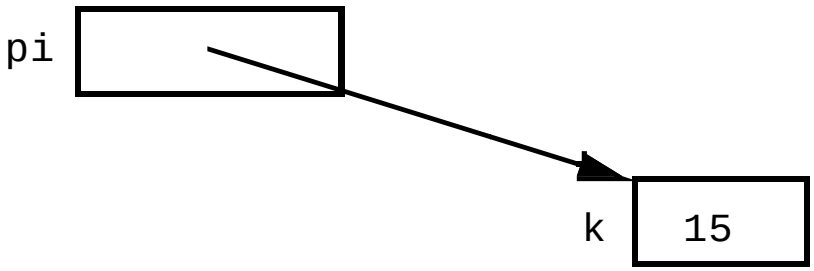
Arrayer (fält)

```
int a[6];
int a[6] = {2, 2, 2, 5, 5, 0};

int b[] = {0, 0, 0, 0, 0, 0};
int b[] = {i, i-1, i+j};

b = {0, 0, 0, 0, 0, 0}; // Otillåtet!

a = b; // Otillåtet!

for (int k=0; k<6; k++)  // istället
  a[k] = b[k];

if (a == b)  // Tillåtet, men inte vad man menar
```

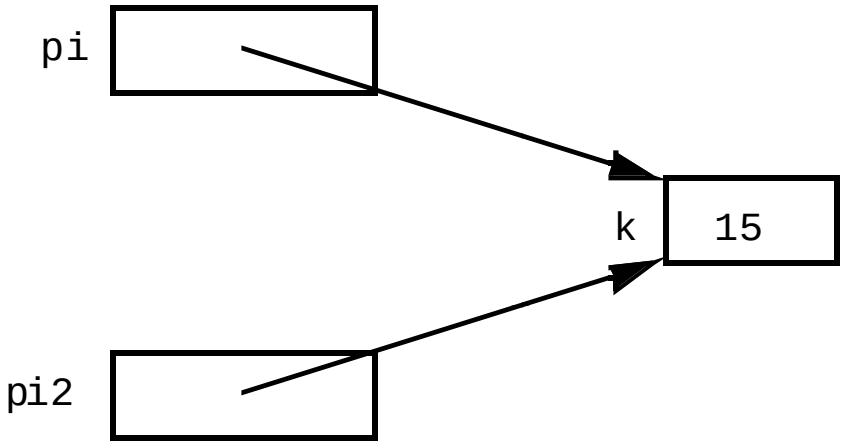
Pekare

```
int *pi;
double *pd;
char   *pc;
int k = 15;
pi = &k;
```




```
printf("%i",*pi);  // skriver ut 15
```

```
int *pi2;
pi2 = pi;
```




```
int k = 15, n;
int *pi, *pi2;
pi   = &k;
pi2  = &n;
*pi2 = *pi;      // samma som n = 15
```

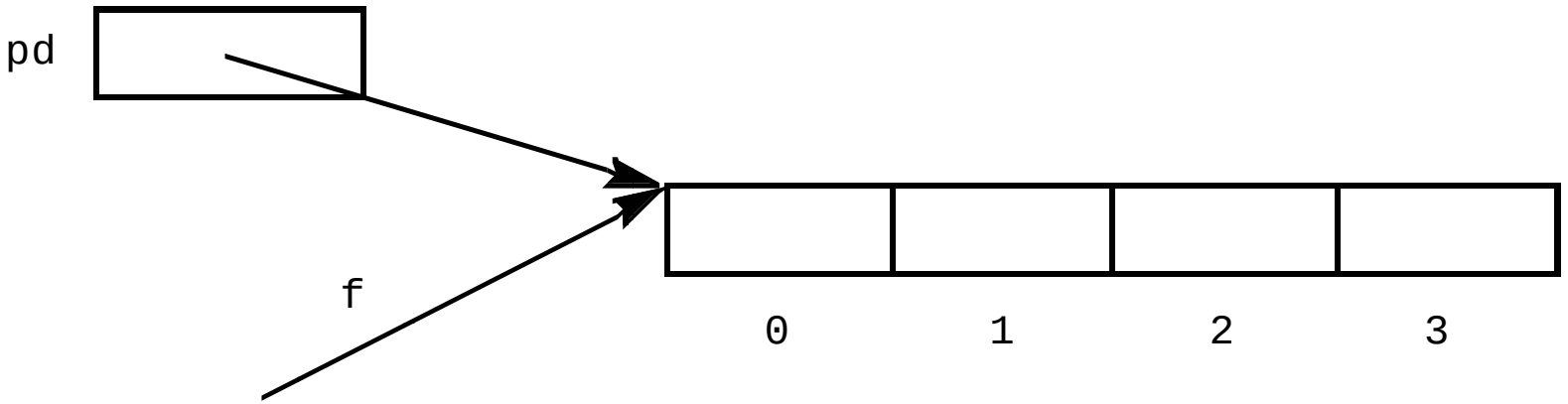
jämför med

```
pi2 = pi;
```

```
int *pi, *pi2;
int k=15, n, *pi=&k, *pi2=&n;

pi = NULL;   // rekommenderat skrivsätt, NULL def i stddef.h
pi = 0;      // går också
```

