

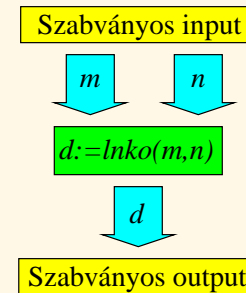
Elemi alkalmazások fejlesztése I

Feladat

Olvassunk be a szabványos inputról két pozitív számot, határozzuk meg a legnagyobb közös osztójukat és az eredményt írjuk ki a szabványos kimenetre.

Legnagyobb közös osztó számítás

Steingart Ferenc
stengi@inf.elte.hu



1.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

2.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Megoldás

- ➡ Állapottér
- ➡ Előfeltétel
- ➡ Absztrakt program
- ➡ Megoldóprogram

Állapottér

- m, n : beolvasott egész számok
- d : a legnagyobb közös osztó, egész szám
- c : segédváltozó

3.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

4.

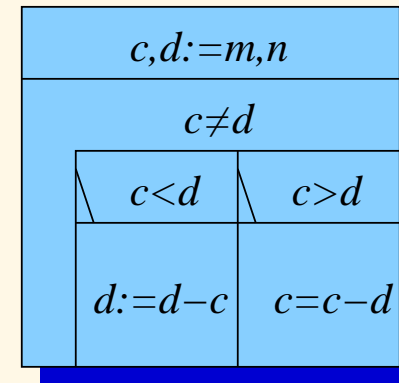
Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Előfeltétel

$$m > 0 \text{ és } n > 0$$

Absztrakt program



5.

Elemi alkalmazások fejlesztése I/2

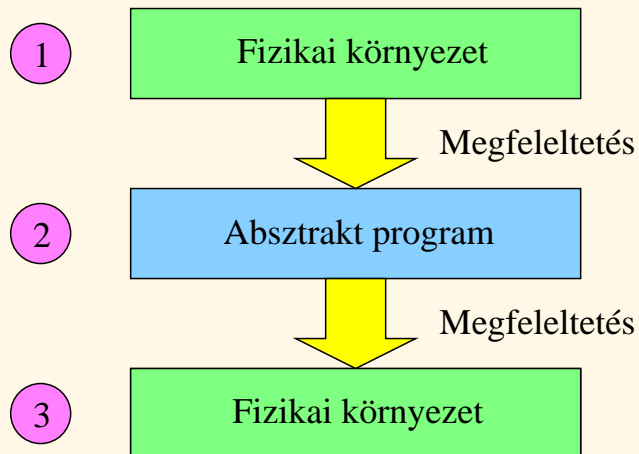
© Steingart Ferenc

6.

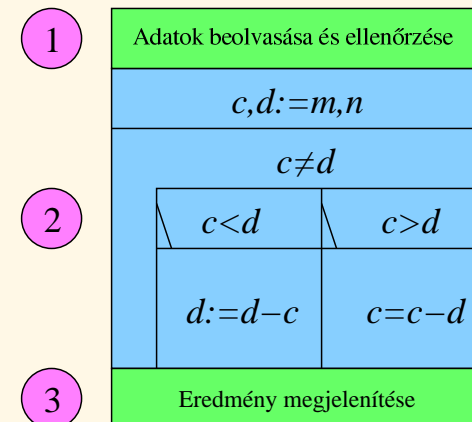
Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Megoldóprogram



Megoldóprogram



7.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

8.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Megoldóprogram: main

```
int main()
{
    // változók definíciója
    // adatok beolvasása és ellenőrzése
    // absztrakt program: lnko számítása
    // eredmény kiírása
    return 0;
}
```

Változók definíciója (állapottér)

```
int m, n, d, c;
```

9.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

10.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Alapvető típusok

`int, short, long, char, bool`
`unsigned int, unsigned char`

`double, float`

`string`

Adatok beolvasása

```
cout << "Legnagyobb közös osztó számítás" << endl;
cout << "Kérem az első számot:";
cin >> m;
cout << "Kérem a második számot:";
cin >> n;
```

11.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

12.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Szabványos I/O

szabványos **input** `cin` `stream`
szabványos **output** `cout` `stream`
szabványos **hiba** `cerr` `stream`

A fenti *streamek* deklarációját az
`iostream`
deklarációs (header) file tartalmazza

```
#include <iostream>
using namespace std;
```

13.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Szabványos I/O

input operátor: `>>`
`cin >> változó {>> változó};`

output operátor: `<<`
`cout << érték {<< érték};`

fontosabb implementált típusok:
`int, char, float, double, string`

fontosabb I/O manipulátorok:
`oct, dec, hex, endl, flush,`
`setbase(int), setfill(int), setprecision(int)`

