

2) Enkla datatyper

Program bearbetar data!



Kursen: röda tråda!

Variabel - typ, namn, värde och adress

Ex) Reellt tal

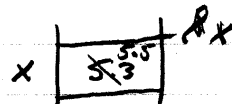
typ namn
float x;

x = 5.3; värde

x = x + 0.2;

↑
Nytt värde! Gammalt värde!

primärminne



printf("%7.1f", x); //uuuu5.5

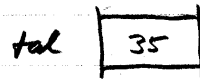
blanka, mellanrum

scanf("%f", &x);

adress

Ex) Helt tal initiering

int tal = 35;



printf("%5d", tal); //uuu35

scanf("%d", &tal);

①

(Ex) Tecken

```
char ch;  
ch = 'A';  
ch = ch + 1; // ch är B
```

ch

'A'

 eller då
ASCII för
ch

65

 A är 65

```
printf("%c %d", ch, ch); // A 65  
scanf("%c", &ch);
```

(Ex) Läs ett binärt tal och skriv ut det decimalt.

```
void main()  
{  
    char ch;  
    int tal = 0;  
  
    printf("Ge ett binärt tal:");  
    scanf("%c", &ch);  
    while (ch != '\n') // Avslut ENTER  
    {  
        tal = tal * 2 + ch - '0';  
        scanf("%c", &ch);  
    }  
    printf("Tallet = %d", tal);  
    getch();  
}
```

Aritmetiska uttryck - variabler + literaler
+ operatorer + funktioner

Ex) Skriv som C-sats $z = \sqrt{R^2 + (\omega L - 1/\omega C)^2}$

```
!
#include "Math.h" //sqrt, pow
```

```
!
float z, R, w, L, C
```

```
!
printf("Ge resistans: ");
scanf("%f", &R);
```

```
!
z = sqrt(pow(R, 2) + pow(w*L - 1/(w*C), 2));
```

Ex) Skissa på ett C-program som beräkna
summan, $1 + 1/2 + 1/3 + \dots + 1/100$.

```
!
float sum = 0.0;
int nr = 1;
```

```
while (nr <= 100)
```

```
{
```

```
    sum += 1.0/nr; //OBS! 1.0 för att få  
    nr++;          reell division
```

```
}
```

```
printf("Summan = %0.1f", sum);
```

③

Ex) Heltalskvot och rest

a) $15.0/6 = 2.5$ Reell division

$15/6 = 2$ Heltalsdivision (Antal helgänger!)

$15 \div 6 = 3$ Heltalsdivisionens rest $15 - 6 \cdot 2 = \textcircled{3}$ eller $15/6 = 2 \overset{\text{rest}}{\textcircled{3}}$

b) Månadsnr i datum 050124
fås som $050124 \% 10000 / 100$

Ex) Läs in ett sexsiffrigt telefonnummer och släp
ett tal mellan 0 och 9 och kontrollera
om det slumpade talet är samma som
slutsiffran i telefonnumret.

```
#include <stdlib.h> // Slump srand, rand  
#include <time.h> // srand använder  
klocka!
```

```
void main()
```

```
{
```

```
    int telnr, slump;
```

```
    printf("Ge ett sexsiffrigt telnr:");  
    scanf("%d", &telnr);
```

(Ex) "Ändra i programmet så att det upprepar inläsningen och rotberäkningen tills 0.0 inmatas.

Kör exempel

Ge ett positivt reellt tal (avslut 0.0): 1.44

Roten: 1.20

Ge ett positivt reellt tal (avslut 0.0): 49

Roten: 7.00

Ge ett positivt reellt tal (avslut 0.0): 0.0

/* Rotupp.c */

#include <stdio.h>

#include <math.h>

void main()

{

float tal, rot;

printf("Ge ett positivt reellt tal (avslut 0.0): ");

scanf("%f", &tal);

while (tal != 0.0) // så länge tal inte är 0.0

{

if (tal > 0.0)

{

(5)

```

    rot = sqrt(tal);
    printf("Roten: %0.2f\n", rot); // \n ny rad!
}
else
{
    printf("Kan ej beräkna roten av negativt tal!\n");
}
printf("Ge ett positivt reellt tal (avslut 0.0):");
scanf("%0f", &tal);
}
}

```

OBS! Uppreningen är en while-loop och villkoret för uppreningen kontrolleras alltid innan man går in i loopen och man måste läsa in nytt tal sist i loopen

//Läs tal!

```

while (tal != 0.0)
{

```

```

    |

```

```

    //Läs tal!
}

```

OBS! Nytt värde till tal måste läsas in annars har tal samma gamla värde och vi får en oändlig loop.

(6)

Logiska uttryck - aritmetiska uttryck +
relationsoperatörer +
logiska operatörer

ETT logiskt uttryck har värdet false (0)
eller true (1). Alla värden förutom 0
beaktas som sanna.

