

3) Styrning av programflöde

Med hjälp av de tre programstrukturerna sekvens, selektion och iteration styr man sitt programflöde så att önskad bearbetning fås.

Analys av programflöde

(Ex) Analysera programflödet för följande programkod genom att studera hur variabelernas värden förändras.

```
|
| tal = 6;
| nr = 0;
| while (tal >= nr)
| {
|   if (tal == nr)
|   {
|     tal--;
|   }
|   tal--;
|   nr++;
| }
|
```

Variabelvärden

tal	nr	
6	0	- start
5	1	
4	2	
3	3	
2	3	
1	4	- slut

Syntes av programflöde - skapa kod så att ett givet flöde av variabelvärden fås

①

Ex) Skriv den kod som från en rad av tecken skapar ett positivt heltal enligt:

Ge en rad av tecken : ab2c3gh5 ENTER

Talet är 235

```
void main()
{
    int tal=0;
    char tecken;

    printf("Ge en rad med tecken : ");
    scanf("%c", &tecken);
    while (tecken != '\n')
    {
        if (tecken >= '0' && tecken <= '9')
        {
            tal = tal * 10 + tecken - '0';
        }
        scanf("%c", &tecken);
    }
    printf("Talet är %d\n", tal);
}
```

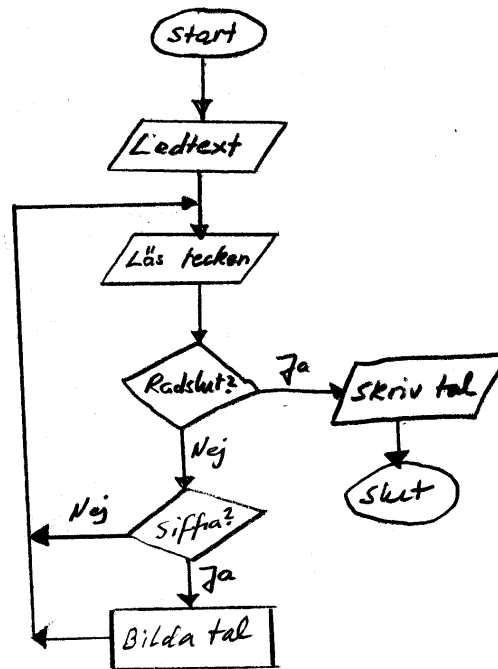
Om '2' så ASCII 48
är ASCII 50
50-48 är 2

Variabelvärdet

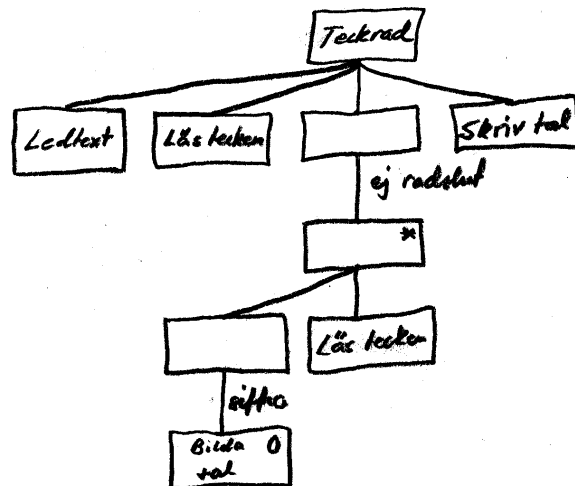
tecken 'a' 'b' '2' 'c' '3' 'g' 'h' '5'
tal 0 0 2 2 23 23 23 235

②

Flödesdiagram - följ pilarna och välj rätt väg!



Strukturdiagram - gå från vänster till höger och neråt om sant villkor!



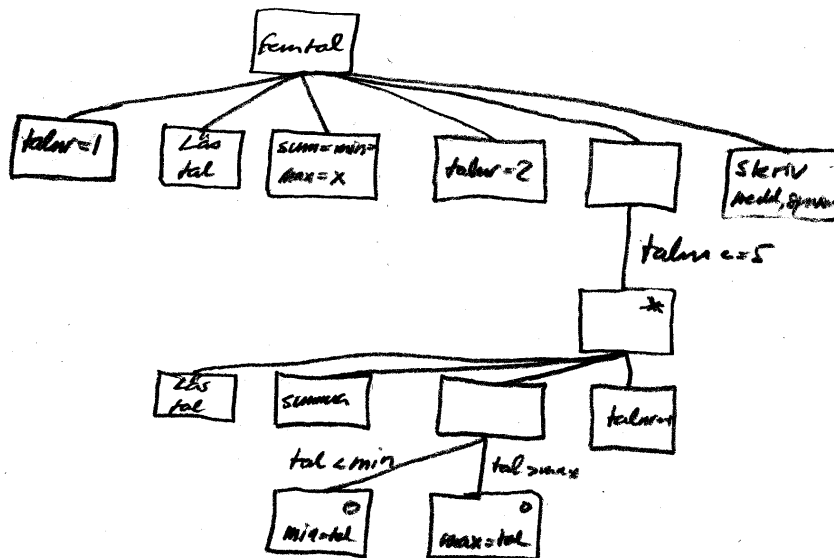
(3)

Ex) Skriv programkod som läser fem reella tal och skriver medelvärde och spridning (max - min).

