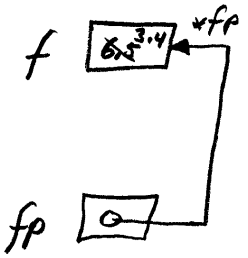
2) Pekare och dynamiska variabler

Pekare

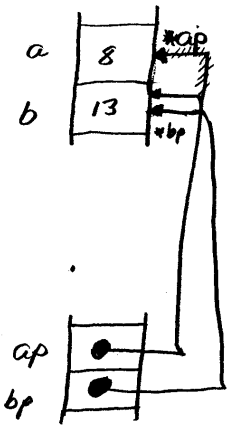
(Ex) :

```
float f;
f = 6.3;
:
float *fp; //Pekare
   Typ
fp = &f; //fp pekar på f
*fp = 3.4; //Gå dit fp pekar
```



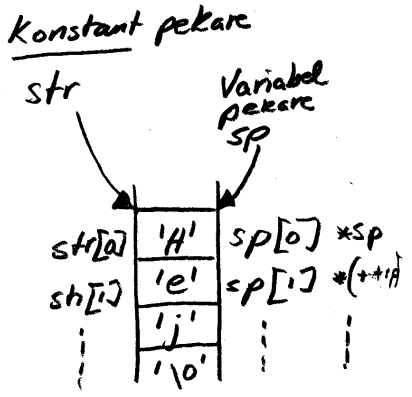
(Ex) :

```
int a=8, b=13;
int *ap=&a, *bp=&b;
Typ
:
*ap = *bp; //Tilldelning av heltal
:
ap = bp; //Tilldelning av adresser
         // eller omställning av
         // pekare.
```



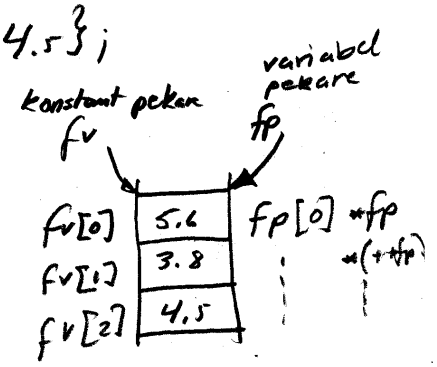
Ex

```
char str[4] = "Hej";
char *sp = str;
str[0] = 'h'; // H → h
sp[1] = 'E'; // e → E
```



Ex

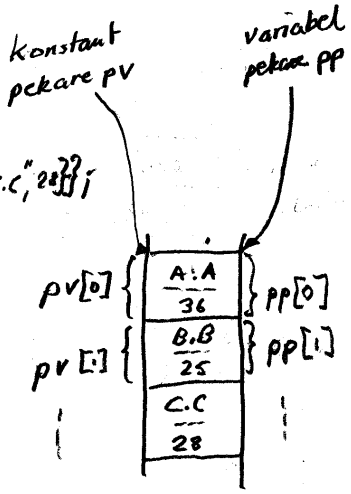
```
float fv[3] = {5.6, 3.8, 4.5};
float *fp = fv;
fv[0] = 3.2; // 5.6 → 3.2
fp[1] = 8.3; // 3.8 → 8.3
```



Ex

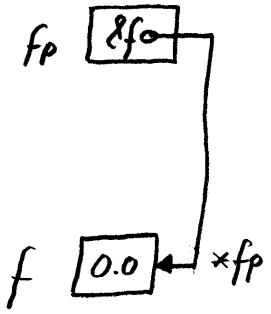
```
struct perspost
{
    char namn[30];
    int alder;
} pv[3] = {{"A.A", 36}, {"B.B", 25}, {"C.C", 28}};

struct perspost *pp = pv;
pv[0].alder++; // 36 → 37
pp[1].alder++; // 25 → 26
```



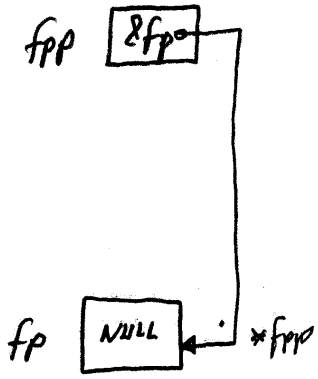
(Ex) a) Förändra ett parametervärde

```
void nolla(float *fp)
{
    *fp = 0.0
}
void main()
{
    float f;
    nolla(&f);
}
```



b) Förändra ett pekarparametervärde

```
void nulla(float **fpp)
{
    *fpp = NULL;
}
void main()
{
    float *fp;
    nulla(&fp);
}
```



Dynamiska variabler

(Ex) a) Vanlig automatisk eller statisk variabel.

