



ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

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

Denna tenta kommer att vara färdigrättad To 1/9 och kan då hämtas på mitt tjänsterum T2221 mellan 10 och 12. 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.

Omtentamen i Programmeringsmetodik, 5p, Au2, D1, Fri 050826

Hjälpmedel : Inga
Tid : 8-13
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.

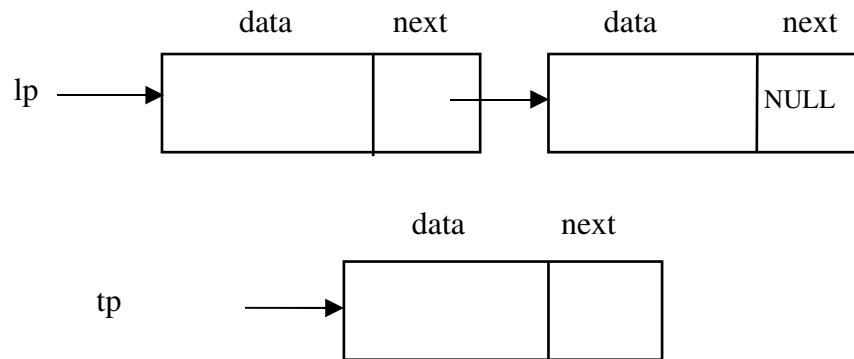
Om inget speciellt anges gäller frågorna Visual C++.

Detta häfte ska du behålla.

Lycka till!

1) (1p) Anta att du har en reell variabel f . Definiera en pekare som pekar på denna variabel och använd pekaren till att fördubbla värdet av f .

2)(1p) Stoppa in länken tp mellan de två länkarna i den länkade strukturen nedan. Inga extra pekare får definieras och tp ska efter instoppning peka på samma länk som lp , alltså den första.



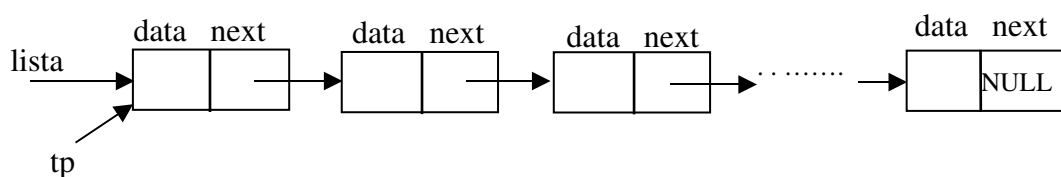
3) (1p) Antag att du har en 8 bitars unsigned char definierad enligt:

```
unsigned char uch = 0x88;
```

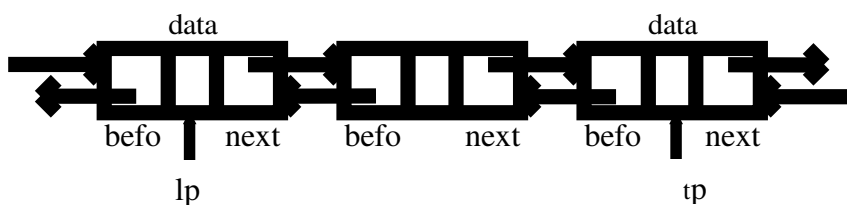
Ange värdet decimalt av uch efter satsen:

```
uch &= 28;
```

4) (1p) Skriv de satser som tar bort och avallokerar den första länken (längst till vänster) nedan. Inga extra pekare får definieras. Efter bortplockandet ska lista och tp peka på den andra länken, som nu är den första.



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



6)(2p) En tentamen kan avbildas som en abstrakt datatyp med kod och erhållna poäng enligt:

```

/* Tentamen.h */

typedef
struct
{
    int kod;                /* Tentandes kod */
    float poang;           /* Antal erhållna poäng på tentan */
}tentamen;

void las_tentamen(tentamen *tp);
/* Läser in data med ledtexter */

void skriv_tentamen(tentamen t);
/* Skriver ut data med ledtexter */

char betyg_tentamen(tentamen t, float bg3, float bg4, float bg5);
/* Returnerar betyget U, 3, 4 eller 5 utgående ifrån betygsgränserna
   bg3 för en trea, bg4 för en fyra och bg5 för en femma */

int mindre_poang(tentamen t1, tentamen t2);
/*Returnerar sant (1) om t1 har mindre poäng än t2 annars falskt (0)*/

```

Implementera funktionen skriv_tentamen

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

8)(2p) Implementera funktionen mindre_poang, i uppgift 6 ovan.

