

1. Feladat

Egy osztályba n diák jár, akik m darab tantárgyat tanulnak. Ismerjük a félév végi osztályzataikat. Igaz-e, hogy minden diáknak van legalább két ötöse?

Specifikáció:

$A = \text{mátrix}(N) \times L$
 napló l

$B = \text{mátrix}(N)$
 napló'

$Q = (\text{napló} = \text{napló}')$

$\text{mátrix}(N) = N^{n \times m}$

$R = (Q \wedge l = \forall i \in [1..n] : \text{ötösdb}(i) \geq 2)$

ahol $\text{ötösdb}(i) = \sum_{j=1}^m \chi(\text{napló}[i,j]=5)$

Absztrakt program:

$l, i := \text{true}, 0$
$l \wedge i < n$
$i := i + 1$
$s := \text{ötösdb}(i)$
$l := s \geq 2$

$s := \text{ötösdb}(i)$

$s, j := 0, 1$			
$j \leq n$			
<table> <tr> <td>$\text{napló}[i,j] = 5$</td></tr> <tr> <td>$s := s + 1$</td></tr> <tr> <td>$SKIP$</td></tr> </table>	$\text{napló}[i,j] = 5$	$s := s + 1$	$SKIP$
$\text{napló}[i,j] = 5$			
$s := s + 1$			
$SKIP$			
$j := j + 1$			

Mátrix

2

A megoldóprogram:

```
#include <iostream>
#include <string>

void beolvas(int** &naplo, int &n, int &m,
             string* &tanulo, string* &targy);
bool mindenkinek_ket_otos(int** naplo, int n, int m);
int otosdb(int* v, int m);
void lezar(int** &naplo, int n, int m)

int main()
{
    int**      naplo;
    int        n,m;
    string*    tanulo;
    string*    targy;

    //Adatok beolvasása
    beolvas(naplo,n,m,tanulo,targy);

    //Kiértékelés és kiiratás
    if (mindenkinek_ket_otos(naplo,n,m)) {
        cout<<"Mindenkinek legalább két ötöse van az osztályban!";
    }
    else {
        cout<<"Van olyan az osztályban, akinek nincs két ötöse!";
    }

    //Lezárás
    lezar(naplo,n,m);

    cout<<endl;
    char kar;
    cin>>kar;
    return 0;
}

bool mindenkinek_ket_otos(int** naplo, int n, int m)
{
    bool l=true;
    int i=-1;
    while (l && i<n-1) {
        l=otosdb(naplo[++i],m)>=2;
    }
    return l;
}

int otosdb(int* v, int m)
{
    int db=0;
    for (int j=0; j<m; j++) {
        if (v[j]==5) db++;
    }
    return db;
}
```

```

}
void beolvas(int** &naplo, int &n, int &m,
              string* &tanulo, string* &targy)
{
    cout<<"Hány tanuló van?";
    cin>>n;
    tanulo=new string[n];
    cout<<"Adja meg a tanulók neveit:"<<endl;
    for(int i=0; i<n; i++) {
        cout<<"Tanuló neve: ";
        cin>>tanulo[i];
    }
    cout<<endl;

    cout<<"Hány tantárgy van?";
    cin>>m;
    targy=new string[m];
    cout<<"Adja meg a tantárgyakat:"<<endl;
    for(int j=0; j<m; j++) {
        cout<<"Tantárgy neve: ";
        cin>>targy[j];
    }
    cout<<endl;

    naplo=new int*[n];
    cout<<"Adja meg az osztályzatokat!"<<endl;
    for(int i=0; i<n; i++) {
        naplo[i]=new int[m];
        cout<<tanulo[i]<<" tanuló"<<endl;
        for(int j=0; j<m; j++) {
            cout<<"\t"<<targy[j]<<":";
            cin>>naplo[i][j];
        }
    }
    cout<<endl;
}

void kiir(int** &naplo, int &n, int &m)
{
    for(int i=0; i<n; i++) {
        delete[] naplo[i]
    }
    delete[] naplo;
}

```

2. Feladat

Szimuláljuk Galton deszkájának működését!

Specifikáció:

$$A = \begin{matrix} N & \times & N & \times & \text{vector}(N) \\ \text{szint} & db & v \end{matrix}$$

$$B = \begin{matrix} N & \times & N \\ \text{szint}' & db' \end{matrix}$$

$$Q = (\text{szint} = \text{szint}' \wedge db = db')$$

$$\text{vector}(N) = N_0^{\text{szint}-1}$$

$$R = (Q \wedge \forall i \in [0..db] : v[\text{jobbradb}(i)]\text{-t egygel növeljük})$$

$$\text{ahol } \text{jobbradb}(i) = \sum_{k=1}^{\text{szint}} \chi(\text{véletlen}(0,1)=1)$$

Absztrakt program:

$v, i := 0, 1$
$i \leq db$
$s := \text{jobbradb}(i)$
$v[s] := v[s] + 1$
$i := i + 1$

$s := \text{jobbradb}(i)$

$s, k := 0, 1$			
$k \leq n$			
<table> <tr> <td>$\text{véletlen}(0,1)=1$</td></tr> <tr> <td>$s := s + 1$</td></tr> <tr> <td>$SKIP$</td></tr> </table>	$\text{véletlen}(0,1)=1$	$s := s + 1$	$SKIP$
$\text{véletlen}(0,1)=1$			
$s := s + 1$			
$SKIP$			
$k := k + 1$			

A megoldóprogram:

```
#include <iostream>
#include <iomanip>
#include <stdlib.h>
#include <time.h>

int hany_jobbra(int szint);
void kiir(int* v, int szint);

int main()
{
    int* v;
    int szint,db;

    cout<<"Hány szintje legyen a Galton deszkának: ";
    cin>>szint;
    cout<<"Hány golyót engedjünk le: ";
    cin>>db;

    v=new int[szint+1];
    for(int i=0; i<=szint; v[i++]=0);

    srand(time(NULL));
    for(int i=0; i<db; i++) {
        ++v[hany_jobbra(szint)];
    }

    kiir(v,szint);
    delete[] v;

    return 0;
}

int hany_jobbra(int szint)
{
    int s=0;
    for(int k=1; k<=szint; k++)
    {
        if(rand()>RAND_MAX/2) s++;
    }
    return s;
}
```

```

void kiir(int* v, int szint)
{
    int max=v[0];
    for(int i=1; i<=szint; i++)
    {
        if (max<v[i]) max=v[i];
    }

    cout<<endl;
    for(int k=max; k>=0; k--)
    {
        for(int i=0; i<=szint; i++)
        {
            cout<<( v[i]>=k ? " * " : " ");
        }
        cout<<endl;
    }
    cout<<endl<<setw(3);
    for(int i=0; i<=szint; i++)
    {
        cout<<v[i]<<setw(5);
    }
    cout<<endl;

    char kar;
    cin>>kar;
}

```