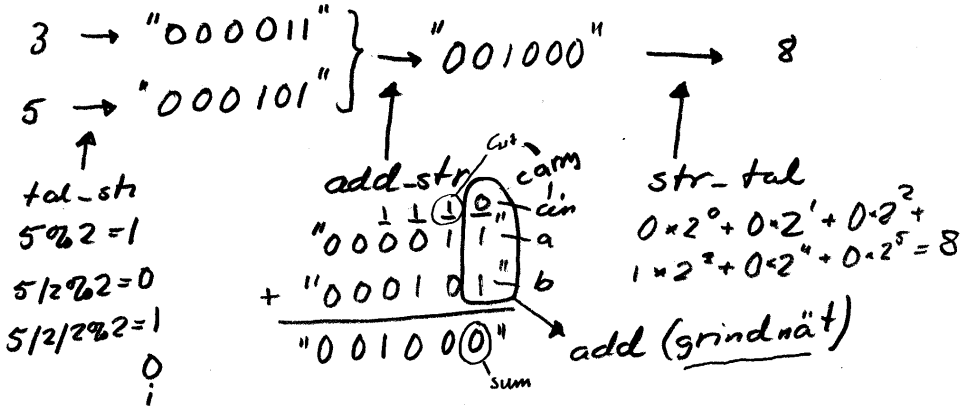
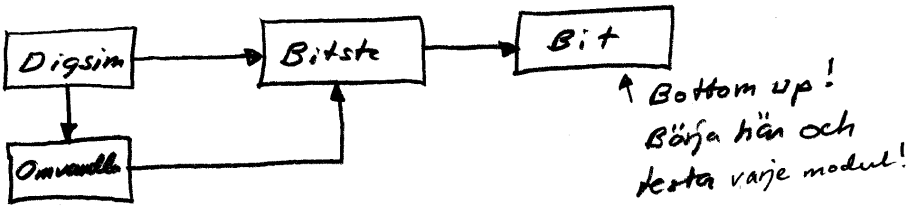
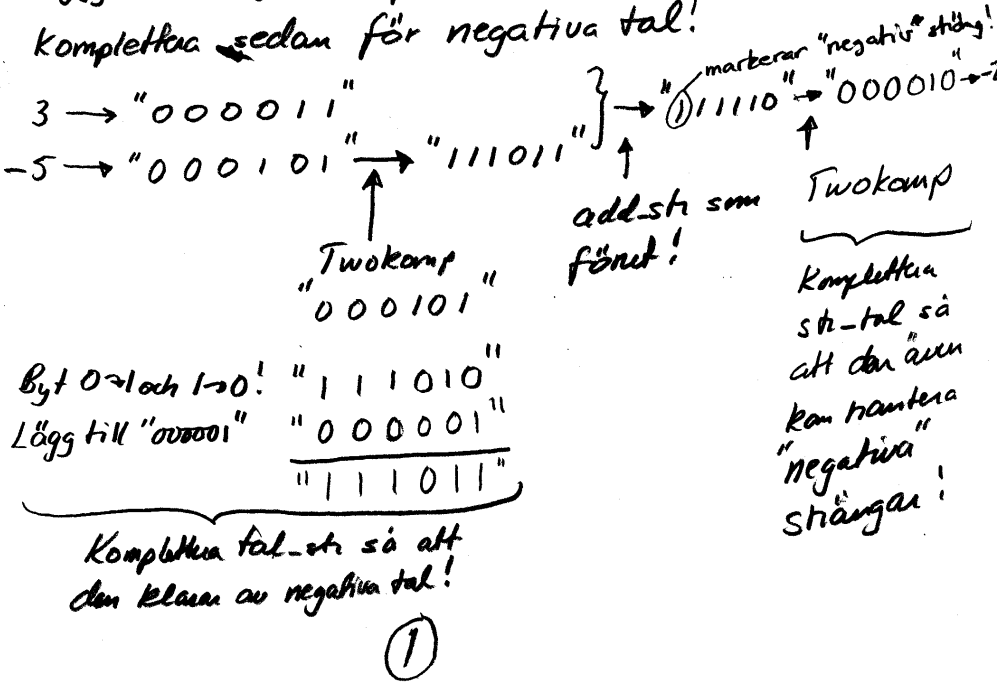
Digsim – simulering av ALU (Aritmetic Logic Unit)





Bygg hela systemet först för positiva tal och komplettera sedan för negativa tal!