9)(2p) Du ska ta emot tecken som unsigned char via en port på adressen portadress(ex 0x80). På adressen portadress + 5 finns det ett register där du ska kontrollera bitnummer 3 som är satt om data har kommit in till porten. Skriv en funktion som kontrollerar om det finns någon data att läsa från porten och om det finns returnerar det lästa tecknet annars returneras tecknet '\0'.

```
unsigned char get_port(unsigned char *portadress);
```

10)(2p) Antag att du har en lista enligt uppgift 4 ovan som innehåller ett antal tentamensdata enligt uppgift 6 ovan. Skriv de satser som visar poängen för alla tentander och skriver ut en tabell enligt:

Kod	Poäng
====	=====
34	23.5
12	32.5
...	...

11)(5p)Skriv ett fullständigt program som skapar en dynamiskt allokerad sträng med inläst längd, läser in strängen och skriver ut strängen baklänges. Programmet ska upprepa så länge den inlästa längden ej är 0. Glöm ej att avallokera det dynamiska minnet mellan varje upprepning.

12)(5p)Implementera funktionen `betyg_tentamen` för den abstrakta datatypen `tentamen` i uppgift 6 ovan och skriv ett huvudprogram som skapar en student, läser in data för studenten och skriver ut studentens betyg om gränserna är samma som på denna tenta.

13)(5p)Skriv ett fullständigt program som läser alla reella tal från textfilen "Retal.txt" till en stack och sedan skriver in talen tillbaks till samma textfil. 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 läser in alla filnamn från textfilen `Filer.txt` till en dynamiskt allokerad vektor av filnamn. Antalet filnamn framgår av det första talet i textfilen. Efter läsning ska programmet kontrollera om det finns några filnamn i vektorn som är lika och i så fall ta bort alla dubletter och endast ha kvar det först förekommande filnamnet. Textfilen kan exempelvis se ut som :

```
34
Hej.c
Lots.cpp
Hej.c
Lita.txt
help.cpp
. . .
```

Som avslutning ska programmet skriva ut vektorn med filnamnen.

15)(5p)Binärfilen Tentamen.dat innehåller ett antal tentamina enligt uppgift 6 ovan. Skriv ett program som läser alla tentamen från binärfilen och placerar dessa i en tvåvägslista sorterad efter poäng. Använd funktionerna för den abstrakta datatypen tentamen i uppgift 6 ovan. Fråga efter kod och skriv ut tentamensresultatet. 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);
/* Elimineras och NULL-ställer listan */
```

Lösningar till tentamen i Programmeringsmetodik, 5p 0508?

- 1)

```
float *fp = &f;
*fp *= 2;
```
- 2)

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

```
10001000
00011100
-----
and 00001000 = 8
```
- 4)

```
tp = tp->next;
free(lista);
lista = tp;;
```
- 5)

```
temp = getlink((tp);
putlink(getlink(lp), tp);
putlink(temp, tp);
```
- 6)

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

void skriv_tentamen(tentamen t)
{
    int i;

    printf("Kod : %d\n", t.kod);
    printf("Poäng : %.1f\n", t.poang);
}

```
- 7)

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

void las_tentamen(tentamen *tp)
{
    int i;
    printf("Ge tentamenskoden : ");
    scanf("%d", &tp->kod);
    printf("Ge antal poäng : ");
    scanf("%f", &tp->poang);
}

```
- 8)

```
int mindre_poang(tentamen t1, tentamen t2)
{
    return t1.poang < t2.poang;
}

```
- 9)

```
unsigned char get_port(unsigned char *portadress)
{
    if( *(portadress + 5) & (1<<3))
        return *port;
    return '\0';
}

```

