



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Lösningarna till tentamensuppgifterna sätts ut på kurssidan på nätet idag kl 13.

Denna tenta kommer att vara färdiggrättad Må 19/1 -04 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.

Omtentamen i Programmeringsmetodik, 5p, Au2, D1, E1, Pr1, 040114.

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

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.

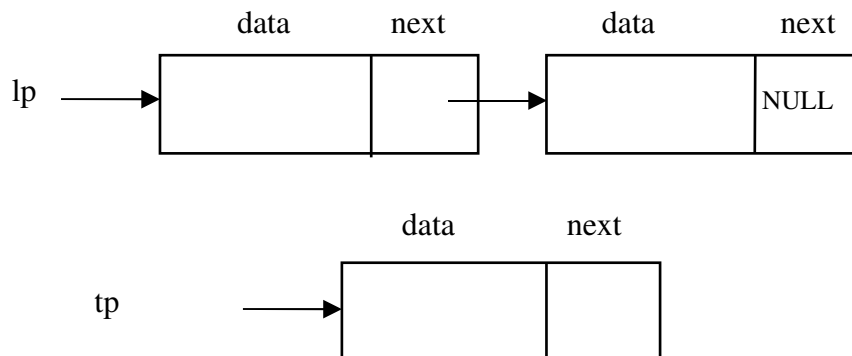
Om inget speciellt anges gäller frågorna Borland C.

Detta häfte ska du behålla.

Lycka till!

1) (1p) Allokera en reell variabel dynamiskt och läs in ett värde till den.

2)(1p) Stoppa in länken tp mellan de två befintliga länkarna i den länkade strukturen nedan. Inga extra pekare får definieras.



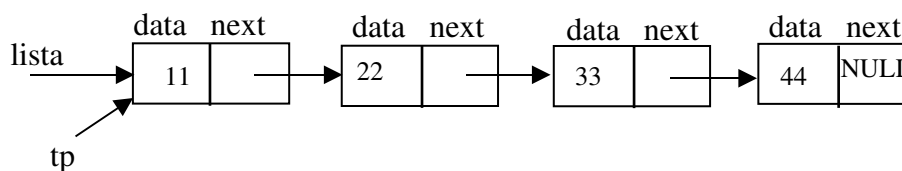
3) (1p) Antag att du har en 8 bitars unsigned char definierad enligt:

```
unsigned char uch = 0xa;
```

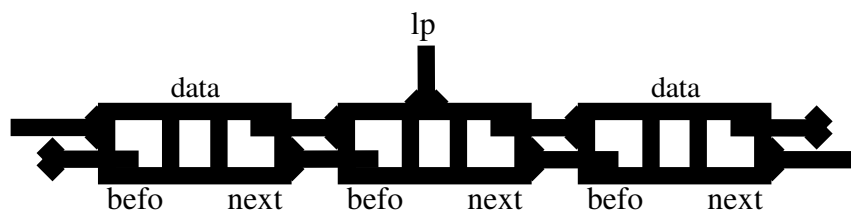
Ange värdet decimalt av uch efter satsen:

```
uch |= 4;
```

4) (1p) Skriv de satser som tar bort och avallokerar den första (längst till vänster) länken i en lista av nedanstående typ. Lista och tp ska efteråt peka på länken med data 22.



5) (1p) Ange hur du tar bort länken lp från nedanstående tvåvägslista så att listan hänger ihop efteråt. Inga extra pekare får definieras och lp ska ha next och befo NULL-ställda efter bortplockandet. Inga färdiga funktioner får användas.



6)(2p) Alla grundämnen kan avbildas som en abstrakt datatyp med atomnummer (nr), symbol(s) och atommassa (m) enligt:

```

/* Grundamne.h */

typedef
struct
{
    int nr;          /* Atomnummer */
    char s[3];      /* Symbol    */
    float m;        /* Atommassa */
} grundamne;

void las_grundamne(grundamne *gp);
/* Läser in nr, s och m för grundämnet med ledtexter*/

void skriv_grundamne (grundamne g);
/* Skriver ut grundämnets nr, s och m med ledtexter */

int lattare(grundamne g1, grundamne g2);
/* Returnerar sant(1) om atommassan för g1 är mindre än för g2,
annars falskt(0) */

int halvledare(grundamne g);
/* Returnerar sant(1) om grundämnet är en halvledare, annars
falskt(0). Halvledare är grundämnena med atomnummer 14, 32, 33, 50,
51, 52 och 85 */

```

Implementera funktionen las_grundamne.

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

8)(2p) Skriv ett huvudprogram som skapar två grundämnen enligt uppgift 6 ovan, läser in dessa och som skriver ut dessa i ordning med den lättaste först.

