





A PROGRAMOZÁS ALAPJAI 1

2. előadás

Vitéz András
egyetemi adjunktus
BME Híradástechnikai Tanszék
vitez@nit.bme.hu

2012. február 14.

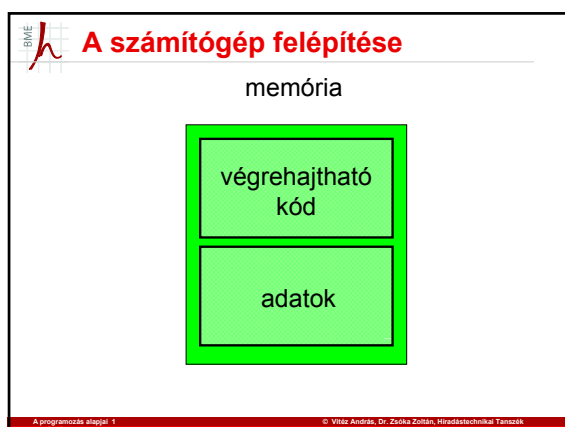
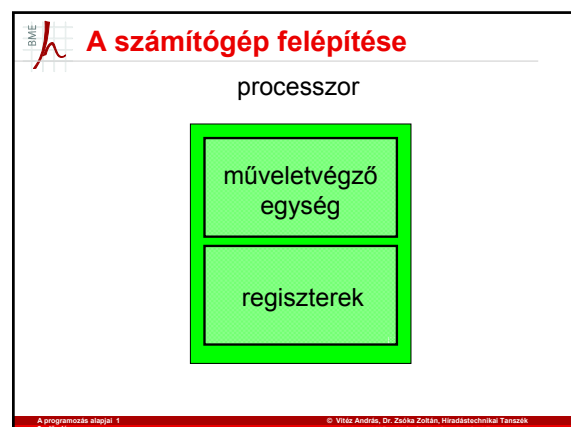
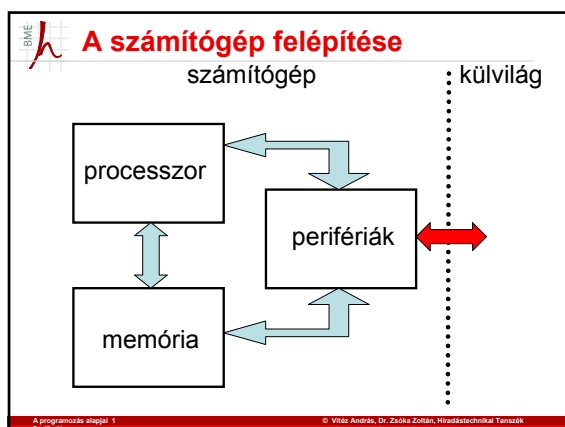


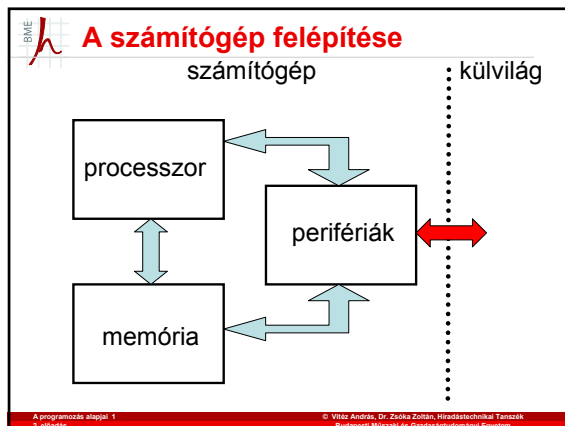
Miről lesz ma szó?

- A számítógép felépítése
- Algoritmus fogalma
- Programozási nyelvek osztályozása
- Programok építőkövei
- A strukturált programozás
- Minimális követelmények
- A C nyelv alaki szabályai
- A C nyelv utasításai
- Példa utasítások beágyazására

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem





Algoritmus fogalma

Algoritmus:

gépiesen (gondolkodás nélkül)
végrehajtható lépések véges sorozata,
amely elvezet a megoldáshoz

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Algoritmus fogalma

Módszer (eljárás):

gépiesen (gondolkodás nélkül)
végrehajtható lépések sorozata,
amely elvezet a megoldáshoz

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Algoritmus fogalma

Mielőtt az algoritmust kódolni kezdjük,
meg kell győződni róla, hogy

helyes, teljes és véges

Valóban azt a feladatot oldja meg, amire szánjuk.
Minden lehetséges esetben megoldja.
Véges sok lépésben befejeződik.

Nem elég kipróbálni, bizonyítani kell!

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Algoritmus fogalma

Példa:

Egy szám köbgyökét úgy
kapjuk, ha elosztjuk 4-gyel.