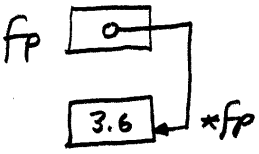
```
float f = 3.6;
```

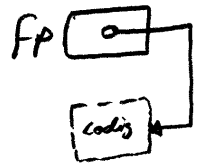
f [3.6]

b) Dynamisk variabel.

Två steg!

```
float *fp; //Steg 1
fp = malloc(sizeof(float)); //Steg 2
if (fp != NULL)
{
    *fp = 3.6;
    :
    free(fp);
    fp = NULL;
}
else
{
    printf("Kan ej allokera minne!");
}
```





(Ex) Textfilen Retal.txt innehåller ett antal reella tal. Först i filen står antalet tal. Skriv ett program som läser alla tal från filen till en dynamiskt allokerad vektor och skriver ut talen omvänt till en ny fil Nyretal.txt.

```
/* Filomv.c */
#include <stdio.h>
:
void main()
{
    FILE *tsin, *tsut;
    float *xv;
    int antal, i = 0;
    tsin = fopen("Retal.txt", "rt");
    fscanf(tsin, "%d", &antal);
    xv = calloc(antal, sizeof(float));
    while ( fscanf(tsin, "%f", &xv[i]) != EOF)
    {
        i++;
    }
    fclose(tsin);
    tsut = fopen("Nyretal.txt", "wt");
    for ( i = antal-1; i>=0; i--)
    {
        fprintf(tsut, "%f ", xv[i]);
    }
    free(xv);
    fclose(tsut);
}
```

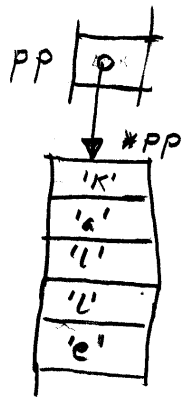
(Ex) Program som upprepat (avslut 0) läser in längden på ett namn, allokerar minne för namnet dynamiskt, läser in namnet och skriver ut namnets delar som åtskilts av blanktecken på var sin rad.

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

    printf("Ge längd: ");
    scanf("%d", &langd);
    while (langd > 0)
    {
        namn = calloc(langd+1, sizeof(char));   // obs! För nolltecknet!
        getchar(); // Tar bort \n från bufferten!
        printf("Ge namn: ");
        gets(namn);
        for(i=0; i<langd; i++)
        { if (namn[i] == ' ')
          {
            printf("\n");
          }
          else
          {
            printf("%c", namn[i]);
          }
        free(namn); // Annars äter programmet minne!
        // Läs ny längd!
```

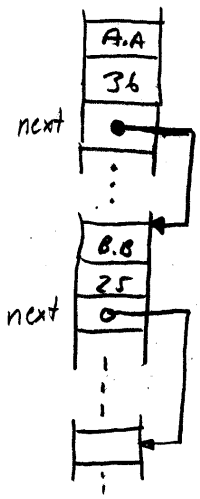
(Ex) Skapa en person dynamiskt, läs in och skriv ut.

```
#include "Person.h"
;
void main()
{
   person *pp;
   pp = malloc(sizeof(person));
   las-person(&*pp);
   free(pp);
   pp=NULL
```



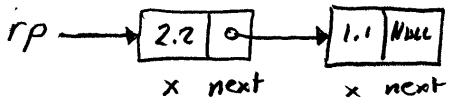
(Ex) Dynamisk kö av personer

```
;
struct personlink
{
    char namn[30];
    int alder;
    struct personlink *next;
};
;
```



Länkade strukturer

(Ex) Skapa den länkade strukturen



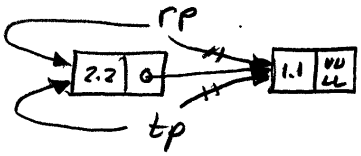

```
struct link
{
   float x;
   struct link *next;
} *rp = NULL, *tp = NULL;
```

Alt1)

```
rp = malloc(sizeof(struct link));
rp->x = 2.2;
rp->next = malloc(sizeof(struct link));
rp->next->x = 1.1;
rp->next->next = NULL;
```

Alt2) Börja bakifrån!