14.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Deklarációs file betöltése

```
#include <iostream>
using namespace std;
int main()
{
    // változók definíciója
    // adatok beolvasása és ellenőrzése
    // absztrakt program: lnko számítása
    // eredmény kiírása
    return 0;
}
```

15.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Előfeltételek ellenőrzése

```
if(m <= 0 || n <= 0){
    cout << "Csak pozitív számokkal számolok!" << endl;
    return 0;
}
```

16.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Elágazás utasítás (if)

<i>feltétel 1</i>	<i>not(feltétel 1)</i>
<i>első ág</i>	<i>SKIP</i>

```
if(feltétel 1){  
    első ág  
}
```

17.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Elágazás utasítás (if)

<i>feltétel 1</i>	<i>not(feltétel 1) and feltétel 2</i>	<i>not(feltétel 1) and not(feltétel 2)</i>
<i>első ág</i>	<i>második ág</i>	<i>SKIP</i>

```
if(feltétel 1){  
    első ág  
}else if(feltétel 2){  
    második ág  
}
```

18.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Elágazás utasítás (if)

<i>feltétel 1</i>	<i>not(feltétel 1) and feltétel 2</i>	<i>not(feltétel 1) and not(feltétel 2)</i>
<i>első ág</i>	<i>második ág</i>	<i>egyébként ág</i>

```
if(feltétel 1){  
    első ág  
}else if(feltétel 2){  
    második ág  
}else{  
    egyébként ág  
}
```

19.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Elágazás utasítás (switch)

<i>változó = érték1</i>	<i>változó = érték2</i>	<i>...</i>	<i>egyébként</i>
<i>első ág</i>	<i>második ág</i>	<i>...</i>	<i>egyébként ág</i>

```
switch(változó){  
    case érték1:  
        első ág  
        break;  
    case érték2:  
        második ág  
        break;  
    case ...:  
        break;  
    default:  
        egyébként ág  
}
```

20.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Feltételek

Összehasonlító operátorok:

`==, !=, <, <=, >, >=`

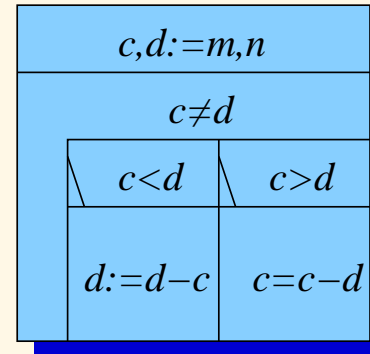
Logikai operátorok:

`!, &&, ||`

A `bool` típus értékei:

`true, false`

Az absztrakt program kódolása



```
c = m;
d = n;
while(c != d) {
    if(c < d) {
        d -= c;
    } else {
        c -= d;
    }
}
```

A while ciklus



```
while(feltétel) {
    ciklusmag
}
```

A for ciklus



```
for(utasítás1; feltétel; utasítás2) {
    ciklusmag
}

for(int i=0; i<10; ++i) {
    cout << i*i;
}
```

Aritmetikai operátorok

Kétooperandusú:	<code>+, -, *, /, %</code>
Egyoperandusú:	<code>-</code>
Kétooperandusú bitenkénti:	<code>>>, <<, &, </code>
Egyoperandusú bitenkénti:	<code>~</code>

25.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Önmódosító aritmetikai operátorok

<code>++i</code>	<code>i=i+1</code>
<code>--i</code>	<code>i=i-1</code>
<code>i+=a</code>	<code>i=i+a</code>
<code>i-=a</code>	<code>i=i-a</code>
<code>i*=a</code>	<code>i=i*a</code>
<code>i/=a</code>	<code>i=i/a</code>
<code>i%=a</code>	<code>i=i%a</code>

26.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Önmódosító bitenkénti operátorok

<code>i&=a</code>	<code>i=i&a</code>
<code>i =a</code>	<code>i=i a</code>
<code>i<=<a</code>	<code>i=i<<a</code>
<code>i>>=a</code>	<code>i=i>>a</code>

27.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

Az eredmény kiírása

```
cout << "LNKO = " << d << endl;
```

28.

Elemi alkalmazások fejlesztése I/2

© Steingart Ferenc

A program futása

```
Legnagyobb közös osztó számítás  
Kérem az első számot: 138  
Kérem a második számot: 52  
LNKO = 2
```

Összefoglalás

- ⇒ Főprogram írás
- ⇒ Ciklusok
- ⇒ Alapvető típusok
- ⇒ Összehasonlító operátorok
- ⇒ Szabványos IO
- ⇒ Logikai operátorok
- ⇒ Elágazások
- ⇒ Aritmetikai operátorok
- ⇒ Értékadás