cp + 1;` Exempelvis `*cp += 1;` och `(*cp)++;` fungerar också.
Det gör däremot **inte** `*cp++;` (pekaren uppräknas).
- 2)

```
tp->next = lp->next;
lp->next = tp;
lp = lp->next;
```
- 3)
$$\begin{array}{l} 01000101 = 0x45 \\ \& 00101101 = 0x2D = 45 \\ \hline 00000101 = 0x05 = 5 \end{array}$$
- 4)

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

```
while (tp) {
    tp->data = 0;
    tp = tp->next;
}
```
- 6)

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

void las_tentres(tentres *trp)
{
    printf("Namn: ");
    gets(trp->namn);
    printf("Poäng: ");
    scanf("%d", &trp->poang);
    getchar();
}

void skriv_tentres(tentres tr)
{
    printf("Namn : %s\n", tr.namn);
    printf("Poäng : %d\n", tr.poang);
}
```
- 7)

```
int jfr_tentres(tentres tr1, tentres tr2)
{
    return tr1.poang < tr2.poang;
}

char satt_betyg(tentres tr, int gr3, int gr4, int gr5)
{
    if (tr.poang < gr3)
        return 'U';
    else if (tr.poang < gr4)
        return '3';
    else if (tr.poang < gr5)
        return '4';
    else
        return '5';
}
```


- 8)

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

    for (i = 0; i <= 7; i++) {
        if ((uch & (1 << i)) == 0)
            sum++;
    }
    return sum;
}
```
- 9)

```
int antal_lankar_iter(linktyp *start, linktyp *stopp)
{
    linktyp *temp;
    int sum = 0;

    if (start != stopp) {
        temp = start->next;
        while (temp != stopp) {
            sum++;
            temp = temp->next;
        }
    }
    return sum;
}
```
- 10)

```
int antal_lankar_rek(linktyp *start, linktyp *stopp)
{
    if (start == stopp)
        return 0;
    else if (start->next == stopp)
        return 0;
    else
        return 1 + antal_lankar_rek(start->next, stopp);
}
```
- 11)

```
#include <stdlib.h>           /* Huvudprogram -- tentamen.c */
#include <stdio.h>
#include "tentres.h"

void main()
{
    tentres *tv;
    int i, nr;

    /* Skapa en vektor med ett inläst antal tentamensresultat */
    printf("Antal studenter : ");
    scanf("%d", &nr);
    getchar();
    tv = calloc(nr, sizeof(tentres));

    /* Läs tentamensresultat till vektorn */
    for (i = 0; i < nr; i++)
        las_tentres(&tv[i]);

    /* Visa tentamensresultat på skärmen */
    for (i = 0; i < nr; i++)
        skriv_tentres(tv[i]);

    /* Frigör minne */
    free(tv);
    tv = NULL;
}
```

```

12) /* Funktionen tal_str ingår ej i uppgiften! */
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>
```

```

void main()
{
    FILE *tsin;
    int nr;
    char bstr[20];

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

```

```

13) #include <stdio.h>
#include "lifo.h"

void reverse(double 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);
}

void print_vek(char str[], double v[], int nr)
{
    int i;

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

```

```

void main()
{
    double vek[] = {1.1, 2.2, 3.3, 4.4, 5.5};
    int size = sizeof(vek)/sizeof(double);

    print_vek("Before: ", vek, size);
    reverse(vek, size);
    print_vek("After:  ", vek, size);
}

```

14) `#include <stdio.h>`
`#include "bil.h"`
`#include "linkhash.h"`

```

void main()
{
    FILE *bsin;
    biltyp bil, nyckel, *bp;
    linktyp *hashtab[HASHTABSIZE];

    /* NULL-ställ hashtab */
    init_linkhash(hashtab);

    /* Stoppa in alla bilar från filen in i hashtab */
    bsin = fopen("Bilar.dat", "rb");
    fread(&bil, sizeof(biltyp), 1, bsin);
    while(!feof(bsin)) {
        into_linkhash(hashtab, bil, hashfunkt_bil);
        fread(&bil, sizeof(biltyp), 1, bsin);
    }
    fclose(bsin);

    /* Läs in sökt registreringsnummer */
    printf("Ge sökt regnr, ägaren kan lämnas blank!\n");
    las_bil(&nyckel);

    /* Sök i hashtab och skriv ut resultat */
    if (bp = search_linkhash(hashtab, nyckel, hashfunkt_bil, samma_bil))
        skriv_bil(*bp);
    else
        printf("Bilen finns ej i registret!\n");
}

```

```
15) /* Huvudprogram -- betyg.c */
#include <stdio.h>
#include "tentres.h"
#include "twolist.h"

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    tentres tr;

    /* Läs data från fil till lista */
    newhead(&hp);
    bsin = fopen("tentres.dat", "rb");
    fread(&tr, sizeof(tentres), 1, bsin);
    while (!feof(bsin)) {
        newlink(&lp);
        putlink(tr, lp);
        insort(lp, hp, jfr_tentres);
        fread(&tr, sizeof(tentres), 1, bsin);
    }
    fclose(bsin);

    /* Skriv ut data och sätt betyg */
    lp = firstlink(hp);
    while (lp != NULL) {
        tr = getlink(lp);
        skriv_tentres(tr);
        printf("Betyg : %c\n\n", satt_betyg(tr, 20, 28, 34));
        lp = succlink(lp);
    }

    /* Frigör minne */
    elimhead(&hp);
}
```