

Funktioner

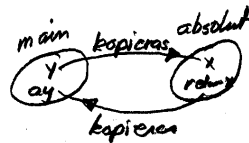
(Ex) Funktion som returnerar absolutbeloppet av ett reellt tal.

```
float absolut (float x)
{
    if (x < 0)
    {
        return -x;
    }
    return x;
}

void main()
{
    float y, ay;

    printf("Ge reellt tal: ");
    scanf("%f", &y);

    ay = absolut(y);
    printf("Absolutbeloppet av %f är %f", y, ay);
    getch();
}
```



(Ex) Funktion som returnerar sant (1) om de två sista siffrorna i ett heltal är lika annars falskt (0).

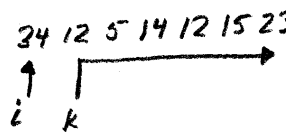
```
int lika(int tal)
{
    if (tal % 10 == tal / 10 % 10) // 3317 tal % 10 är 7
    {                               // tal / 10 % 10 är 7
        return 1;
    }
    return 0;
}

void main()
{
    int nr;
    srand((unsigned)time(NULL));
    nr = rand() % 999 + 100; // 100 - 999
    if (lika(nr))
    {
        printf("Lika!");
    }
    else
    {
        printf("Olika!");
    }
}
```

(Ex) Funktion som kontrollerar om alla element
i en heltalsvektor är olika

```
int olika(int v[], int nr)
{
    int i, k;
    for(i=0; i<nr-1; i++)
    {
        for(k=i+1; k<nr; k++)
        {
            if(v[k] == v[i])
            {
                return 0;
            }
        }
    }
    return 1;
}

void main()
{
    int lottorad[7], i;
    randomize();
    do
    {
        for(i=0; i<7; i++)
        {
            lottorad[i] = random(35)+1;
        }
    } while (!olika(lottorad, 7));
}
```

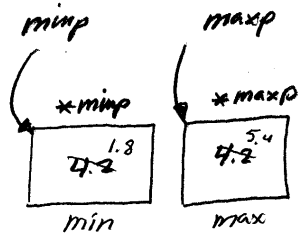


The diagram shows an array of numbers: 34, 12, 5, 14, 12, 15, 23. Below the first two numbers, '34' and '12', are the labels 'i' and 'k' respectively. An upward-pointing arrow is under 'i', and a horizontal arrow points from 'i' to 'k', illustrating the nested loop logic where each element is compared with all subsequent elements.

(Ex) Funktion som beräknar min och max för en reell vektor.

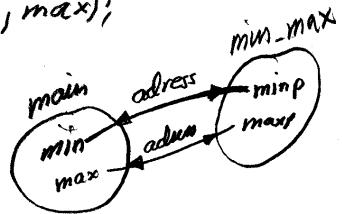
```
void min_max(float v[], int nr, float *minp, float *maxp)
```

```
{
    int i;
    *minp = *maxp = v[0];
    for(i=1; i<nr; i++)
    {
        if(v[i] < *minp)
        {
            *minp = v[i];
        }
        else if(v[i] > *maxp)
        {
            *maxp = v[i];
        }
    }
}
```



```
void main()
```

```
{
    float vek[4] = {4.2, 3.5, 5.4, 1.8}, min, max;
    min_max(vek, 4, &min, &max);
    printf("Min = %.1f\n", min);
    printf("Max = %.1f\n", max);
}
```



Ex) Funktion som returnerar antal tecken av angiven typ i en string.

```
int nr_ch(char *str, char ch)
{
    int i, sum=0;

    for(i=0; str[i]!='\0'; i++)
    {
        if(str[i]==ch)
        {
            sum++;
        }
    }
    return sum;
}

void main()
{
    char s[50], c;

    printf("Ge string (avslut RETURN): ");
    gets(s);

    while (s[0] != '\0') //ej tom string
    {
        printf("Vilket tecken? ");
        c=getchar();
        printf("Det finns %d %c i %s\n", nr_ch(s,c),
               getch(), c, s);
        /*Läs ny string*/
    }
}
```