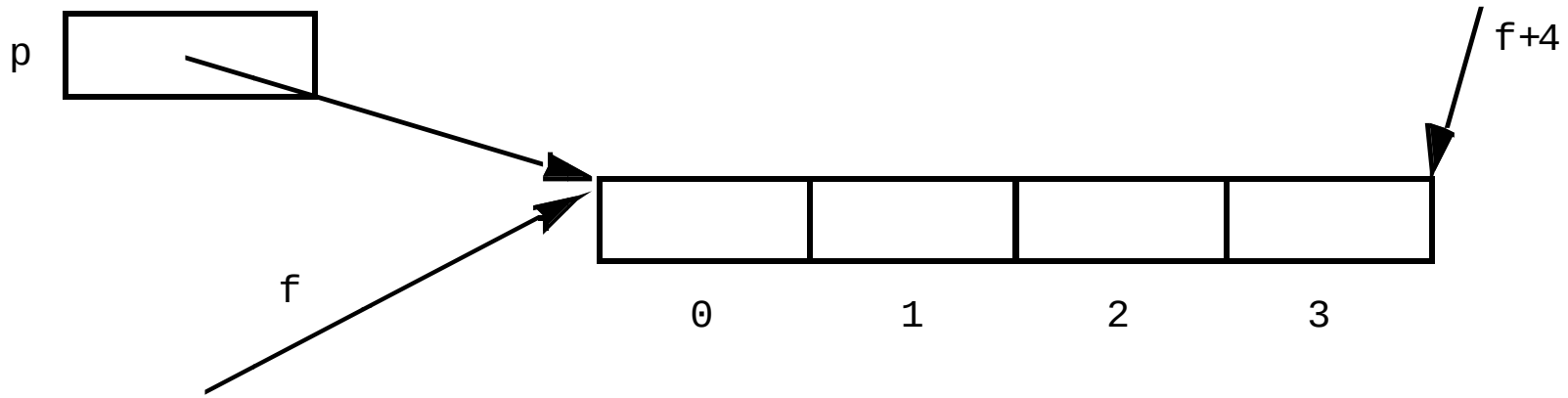
```
int k;            // ändringsbar int
const int c=100; // konstant int
const int *pc;   // pekare till konstant int
int *pi;          // pekare till ändringsbar int


pc = &c;   // OK
pc = &k;   // OK


pi  = &c;   // Fel! pi får inte peka på en konstant
*pc = 0;   // Fel! pc pekar på en konstant


int *const cp = &k;  // k kan ändras men inte cp
```

```
float f[4];
float *pd;
pd = &f[0];
pd = f;
```




```
printf("%f",*pd);          // skriver ut f[0]
printf("%f",*(pd+3));      // skriver ut f[3], pekararitmetik!!
```

```
for (int i=0; i < 4; i++)
  f[i] = 0;

for (float *p=f; p < f+4; p++)
  *p = 0;
```