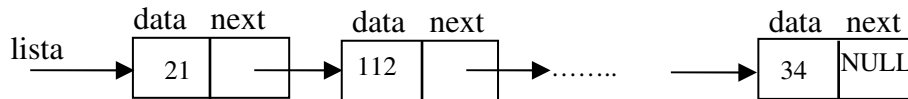
9)(2p) Skriv en funktion, is_set_bit, som tar ett heltal och ett bitnummer (0 – 31) som parameter och som returnerar sant (1) om denna bit är satt i talet annars falskt (0). Funktionsprototyp enligt:

```
int is_set_bit(int tal, int bit_nr);
```

10)(2p) Skriv den rekursiva funktionen, rek-summa, som tar två heltalsparametrar och returnerar summan av alla heltal från och med det första parametervärdet till och med det andra. Ett exempel på anrop som visar hur man skriver ut summan av talen mellan 1 och 100 :

```
printf("Summan 1+2+3+. . .+100 = %d", rek_summa(1, 100));
```

11)(5p) Skriv ett fullständigt program som bestämmer det minsta och största värdet av heltalen data i envägslistan enligt :



Listan antas given av den konstanta globala pekaren lista som ej kan ändras och länkarna är av typen linktyp enligt :

```

typedef struct link
{
    int data;
    struct link *next;
} linktyp;
  
```

12)(5p) Skriv ett fullständigt program som simulerar ett test av tärningar. Programmet ska upprepat (avslut 0) fråga efter hur många tärningskast du vill slumpa och dynamiskt allokera en vektor med detta antal element, så att varje kast kan sparas i den. Programmet ska sedan upprepat (avslut 0) fråga efter vilket tärningsutfall (1-6) som du är intresserad av att undersöka, räkna ut antalet sådana utfall och skriva ut hur stor del av det totala antalet utfall som dessa utgör. Resultatet ska ges i procent med en decimal.

13)(5p) Fullborda nedanstående program genom att skriva koden som utför det som står i varje kommentar. Programmet använder en hashtabell för sökning av grundämnen enligt uppgift 6 ovan med hashfunktionen atomnummer % 10 och stackar för kollisionshantering. Du ska anropa de givna funktionerna :

```

struct link
{
    grundamne g;
    struct link *next;
};

void into_linkhash(struct link *htab[], grundamne g);
/* Stoppar in g i htab */

struct link *search_linkhash(struct link *htab[], grundamne nyckel);
/* Returnerar pekare till nyckel om den finns i htab annars NULL*/

void main()
{
    struct link *hashtab[10], *keypek;
    grundamne gv[25] = {{11, "Na", 22.99}, {2, "He", 4.003}, ...}, key;
    int i;
  
```

```

    /* NULL-ställ hashtab */
    /* Stoppa in alla element från vektorn gv i htab */
    /* Läs in sökt nyckel */
    /* Sök i hashtabellen */
    /* Skriv sökresultat */
}

```

14)(5p)Vissa grundämnen är halvledare. Skriv funktionen halvledare, med specifikation enligt uppgift 6 ovan och ett fullständigt program som läser alla grundämnen från textfilen Grundamne.txt enligt nedan och skriver symbol, atomnummer och atommassa för alla som är halvledare.

```

Cu    29    63.55
Ge    32    72.59
..    ..    .. ..

```

15)(5p)Det periodiska systemet som innehåller alla grundämnen med symbol, atomnummer och atommassa finns i binärfilen Period.dat. Skriv ett fullständigt program som läser alla grundämnen från binärfilen till en tvåvägslista. Programmet ska sedan fortsätta med att fråga antingen efter symbol eller atomnummer och i tvåvägslistan söka upp detta grundämne och skriva ut detta grundämnes övriga data. För hantering av tvåvägslistan ska du använda:

```

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

#include "Grundamne.h"
typedef grundamne 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, 040114

1)

```
float *xp = malloc(sizeof(float));
printf("Ge ett reellt tal : ");
scanf("%f", xp);
```

2)

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

3) 14

4)

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

5)

```
lp->befo->next = lp->next;
lp->next->befo = lp->befo;
lp->next = lp->befo = NULL;
```

6)

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

void las_grundamne(grundamne *gp)
{
    printf("Ge atomnummer : ");
    scanf("%d", &gp->nr);
    printf("Ge atommassa: ");
    scanf("%f", &gp->m);
    getchar();
    printf("Ge symbol : ");
    gets(gp->s);
}
```

7)

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

void skriv_grundamne(grundamne g)
{
    printf("Atomnummer : %d\n", g.nr);
    printf("Atommassa : %.1f\n", g.m);
    printf("Symbol : %s\n", g.s);
}
```

8)

```
#include "Grundamne.h"
void main()
{
    grundamne g1, g2;

    las_grundamne(&g1);
    las_grundamne(&g2);
    if (lattare(g1, g2))
    {
        skriv_grundamne(g1);
        skriv_grundamne(g2);
    }
    else
    {
        skriv_grundamne(g2);
        skriv_grundamne(g1);
    }
}
```

