

Detta är andra problemlösningsföreläsningen, vi diskuterar problemen ur Problem II.

Första problemet:

Frekvenstabell

Skriv ett program som slumpar ett tärningskast n gånger. Programmet skall därefter skriva ut en frekventabell över observationerna. (Layouten behöver inte se ut precis som i körexemplet nedan.)

Hur många tärningskast vill du göra? **60**

Värde	Frekvens
1	8
2	11
3	12
4	7
5	10
6	12

Hur har det gått för er? har ni löst problemet? De som inte har löst det, vilka svårigheter har ni stött på? De som har löst problemet vilka svårigheter har ni klarat av? Kan ni precisera hur ni har klarat av dessa svårigheter? De som inte har löst det, kan ni ta till er av hur man kan angripa problem i framtiden?

Vi ser på ett lösningsförslag:

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

/* Programmet simulerar n tärningskast, och skriver ut en
frekvenstabell */
void main(void)
{
    const int sidor=6;
    int i, j, n, slumptal, frekv[sidor+1]; /* frekv[0] används ej.
*/

    printf("Hur många kast med en %d-sidig tärning vill du göra?
",sidor);
    scanf("%d",&n);

    /* Frekvenstabellen nollställs. */
    for(i=1;i<=sidor;i++)
    {
        frekv[i]=0;
    }

    srand(time(0));
    for(j=1;j<=n;j++)
    {
        slumptal=(rand()%sidor)+1;
        (frekv[slumptal])++;
    }

    /* Utskrift av frekvenstabell. */
    printf("Värde   Frekvens \n");
    printf("----- \n");
    for(i=1;i<=sidor;i++)
    {
        printf("   %3d       %3d \n",i,frekv[i]);
    }

    getch();
}
```

Andra problemet:

Lottorad

Skriv ett program som slumpar fram en lottorad. En lottorad består av 7 olika heltal i intervallet $[1,35]$. (Det är inget krav att lottoraden skall skrivas ut sorterad, men det brukar bli resultatet av den "enklaste lösningen" på detta problem.)

Lottoraden är 5 13 19 20 23 28 35

Hur har det gått för er? har ni löst problemet? De som inte har löst det, vilka svårigheter har ni stött på? De som har löst problemet vilka svårigheter har ni klarat av? Kan ni precisera hur ni har klarat av dessa svårigheter? De som inte har löst det, kan ni ta till er av hur man kan angripa problem i framtiden?

Vi ser på ett lösningsförslag:

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

void main(void)
{
    /* dragen[i] är TRUE om boll nr i redan dragits. (Index 0
    används ej.) */
    bool dragen[36];

    int nr, i, antal=0;
    srand(time(0));

    /* Från början är inga bollar dragna */
    for(i=1;i<=35;i++)
    {
        dragen[i]=false;
    }

    /* Sju bollar skall dras */
    do
    {
        nr=(rand()%35)+1;

        /* Om bollen inte dragits tidigare är allt bra,
        i annat fall måste en ny boll dras. */
        if(dragen[nr]==false)
        {
            dragen[nr]=true;
            antal++;
        }
    }

    while(antal<7);

    /* Lottoraden skrivs ut, i storleksordning */
    printf("Veckans lottorad : ");
    for(i=1;i<=35;i++)
    {
        if(dragen[i]==true)
        {
            printf("%d ",i);
        }
    }

    getch();
}
```

Tredje problemet:

Sökning

Skriv ett program som slumpar 10 heltal i intervallet [1,20] och därefter frågar användaren vilket tal som söks.

Om talet finns med bland de slumpade, skall index till den första förekomsten av talet skrivas ut (det kan finnas dubletter). Om talet ej finns med bland de slumpade skall texten "Talet finns ej!" skrivas ut.

```
Slumptalen      : 3 10 7 12 4 7 18 13 1 4  
Vilket tal söks? 7  
Talet finns på index nr 2.
```

Hur har det gått för er? Har ni löst problemet? De som inte har löst det, vilka svårigheter har ni stött på? De som har löst problemet vilka svårigheter har ni klarat av? Kan ni precisera hur ni har klarat av dessa svårigheter? De som inte har löst det, kan ni ta till er av hur man kan angripa problem i framtiden?

Vi ser på ett lösningsförslag:

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

/* Programmet slumpar tal i intervallet [0..20] och lagrar dessa i
en array. Användaren söker ett tal;
om talet finns returneras index till första förekomsten. */

void main(void)
{
    const int antal=5;
    int m[antal], sokt, i;
    bool found=false;
    srand(time(0));

    /* Talen slumpas och skrivs ut */
    for(i=0;i<antal;i++)
    {
        m[i]=(rand()%20)+1;
        printf("%d  ",m[i]);
    }

    /* Sökningen genomförs */
    printf("Vilket tal eftersöks?");
    scanf("%d",&sokt);
    i=0;

    /* Det finns två skäl att avbryta sökningen:
    1) Vi har hittat det sökta talet.
    2) Det finns inga fler tal att kontrollera.  */
    do
    {
        if(m[i]==sokt)
        {
            found=true;
        }
        else i++;
    }
    while(found==false && i<antal);

    if(found==true)
        printf("Sökta talet finns på index %d",i);
    else
        printf("Sökta talet finns ej!");

    getch();
}
```