```
}

```

```
10) #include <conio.h>
#include <stdio.h>
#include "Tentamen.h"

void main()
{
    struct link
    {
        tentamen data;
        struct link *next;
    } *tp;

    printf("Kod          Poäng\n");
    printf("====          =====\n");
    tp = lista;
    while(tp != NULL)
    {
        printf("%d          %.1f\n", tp->data.kod, tp->data.poäng);
        tp = tp->next;
    }
    getch();
}

```

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

void main()
{
    char *namn;
    int langd, i;

    printf("Ge namnets längd (avslut 0) : ");
    scanf("%d", &langd);
    while (langd > 0)
    {
        namn = calloc(langd + 1, sizeof(char));
        printf("Ge namn : ");
        gets(namn);
        for (i = langd-1; i >= 0; i--)
            printf("%c", namn[i]);
        free(namn);
        printf("Ge namnets längd (avslut 0) : ");
        scanf("%d", &langd);
    }
}

```

```
12) #include "Tentamen.h"

char betyg_tentamen(tentamen t, float bg3, float bg4, float bg5)
{
    if (t.poang >= bg5)
        return '5';
    else if (t.poang >= bg4)
        return '4';
    else if (t.poang >= bg3)

```



```

        return '3';
    else
        return 'U';
}

void main()
{
    tentamen a;

    las_tentamen(&a);
    printf("Betyg : %c", betyg_tentamen(a, 20, 28, 34);
    getch();
}

```

13)

```

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

void main()
{
    FILE *tsinut;
    linktyp *lp = NULL;
    float x;

    tsinut = fopen("Retal.txt", "r+t");
    while (fscanf(tsinut, "%f", &x) != EOF)
    {
        push(x, &lp);
    }
    rewind(tsinut)
    while (lp != NULL)
    {
        x = pop(&lp);
        fprintf(tsinut, "%f ", x);
    }
    fclose(tsinut);
    getch();
}

```

14)

```

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

void main()
{
    FILE *tsin;
    char filnamn[30];
    char **filvek;
    int i, k, nr;

    tsin = fopen("Filer.txt", "rt");
    fscanf(tsin, "%d", &nr);
    fgetc(tsin);
    filvek = calloc(nr, sizeof(char *));
    for (i = 0; i < nr; i++)
    {
        fgets(filnamn, strlen(filnamn), tsin);
        filvek[i] = calloc(strlen(filnamn) + 1, sizeof(char));
        strcpy(filvek[i], filnamn);
    }
}

```

```

}
for (i = 0; i < nr-1; i++)
{
    for (k = i+1; k < nr; k++)
    {
        if (strcmp(filvek[i], filvek[k]) == 0)
        {
            free(filvek[k]);
            filvek[k] = NULL;
            nr--;
        }
    }
}
for (i = 0; i < nr; i++)
{
    if (filvek[i] != NULL)
        printf ("%s", filvek[i]);
}
getch();
}

```

15)

```

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

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    tentamen t;
    int sokkod;

    newhead(&hp);
    bsin = fopen("Tentamen.dat", "rb");
    fread(&t, sizeof(tentamen), 1, bsin);
    while(!feof(bsin))
    {
        newlink(&lp);
        putlink(t, lp);
        insort(lp, hp, mindre_poang);
        fread(&t, sizeof(tentamen), 1, bsin);
    }
    fclose(bsin);
    printf("Ge den sökta koden : ");
    scanf("%d", &sokkod);
    lp = firstlink(hp);
    while (lp != NULL)
    {
        t = getlink(lp);
        if (t.kod == sokkod)
        {
            skriv_tentamen(t);
            lp = succlink(lp);
        }
    }
    elimhead(&hp);
    getch();
}

```


Lösningar till tentamen i Programmeringsmetodik, 5p 0508?

- 1) `float *fp = &f;`
`*fp *= 2;`
- 4) `tp->next = lp->next;`
`lp->next = tp;`
`tp = lp;`
- 5) `10001000`
`00011100`

`and 00001000 = 8`
- 4) `tp = tp->next;`
`free(lista);`
`lista = tp;;`
- 5) `putlink(getlink(tp), lp);`
- 7) `#include "Tentamen.h"`
`#include <stdio.h>`

`void skriv_tentamen(tentamen t)`
`{`
`int i;`

`printf("Kod : %d\n", t.kod);`
`printf("Poäng : %.1f\n", t.poang);`
`for (i = 0; i < 3; i++)`
`printf("Gräns för betyget %d %.1f\n", i+3, t.betygsgranser[i]);`
`}`
- 7) `#include "Tentamen.h"`
`#include <stdio.h>`