```

}
```

```

9) int is_set_bit(int tal, int bit_nr)
{
    return tal & (1 << bit_nr);
}
```

```

10) int rek_summa(int min, int max)
{
    if (min < max)
    {
        return min + reksumma(min+1, max);
    }
    return min;
}
```

```

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

void main()
{
    linktyp *tp = lista;
    int min, max;

    min = max = tp->data;
    tp = tp->next;
    while (tp != NULL)
    {
        if (tp->data < min)
        {
            min = tp->data;
        }
        else if (tp->data > max)
        {
            max = tp->data;
        }
        tp = tp->next;
    }
    printf("Min = %d\nMax = %d\n", min, max);
    getch();
}
```

```

12) #include <stdio.h>
#include <stdlib.h>

void main()
{
    int i, nr, *v, utfall, antal;

    randomize();
    printf("Ge antal tärningskast som ska slumpas (avslut 0) : ");
    scanf("%d", &nr);
    while (nr > 0)
    {
        v = calloc(nr, sizeof(int));

        for(i = 0; i < nr; i++)
```



```

        {
            v[i] = random(6) + 1;
        }
printf("Vilket utfall vill du kontrollera (avslut 0) : ");
scanf("%d", &utfall);
while(utfall > 0)
    {
        antal = 0;
        for (i = 0; i < nr; i++)
            {
                if (v[i] == utfall)
                    {
                        antal++;
                    }
            }
        printf("Procentuell förekomst: %.1f\n", (float)antal/nr*100);
printf("Vilket utfall vill du kontrollera : ");
scanf("%d", &utfall);
    }
    free(v);
    printf("Ge antal tärningskast som ska slumpas (avslut 0) : ");
    scanf("%d", &nr);
}
}

```

13)

```

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

struct link
{
    grundamne g;
    struct link *next;
};

void into_linkhash(struct link *htab[], grundamne g);
/* Stoppar in g i htab */

struct link *search_linkhash(struct link *htab[], grundamne nyckel);
/* Returnerar pekare till nyckel om den finns i htab annars NULL*/

void main()
{
    struct link *hashtab[10], *keypek;
    grundamne gv[25] = {{11, "Na", 22.99}, {2, "He", 4.003}, ...};
    grundamne key;
    int i;

    /* NULL-ställ hashtab */
    for (i = 0; i < 10; i++)
        {
            hashtab[i] = NULL;
        }

    /* Stoppa in alla element från vektorn gv i htab */
    for(i = 0; i < 25; i++)
        {
            into_linkhash(hashtab, gv[i])
        }

    /* Läs in sökt nyckel */

```

```

las_grundamne(&key);

/* Sök i hashtabellen */
keypek = search_linkhash(hashtab, key);

/* Skriv sökresultat */
if (keypek != NULL)
{
    skriv_grundamne(keypek->g);
}
else
    printf("Nyckeln finns ej!");
getch();
}

```

14)

```

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

int halvledare(grundamne g)
{
    return g.nr == 14 || g.nr == 31 || g.nr == 32
           || (g.nr >= 50 && g.nr <= 52) || g.nr == 85;
}

void main()
{
    FILE *tsin;
    grundamne g;

    tsin = fopen("Grundamne.txt", "rt");
    while (fscanf(tsin, "%s%d%f", g.s, &g.nr, &g.m) != EOF)
    {
        if (halvledare(g))
        {
            skriv_grundamne(g);
        }
    }
    fclose(tsin);
    getch();
}

```

15)

```

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

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    grundamne g;
    int found;
    char val, soksym[3];
    int sokanr;

    newhead(&hp);
    bsin = fopen("Period.dat", "rb");

```

```

fread(&g, sizeof(grundamne), 1, bsin);
while(!feof(bsin))
{
    newlink(&lp);
    putlink(g, lp);
    inlast(lp, hp);
    fread(&g, sizeof(grundamne), 1, bsin);
}
fclose(bsin);

printf("Vill du leta med atomnummer(a), symbolnamn(s) eller
        avsluta sökningen (e) : ");
val = getch();
while (val != 'e')
{
    if (val == 'a')
    {
        printf("Ge sökt atomnummer : ");
        scanf("%d", &sokanr);
    }
    else if (val == 's')
    {
        printf("Ge sökt symbolnamn : ");
        scanf("%s", soksym);
    }
}
found = 0;
lp = firstlink(hp);
while (lp != NULL && !found)
{
    g = getlink(lp);
    if (val == 'a' && sokanr == g.nr)
        found = 1;
    else if (val == 's' && strcmp(soksym, g.s) == 0)
        found = 1;
    else
        lp = succlink(lp);
}
if (found)
    skriv_grundamne(g);
else
    printf("Hittar ingen sådant grundämne!");

printf("Vill du leta med atomnummer(a), symbolnamn(s) eller
        avsluta sökningen (e) : ");
val = getch();
}

elimhead(&hp);
getch();
}

```