Ex a) Relationsoperatörer > < >= <= == !=

5 > 2 True, 1

5 < 2 False, 0

if (tal == 0) // True om tal 0

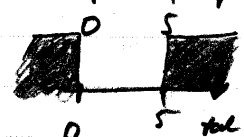
if (tal != 0) // True om tal ej 0

b) Logiska operatörer && || !
och eller inte

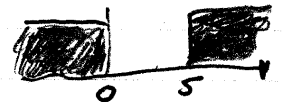
while (tal >= 0 && tal <= 5)



while (! (tal >= 0 && tal <= 5))



while (tal < 0 || tal > 5)



De Morgans $\neg (a \&\& b) \Leftrightarrow \neg a \ || \ \neg b$
 $\neg (a \ || \ b) \Leftrightarrow \neg a \ \&\& \ \neg b$

Ex) $!(tal \geq 0 \ \&\& \ tal \leq 5) \Leftrightarrow tal < 0 \ || \ tal > 5$

Upprepningsvillkor \leftrightarrow ! Avslutningsvillkor

while (tal \geq 0) // Avsluta då tal < 0

Ex) Sannstabell

<u>a</u>	<u>b</u>	<u>a && b</u>	<u>a b</u>	<u>!a</u>
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Ex) Program som läser in ett månadsnummer 1-12, säkert och skriver ut motsvarande månadsnamn.

```
#include <stdio.h>
#include <conio.h>
```

```
void main()
```

```
{
```

```
    int mon_nr = 0; // Fel från början
```

```
    printf("Ge månadsnr:");
```

```
    scanf("%d", &mon_nr);
```

8

Ett inmatnings
filter som ej
ger sig förtän
Korrekt mon-nr
1-12.

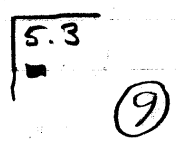
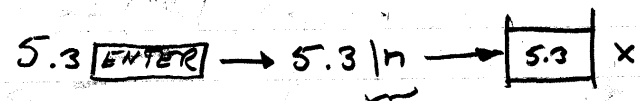
```
while (!(mon-nr >= 1 && mon-nr <= 12))
{
    printf("Fel månadsnr (1-12);  
fflush(stdin); // Tömmer inbufferten  
se nedan!  
printf("Ge månadsnr (1-12): ");  
scanf("%d", &mon-nr);
}
```

```
if (mon-nr == 1)
    printf("Jan!\n");
else if (mon-nr == 2)
    printf("Feb!");
...
else printf("Fel månad!");
!
```

In- och ut-matnings

In Tangentbord → Inbuffert(stdin) → Primärlinje
→ stora skär

Ex) a) scanf("%d", &x);



Tömmer bufferten men
blir ejäk kvar!

b) `scanf("%d", &tal);`

386 **ENTER** → `\n 386 \n` → `386` tal

5.3
386
-
Kvar från föregående lösning hoppas över!

c) `scanf("%c", &ch);`

A **ENTER** → `\n` ^{Hinner ej skriva något!} → `'\n'` ch

5.3
386
A
-

Rensa först bort `\n` från bufferten med `getchar()` eller `fflush(stdin)`.

A **ENTER** → `A\n` → `'A'` ch

5.3
386
A
-

Ut

Minne → utbuffet → skärm

ⓔx `printf("x = %8.2f \n", x);`

5.3 → `uuuu5.30` → `5.30`
-

(10)

b) `scanf("%d", &tal);`

386 **ENTER** → `\n 386 \n` → `386` tal

5.3
386
-
Kvar från föregående lösning hoppas över!

c) `scanf("%c", &ch);`

A **ENTER** → `\n` ^{Hinner ej skriva något!} → `'\n'` ch

5.3
386
A
-

Rensa först bort `\n` från bufferten med `getchar()` eller `fflush(stdin)`.

A **ENTER** → `A \n` → `'A'` ch

5.3
386
A
-

Ut

Minne → utbuffet → skärm

Ⓧ `printf("x = %8.2f \n", x);`

5.3 → `uuuu5.30` → `5.30`
-

(10)

(Ex) Skriv ett program som slumpar ett 7-stigt jokertal siffror för siffror och läser in ett annat jokertal och beräknar antalet rätt i bägge riktningar och skriver ut dem i ordning med största först.

Körlex!

Ge ditt jokertal : 3727441

Slumpat jokertal : 3757321

Antal rätt : 2

```
void main()
{
    int slump, antal, jokertal, mitt-jokertal;
    int delare, nr_fram, nr_bak;
    printf("Ge ditt jokertal : ");
    scanf("%d", &mitt-jokertal);

    srand((unsigned)time(NULL));
    jokertal = 0;
    antal = 1;
    while (antal <= 7)
    {
        slump = rand() % 10;
        jokertal = jokertal * 10 + slump;
        antal++;
    }
    printf("Slumpat jokertal : %d\n", jokertal);
}
```

(11)

```

delay = 1000000;
nr_fram = 0;
while (delay >= 1 && mitt_jokertal / delay ==
      jokertal / delay);
{
    nr_fram++;
    delay /= 10;
}

```

```

delay = 10;
nr_bak = 0;
while (delay <= 10000000 && mitt_jokertal % delay
      == jokertal % delay)
{
    nr_bak++;
    delay *= 10;
}

```

```

if (nr_fram > nr_bak)
{
    printf("Antal rätt fram : %d\n", nr_fram);
    printf("Antal rätt bak : %d\n", nr_bak);
}
else
{
    //tvärtom!
}

```

Hemuppgift: Skriv ett program som upptaget (avslut 0.0)
frågar efter antalet grader i Celsius och skriver
ut antalet grader Fahrenheit om $Fahrenheit =$
 $1.8 * Celsius + 32.$

(12)