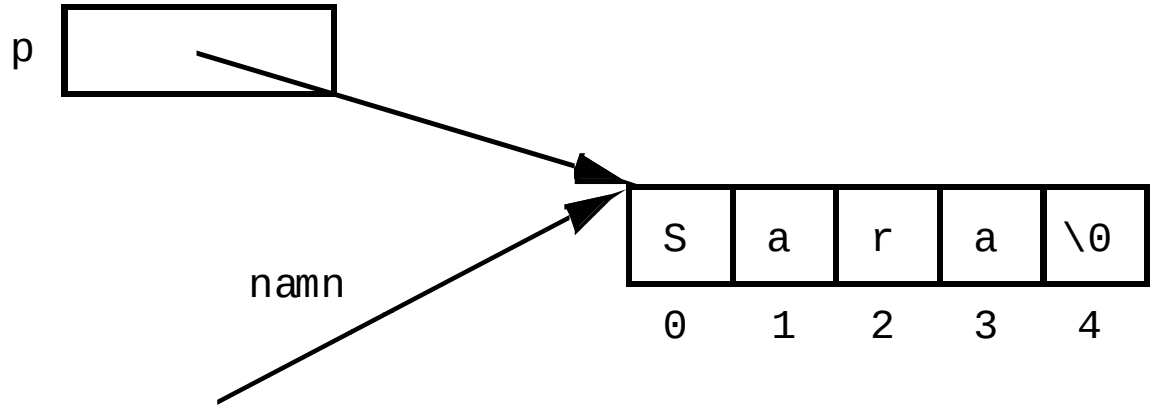
```
int b[] = {1,2,3,4,1,2,3,4,1,2};
printf("Summan är %f", sum(b,10));

int sum(int *a, int n)   // summerar elementen i a
{
   int s = 0;
   for (int *ip=a; ip < a+n; ip++)
     s += *ip;
   return s;
}
```

eller

```
int sum(int a[], int n)   // summerar elementen i a
{
 ...
}
```

Pekare och textsträngar

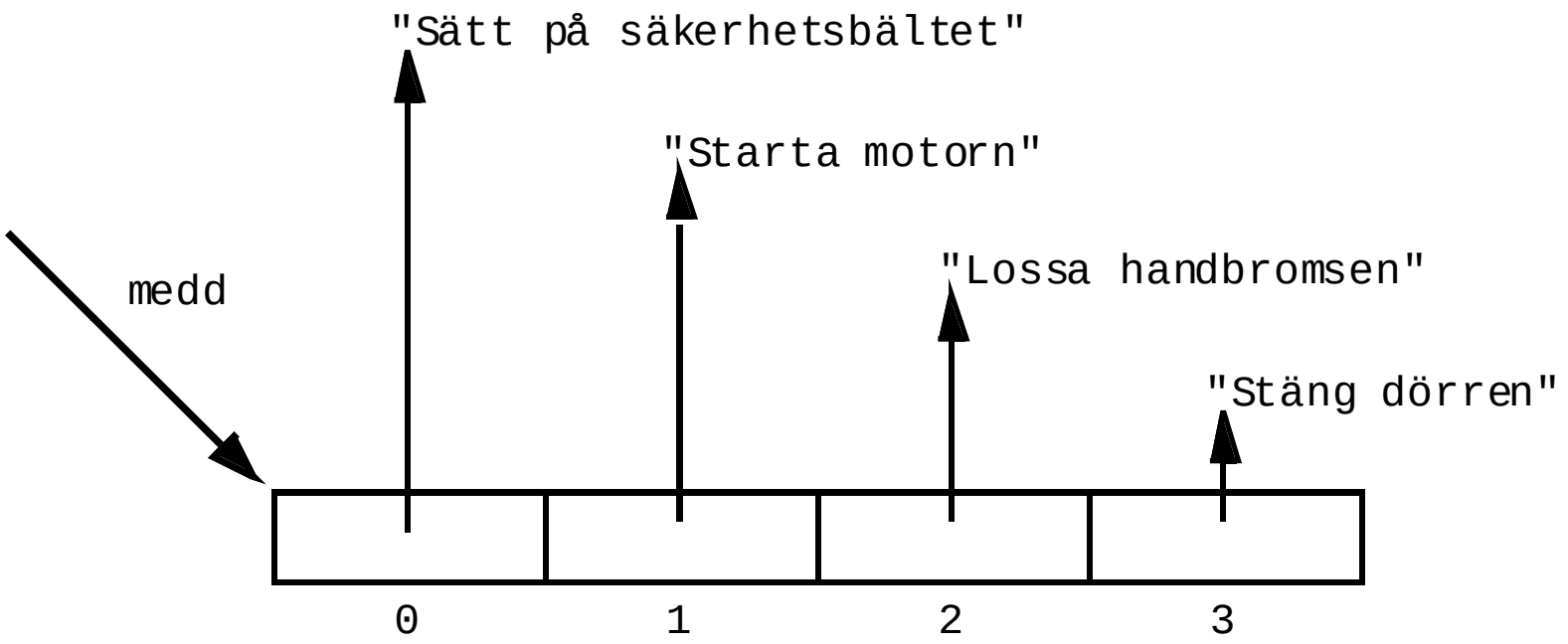
```
char txt[50];
char namn[] = "Sara";
char *p;
p = namn;
```




```
printf("%s", p);          // Sara skrivs ut
printf("%s", p+2);        // ra   skrivs ut
printf("%c", *(p+2));     // r    skrivs ut
```

```
char *p = "Hanna";
*(p+1)  = 'i';      // Tveksamt!!


int strlen (const char *s)   // ger längden av s
{
   const char *p=s;
   while (*p++)
     ;
   return p-1-s;
}



void strcpy (char *s1, const char *s2)
{
   while (*s1++ = *s2++)
     ;
}

strcpy(txt, q);           // "Hanna" kopieras till txt
strcpy(txt, namn);        // "Sara"  kopieras till txt
strcpy(txt+4," Maria");
printf("%s", txt);        // "Sara Maria" skrivs ut
```

```
const char *medd[4] =
          {"Sätt på säkerhetsbältet", "Starta motorn",
           "Lossa handbromsen",       "Stäng dörren"};
```