Variabelvärden

x	3.4	3.2	3.3	3.5	3.1	
sum	3.4	6.6	9.9	13.4	16.5	} Medel 3.3 } Sprid 0.3
min	3.4	3.2	3.2	3.2	3.2	
max	3.4	3.4	3.4	3.5	3.5	

Strukturdiagram



Program

```
#include <stdio.h>
#include <conio.h>
void main()
{
    float tal, sum, min, max;
    int talnr = 1;

    printf("Ge tal %d", talnr);
    scanf("%f", &tal);
    sum = min = max = tal;

    talnr = 2;
    while (talnr <= 5)
    {
        printf("Ge tal %d", talnr);
        scanf("%f", &tal);

        sum += tal;
        if (tal < min)
        {
            min = tal;
        }
        else if (tal > max)
        {
            max = tal;
        }
        talnr++;
    }
    printf("Medel = %.1f \n Spridning = %.1f", sum/5, max-min,
    getch);
}
```

Alternativt
med
for-loop.
se nästa sida!

Alternativt med for-loop

```

|
|
| for (talnr = 2; talnr <= 5; talnr++)
| {
|     printf("Ge tal %d", talnr);
|     scanf("%d", &tal);
|     sum += tal;
|     if (tal < min)
|     {
|         min = tal;
|     }
|     else if (tal > max)
|     {
|         max = tal;
|     }
| }
|
|
|

```

for-Loopen

```

for (uttryck1; uttryck2; uttryck3)
{
|
|
}

```

↑ ↑ ↑
En gång loopvillkor sist i loopen
innan loop

och

while-Loopen

```

uttryck1;
while (uttryck2)
{
|
| uttryck3;
}

```

är ekvivalenta

⑥

(Ex) Program som läser in ett heltal och skriver ut talets multiplikationstabell.

Ge ett heltal: 7

0 * 7 = 0

1 * 7 = 7

⋮

9 * 7 = 63

⋮

void main()

{

int tal, mul;

printf("Ge ett heltal: ");

scanf("%d", &tal);

for(mul = 0; mul <= 9; mul++)

{

printf("%d * %d = %d", mul, tal, mul * tal);

}

⋮

(Ex) Program som upprepat visar en meny

R -- Read Data

W -- Write Data

E -- End

choose >> R

(7)

```

:
void main()
{
    char val;

    do
    {
        printf("R -- Read Data\n");
        printf("W -- Write Data\n");
        printf("E -- End\n");
        printf("Choose >> ");
        scanf("%c", &val);
        getchar();
        if (val == 'R' || val == 'r')
            /* Läs data */
        else if (val == 'W' || val == 'w')
            /* Skriv data */
        else if (val == 'E' || val == 'e')
            /* Hej då! */
        else
            /* Skriv felmeddelande */
    }
    while (! (val == 'E' || val == 'e'));
}

```

Alternativt
med
switch-
case se
nästa
sida!

OBS! Koden har ingen styrning av i vilken
ordning de olika alternativen körs. Vill man
att Read ska göras först kan man

⑧

använda en flagga $r=0$ och sedan inkapsla läsningen med $if(!r)\{\dots r=1;\}$. Write-alternativet inkapslas sedan med $if(r)\{\dots r=0;\}$

Alternativ med switch-case.

```
switch (val)
{
  case 'R':
  case 'r': /*Läs data*/
    break;
  case 'W':
  case 'w': /*Skriv data*/
    break;
  case 'E':
  case 'e': /*Hej då!*/
    break;
  default: /*Skriv felmeddelande*/
}
```

Använd

|| if ... else if ... else då selektionsvillkoren är intervall ex. $tal \geq 0$ && $tal \leq 5$

|| switch ... case då selektionsvillkoren är diskreta värden ex $tal = 0, 4, 8$.

(Ex) Program som läser ett månadsnummer och skriver ut månads antal dagar om nov, april, juni, sep har 30, feb har 28 och resten 31.

```
void main()
{
    int monr, dagar;
    printf("Ge månadsnummer 1-12 : ");
    scanf("%d", &monr);
    switch (monr)
    {
        case 11:
        case 4:
        case 6:
        case 9: dagar = 30;
                break;
        case 2: dagar = 28;
                break;
        default : dagar = 31;
                break;
    }
    printf("Antal dagar i månad %d är %d", monr,
        dagar);
}
```

ⓧ Program som slumpar en tipsrad
innehållande 13 matcher.

```
!
void main()
{
    int slump, matchnr;
    srand((unsigned)time(NULL));

    for(matchnr = 1; matchnr <= 13; matchnr++)
    {
        slump = rand() % 3;
        switch (slump)
        {
            case 0: printf(" x \n");
                    break;
            case 1: printf(" | \n");
                    break;
            case 2: printf(" 2 \n");
                    break;
        }
    }
}
```

ⓧ

Hemuppgift: Skriv ett program som
slumpar ett inläst antal
- Lottorader med 7 tal per
rad där talen ska ligga
mellan 1 och 35

Körex:

Ge antal Lottorader: 5

22	13	5	17	32	30	6
12	6	8	19	31	21	25
5	9	14	8	25	11	12
35	21	1	8	19	32	14
5	6	19	13	12	24	15

Du behöver ej tänka på
sortering eller att slumpa
ger två eller flera tal som är
lika på en och samma rad.
Sådana rader styker vi bara

(12)