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Algoritmus fogalma

Próba:

$$8 : 4 = 2$$

$$2 \cdot 2 \cdot 2 = 8$$

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Algoritmus fogalma

1 kísérlet nem kísérlet!

$$-8 : 4 = -2$$

$$-2 \cdot -2 \cdot -2 = -8$$

A programozás alapjai 1
9 előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Rudolfi Miklós és Károlyi Juditoktól Emlékező Alapítvány



Algoritmus fogalma

Na még egyet!

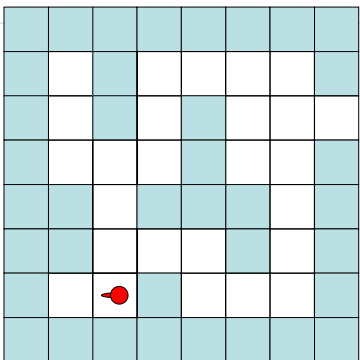
$$12 : 4 = 3$$

$$3 \cdot 3 \cdot 3 = 27$$

A programozás alapjai 1
9 előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Rudolfi Miklós és Károlyi Juditoktól Emlékező Alapítvány

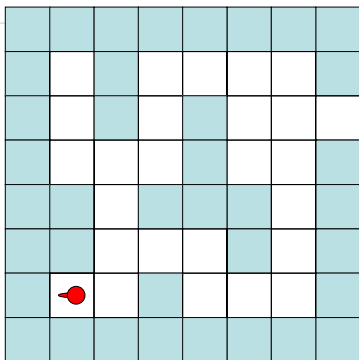




A programozás alapjai 1
2 előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Rudolfi Miklós és Károlyi Juditoktól Emlékező Alapítvány

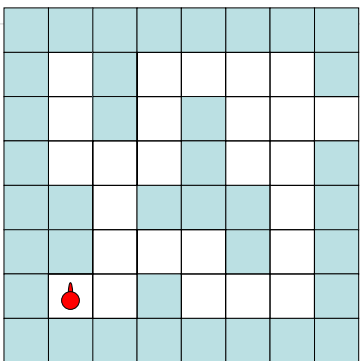




A programozás alapjai 1
2 előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Rudolfi Miklós és Károlyi Juditoktól Emlékező Alapítvány