```
tp = malloc(sizeof(link));
tp->x = 1.1;
tp->next = NULL;
rp = tp;
```

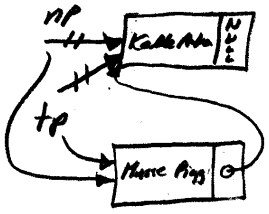


(Ex) Program som läser alla namn från textfilen Namn.txt och stoppar in namnen i en länkad struktur och sedan skriver ut namnen på skärmen så att

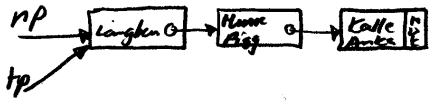
a) det sist inlästa namnet hamnar överst (stack)

```
:
void main()
{
    struct link
    {
       char namn[30];
       struct link *next;
    } *np=NULL, *tp=NULL;
    FILE *tsin;
    char str[30];

    tsin = fopen("Namn.txt","rt");
    while (fgets(str, sizeof(str))
    {
       tp = malloc(sizeof(struct link));
       strcpy(tp->namn, str);
       tp->next = np;
       np = tp;
    }
    fclose(tsin);
    tp = np;
```

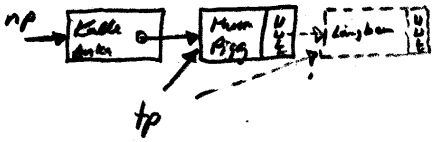


```
    while (tp != NULL)
    {
      printf("%s", tp->namn);
      tp = tp->next;
      free(np);
      np = tp;
    }
}
```



b) det sist inlästa namnet hamnar sist (kö)

```
    :
    while (fgets(str, sizeof(str))
    {
      if (np == NULL) // Första länken
      {
        np = malloc(sizeof(struct link));
        strcpy(np->namn, str);
        np->next = NULL;
      }
      else //Leta upp sista länk
      {
        tp = np;
        while (tp->next != NULL)
        {
          fp = tp->next;
        }
        tp->next = malloc(sizeof(struct link));
        strcpy(tp->next->namn, str);
        tp->next->next = NULL;
      }
      :
```

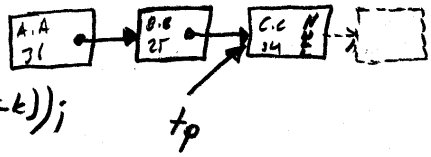


(Ex) I binärfilen Person.dat finns ett antal personer av typen person enligt Person.h. Läs alla personer från filen till en kö och skriv ut kön.

```
#
void main()
{
   struct link
   {
      person p;
      struct link *next;
   } *pp = NULL, *tp = NULL;
   person pers;
   FILE *bsin;

   bsin = fopen("Person.dat", "rb");
fread(&pers, sizeof(person), 1, bsin);
while (!feof(bsin))
{
   if (pp == NULL) // Första Länken
   {
      pp = malloc(sizeof(struct link));
      pp->p = pers;
      pp->next = NULL;
   }
   else
   {
```

```
        tp = pp;
        while (tp->next != NULL) //Leta upp sista länk
        {
            tp = tp->next;
        }
        tp = malloc(sizeof(struct link));
        tp->next->p = pers;
        tp->next->next = NULL;
    }
```




```
    fread(&pers, sizeof(person), 1, bsin);
  }
  fclose(bsin);
  tp = pp;
  while (tp != NULL)
  {
      skriv_person(tp->p);
      tp = tp->next;
  }
  getch()
}
```

Hemuppgift: Program som upprepat frågar efter längden på en stång, allokerar minne för stången och läser in stången som ska utgöra ett binärt tal ex. 1001. Efter inmatning av den binära stången ska det binära talets värde skrivas ut ex. 9. Glöm ej att avallokera minne.