Ex Funktion som returnerar antalet lika tecken i samma position hos två lika långa strängar.

```
int nr_lika(char *as, char *bs)
{
    int i, sum=0;
    for(i=0; as[i]!='\0' && bs[i]!='\0'; i++)
    {
        if (as[i] == bs[i])
        {
            sum++;
        }
    }
    return sum;
}

void main()
{
    char ratt_rad[14] = "x211xx12211x2",
        min_rad[14];
    printf("Ge din tipsrad: ");
    gets(min_rad);
    printf("Antal rätt = %d", nr_lika(ratt_rad,
        min_rad));
}
```

(Ex) Funktion som kontrollerar om det 6:te tecknet i nedanstående artikelnamn är en siffra.

```
struct artikel
{
    char namn[10];
    float pris;
};
```

```
int art-ok(struct artikel a)
```

```
{
    return a.namn[5] >= '0' && a.namn[5] <= '9';
}
```

```
void main()
```

```
{
    struct artikel art;
    printf("Ge artikelnamn: ");
    gets(art.namn);
    while (!art-ok(art))
    {
        printf("Fel artikelnamn!\n");
        printf("Ge artikelnamn: ");
        gets(art.namn);
    }
}
```

(Ex) Funktion som sätter pris på en artikel med hjälp av 6:te siffran i artikelnamnet enligt

<u>6:te siffra</u>	<u>Pris</u>
0-5	5.70
6,7	6.50
8,9	7.40

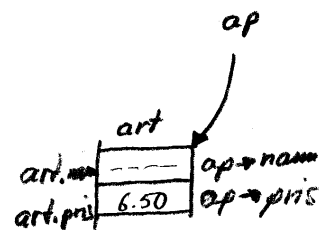
a) med adressparameter

```
void satt-pris (struct artikel *ap)
```

```
{
    if (ap->namn[5] >= '0' && ap->namn[5] <= '5')
        ap->pris = 5.70;
    else if (ap->namn[5] == '6' || ap->namn[5] == '7')
        ap->pris = 6.50;
    else
        ap->pris = 7.40;
}
```

```
void main()
```

```
{
    struct artikel art;
    printf("Ge artikelnamn: ");
    gets(art.namn);
    satt-pris(&art);
    printf("Pris = %.2f", art.pris);
}
```



b) med funktionsvärde

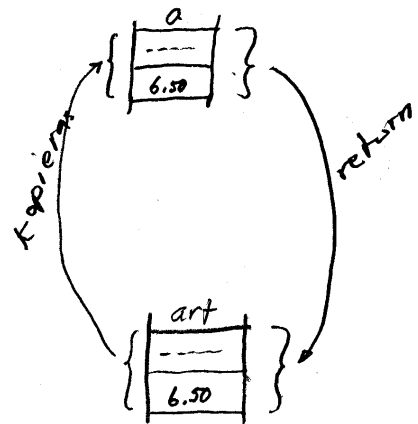
```
struct artikel get-pris(struct artikel a)
{
    if (a.namn[5] >='0' && a.namn[5] <='5')
        a.pris = 5.70;
    else if (a.namn[5] == '6' || a.namn[5] == '7')
        a.pris = 6.50;
    else
        a.pris = 7.40;
    return a;
}
```

```
void main()
```

```
{
    struct artikel art;

    printf("Ge artikelnamn : ");
    gets(art.namn);

    art = get-pris(art);
    printf("Pris = %.2f", art.pris);
    |
}
```



ⓔ I textfilen artiklar.txt finns ett antal gamla artikelnamn följt av nya enligt:

bgrasbusat	bgras7usat
afgur9usat	afgur8swet

Uppdatera alla artiklar i Lager.dat.

```
void main()
{
    int found=0;
    FILE *tsin, *bsinut;
    struct artikel gammal, ny, art;
    tsin = fopen("Artiklar.txt", "r");
    bsinut = fopen("Lager.dat", "r+b");
    while (fscanf(tsin, "%s", gammal.namn) != EOF)
    {
        fscanf(tsin, "%s", ny.namn);
        fread(&art, sizeof(struct artikel), 1, bsinut);
        while (!feof(bsinut) && !found)
        {
            if (strcmp(gammal.namn, art.namn) == 0)
            {
                found = 1;
            }
        }
    }
}
```