`void las_tentamen(tentamen *tp)`
`{`
`int i;`
`printf("Ge tentamenskoden : ");`
`scanf("%d", &tp->kod);`
`printf("Ge antal poäng : ");`
`scanf("%f", &tp->poang);`
`printf("Ge betygsgränser : ");`
`for (i = 0; i < 3; i++)`
`scanf("%f", &tp->betygsgrans[i]);`
`}`
- 8) `int mindre_poang(tentamen t1, tentamen t2)`
`{`
`return t1.poang < t2.poang;`
`}`
- 9) `unsigned char get_port(volatile unsigned char *portadress)`
`{`
`if(*(portadress + 5) & (1<<3))`
`return *port;`
`return '\0';`
`}`

```

}

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

void main()
{
    struct link
    {
        tentamen data;
        struct link *next;
    } *tp;

    printf("Kod          Poäng\n");
    printf("===          =====\n");
    tp = lista;
    while(tp != NULL)
    {
        printf("%d          %.1f\n", tp->data.kod, tp->data.poäng);
        tp = tp->next;
    }
    getch();
}

```

```

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

void main()
{
    char *namn;
    int langd, i;

    printf("Ge namnets längd (avslut 0) : ");
    scanf("%d", &langd);
    while (langd > 0)
    {
        namn = calloc(langd + 1, sizeof(char));
        printf("Ge namn : ");
        gets(namn);
        for (i = langd-1; i >= 0; i--)
            printf("%c", namn[i]);
        free(namn);
        printf("Ge namnets längd (avslut 0) : ");
        scanf("%d", &langd);
    }
}

```

```

12) #include "Tentamen.h"

char betyg_tentamen(tentamen t)
{
    if (t.poang >= t.betygsgrans[2])
        return '5';
    else if (t.poang >= t.betygsgrans[1])
        return '4';
    else if (t.poang >= t.betygsgrans[0])
        return '3';
}

```

```

        else
            return `U`;
    }

void main()
{
    tentamen a;

    las_tentamen(&a);
    printf(Betyg : %c", betyg_tentamen(a));
    getch();
}

```

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

```

void main()
{
    FILE *tsinut;
    linktyp *lp = NULL;
    float x;

    tsinut = fopen("Retal.txt", "r+t");
    while (fscanf(tsinut, "%f", &x) != EOF)
    {
        push(x, &lp);
    }
    rewind(tsinut)
    while (lp != NULL)
    {
        x = pop(&lp);
        fprintf(tsinut, "%f ", x);
    }
    fclose(tsinut);
    getch();
}

```

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

```

void main()
{
    FILE tsin;
    char filnamn[20];
    char **filvek;
    int i, nr;

    tsin = fopen("Filer.txt", "rt");
    fscanf(tsin, "%d", &nr);
    filvek = malloc(nr * sizeof(char *));
    for (i = 0; i < nr; i++)
    {
        filvek[i] = calloc(strlen(filvek[i]) + 1, sizeof(char));
        fgets(filvek[i], sizeof(filvek[i]), tsin);
    }
    for (i = 0; i < nr-1; i++)
    {
        for (k = i+1; k < nr; k++)
        {

```

```

        if strcmp(filvek[i], filvek[k]) == 0)
        {
            free(filvek[k]);
            filvek[k] = NULL;
        }
    }
}

```

```

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

void main()
{
    FILE *bsin;
    headtyp *hp;
    linktyp *lp;
    tentamen t;
    int sokkod;

    newhead(&hp);
    bsin = fopen("Tentamen.dat", "rb");
    fread(&t, sizeof(tentamen), 1, bsin);
    while(!feof(bsin))
    {
        newlink(&lp);
        putlink(t, lp);
        insort(lp, hp, mindre_poang);
        fread(&t, sizeof(tentamen), 1, bsin);
    }
    fclose(bsin);
    printf("Ge den sökta koden : ");
    scanf("%d", &sokkod);
    lp = firstlink(hp);
    while (lp != NULL)
    {
        t = getlink(lp);
        if (t.kod == sokkod)
        {
            skriv_tentamen(t);
            lp = succlink(lp);
        }
    }
    elimhead(&hp);
    getch();
}

```