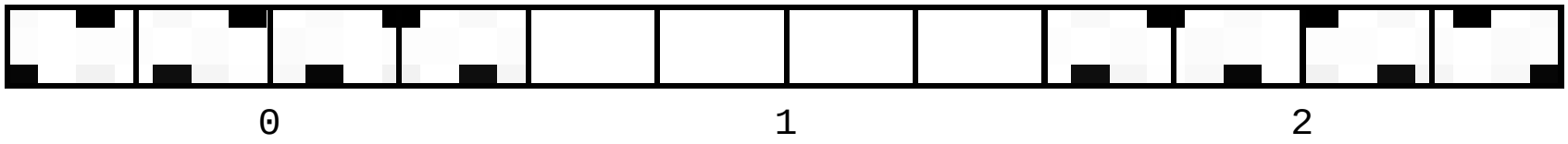

```
printf("%s", medd[i]);
```

```
void skriv(const char *pp[], int i)
{
   printf("%s", pp[i]);

}
```

eller

```
void skriv(const char **pp, int i)
{
   ...
}
```

```
int m[3][4];
```




```
float temp [4][12] =
   {{11.5, 12.0, 12.9, 13.5, 17.0, 20.5,
     22.2, 22.5, 21.5, 16.0, 13.5,  9.9},
    {19.0, 20.5, 21.0, 23.0, 27.0, 29.0,
     33.0, 35.0, 31.5, 27.0, 22.0, 20.0},
    {15.0, 15.5, 17.0, 19.0, 20.1, 24.0,
     27.0, 27.4, 26.5, 22.7, 17.5, 14.5},
    {17.0, 17.5, 20.5, 21.0, 21.5, 23.0,
     25.0, 26.0, 26.0, 22.7, 21.0, 17.5}};

m[1][2] = 14;

printf("%f", temp[2][4]);
```

```
void skriv (int a[][4], int rad_ant)
{
   for (int i=0; i < rad_ant; i++)
   {
     for (int j=0; j < 4; j++)
       printf("%i ", a[i][j]);
     printf("\n");
   }
}
```

```
void add (int a[][4], int b[][4], int c[][4], int n)
{
   for (int i=0; i < n; i++)
     for (int j=0; j < 4; j++)
       c[i][j] = a[i][j]+b[i][j];
}
```