3) Listor – ordnad mängd av 0 eller flera element

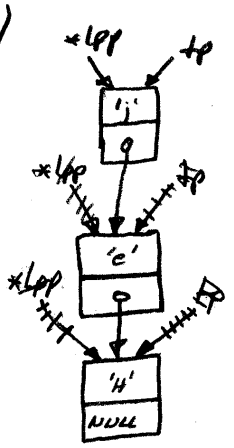
LIFO-lista (Last In First Out, stack)

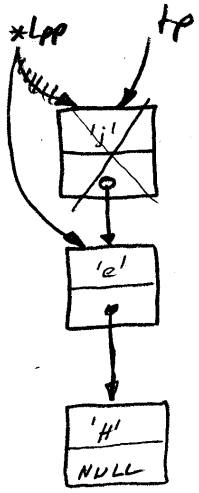
(Ex) Skapa en abstrakt datatyp för hantering av en generell stack. Använd sedan denna till att stacka alla tecken från en textrad och töm stacken och skriv ut texten.

```
/*Lifo.h */

typedef char datatyp; // Anpassa stacken
                      // till char (tecken)

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

void push(linktyp **lpp, datatyp d);
/* sätter d överst på stacken */

datatyp pop(linktyp **lpp);
/* Returnerar data från stackens topp */
```

anropas med adressen till en pekare!

```
/* Lifo.c */
#include "Lifo.h"
#include <stdlib.h>

void push(linktyp **lpp, datatyp d)
{
   Linktyp *tp;

   tp = malloc (sizeof (linktyp));
   tp->data = d;
   tp->next = *lpp; // NULL vid första
                    // anropet.
   *lpp = tp;
}

datatyp pop(linktyp **lpp)
{
   Linktyp *tp;
   datatyp d;

   tp = *lpp;
   d = tp->data;
   *lpp = tp->next;
   free (tp);
   tp = NULL;
   return d;
}
```

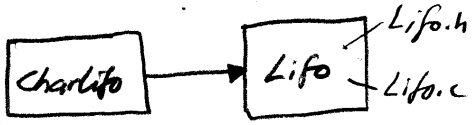




```
/*Charlifo.c*/

#include "Lifo.h"

void main()
{
    linktyp *Lp = NULL;          //OBS! NULL annars fungerar ej push och pop!
    char tecken;
    printf("Ge en textrad : ");
    tecken =getchar();
    while (tecken != '\n')
    {
        push(&Lp, tecken);       //OBS! Måste skicka adressen till pekare för att få den förändrad!
        tecken = getchar();
    }
    while(Lp != NULL)
    {
        putchar (pop(&Lp));      //OBS! Skriver tecknen med den sist inmatade först!
    }
    getch();
}
```

OBS! Om Hej inmatas fås utskriften jeH!

Beroendediagram :



FIFO-lista (First In First Out, kö)

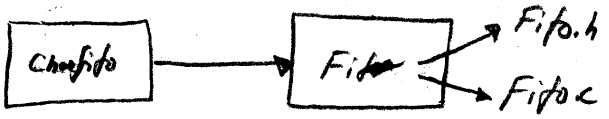
(Ex) Använd den abstrakta datatypen Fifo till att stoppa in tecken från en inläst sträng och töm kön och skriv ut tecknen.

```
/* Fifo.h */

typedef char datatyp; //Anpassa till aktuell datatyp

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

void infifo(linktyp **fpp, datatyp d);
/* Stoppar in d sist i kön */

datatyp utfifo(linktyp **fpp);
/* Returnerar första datat i kön */
```

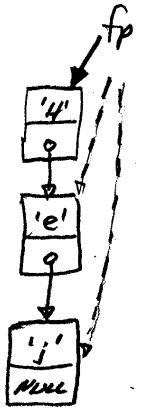
anropas med adressen till en pekare!

```
/* Fifo.c */
```

Se kompendiet!



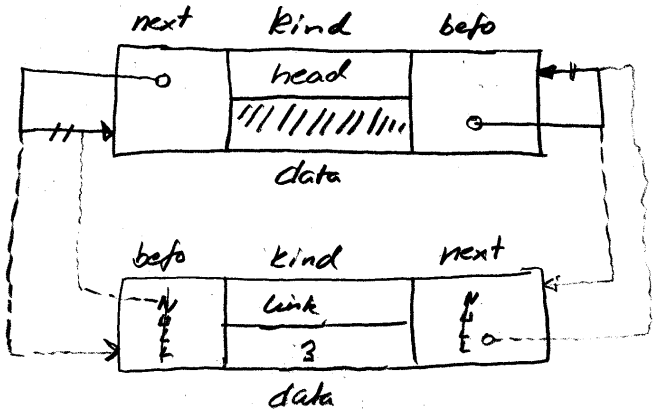

```
/* Charfifo.c */
#include "Fifo.h"

void main()
{
    linktyp *fp = NULL;
    char str[80];
    int i=0;
    printf("Ge en sträng :");
    geb(str);
    while( str[i] != '\0')
    {
        infifo(&fp, str[i]);
        i++;
    }
    while(fp != NULL)
    {
        printf("%c", utfifo(&fp));
    }
    getch();
}
```



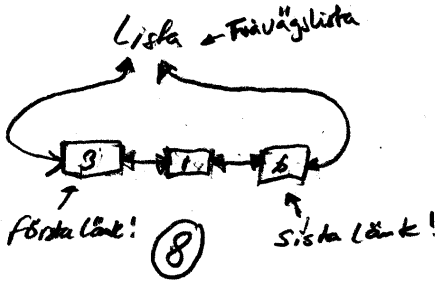
Generell lista där man kan stoppa in element var som helst och ta bort element hur som helst, förverkligas med en generell tvåvägslista.

(Ex) Slumpa ett inläst antal tärningskast och stoppa in dem i en tvåvägslista. Sök sedan igenom listan och skriv ut antalet ettor och sexor.

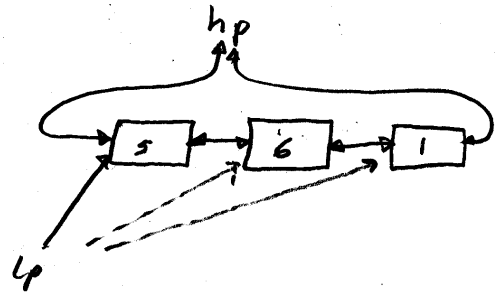

```
/* Twolist.h */

typedef int datatyp; //Anpassa till aktuell data

typedef struct twolink
{
    enum {head, link} kind;
    struct twolink *next, *befo;
    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);
/* Stoppar in d i länken */

datatyp getlink(linktyp *lp);
/* Returnerar data för länken */

void inlast(linktyp *lp, headtyp *hp);
/* Stoppar in länk i listan */

void infirst(linktyp *lp, headtyp *hp);
/* Stoppar in länken först i listan */
```

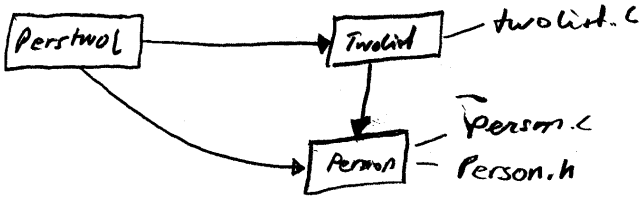


```
/* Twolist.c */
Se kompendiet!

/* Tarning.c */
#include "Twolist.h"
void main()
{
    headtyp *hp = NULL;
    linktyp *lp = NULL;
    int utfall, antal_6 = 0, antal_1 = 0, i, antal_kast;

    printf("Ge antal kast: ");
    scanf("%d", &antal_kast);

    newhead(&hp);
    srand((unsigned) time(NULL));
    for(i=1; i<= antal_kast; i++)
    {
        utfall = rand() % 6 + 1;
        newlink(&lp);
        putlink(utfall, lp);
        inlast(lp, hp);
    }
```




```
lp = firstlink(hp);
while (lp != NULL)
{
    utfall = getlink(lp);
    if (utfall == 1)
    {
        antal_1 ++;
    }
    else if (utfall == 6)
    {
        antal_6 ++;
    }
    lp = succlink(lp);
}
printf( "Antal ettor = %d", antal_1);
printf( "Antal sexor = %d", antal_6);
}
```

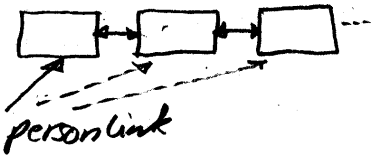
(Ex) Läs alla personer från binärfilen Person.dat till en tvåvägslista sorterad efter ålder och skriv ut alla personer på skärmen.




```
/*Twolist.h*/
#include "Person.h"   } Aktuell data
typedef person datatyp;
;
void insort(linktyp *lp, headtyp *hp,
                int (*is-less)(datatyp d1, datatyp d2));
!
!
```

```
/*Perstwol.c*/
void main()
{
    FILE *bsin;
    headtyp *personlista;
    linktyp *personlink;
    person p;
    bsin = fopen("Person.dat", "rb");
    newhead(&personlista);
```

```
fread(&p, sizeof(persm), 1, bsin);
while(!feof(bsin))
{
    newlink(&personlink);
    putlink(p, personlink);
    insort(personlink, personlista, synyrke_person);
    fread(&p, sizeof(persm), 1, bsin);
}
fclose(bsin);
personlink = firstlink(personlista);
while(personlink != NULL)
{
    p = getlink(personlink);
    skriv_persm(p);
    personlink = succlink(personlink);
}
clrmem(&personlista);
}
```



Hemuppgift: Läs inlämning 2, Oading
och försök att förstå upplägg