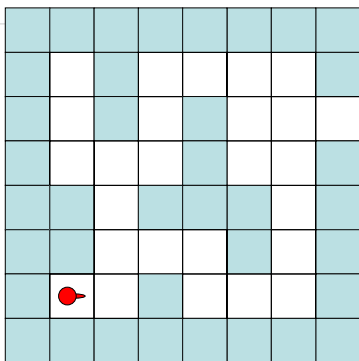




A programozás alapjai 1
2 előadás

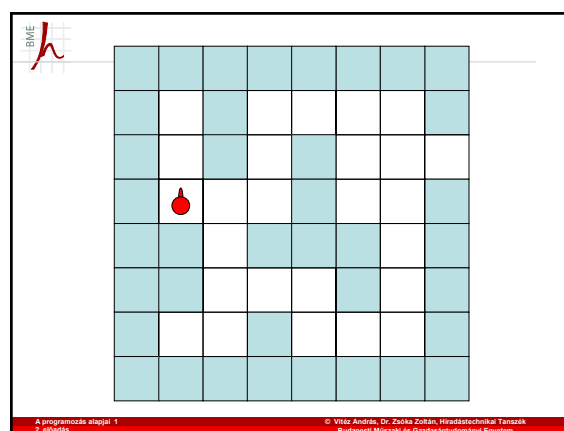
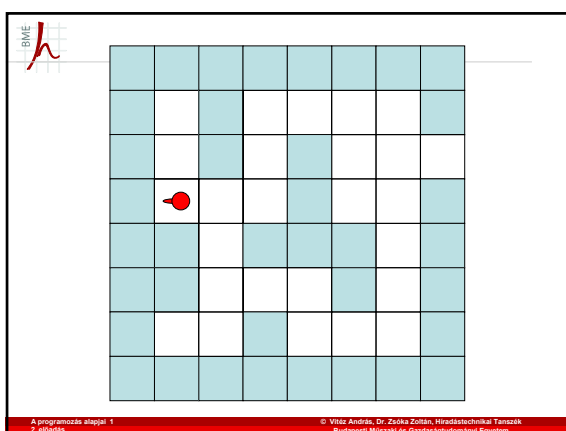
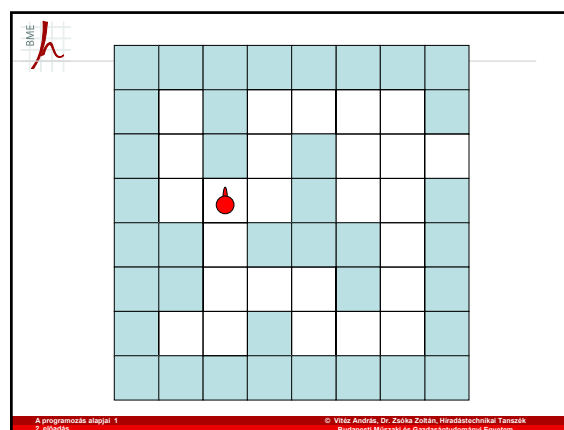
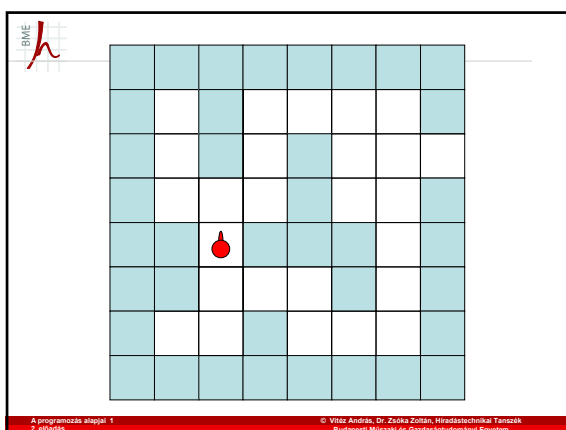
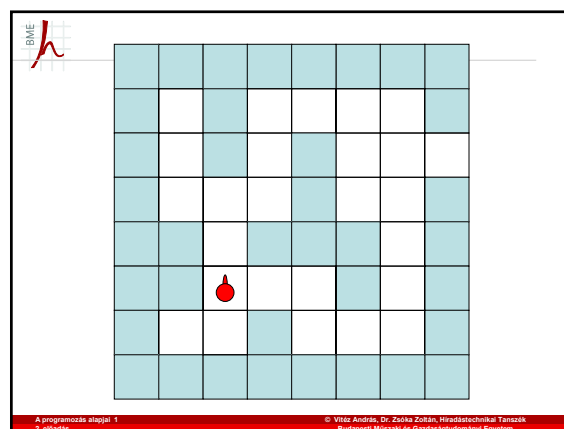
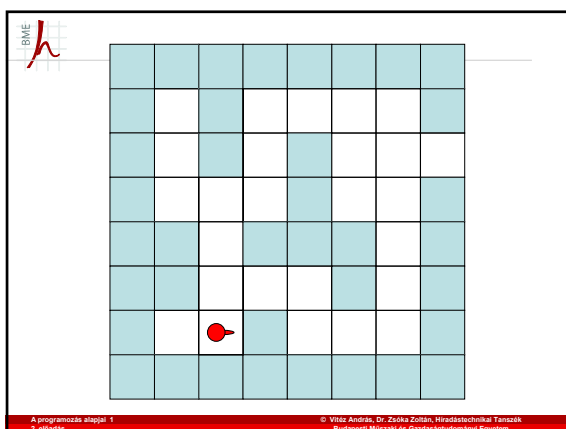
© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Rudolfi Miklós és Károlyi Juditoktól Emlékező Alapítvány

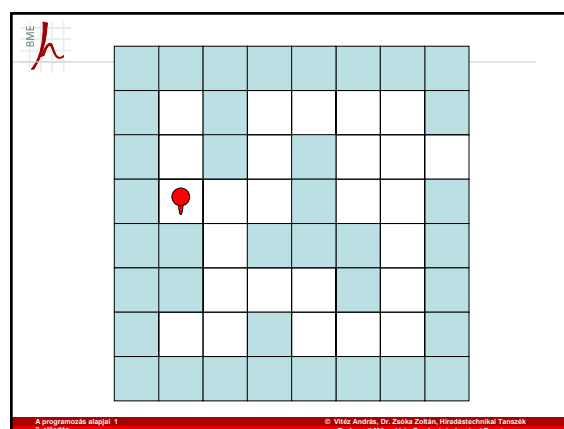
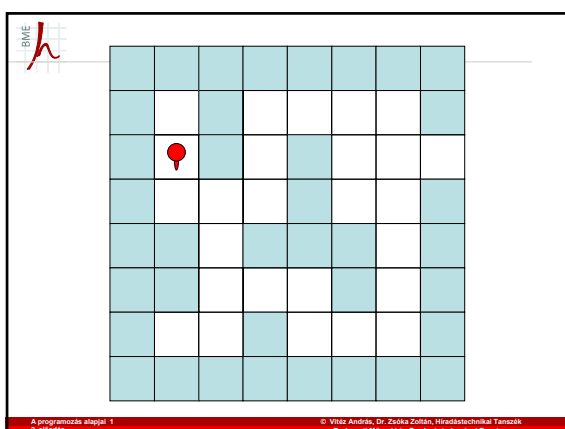
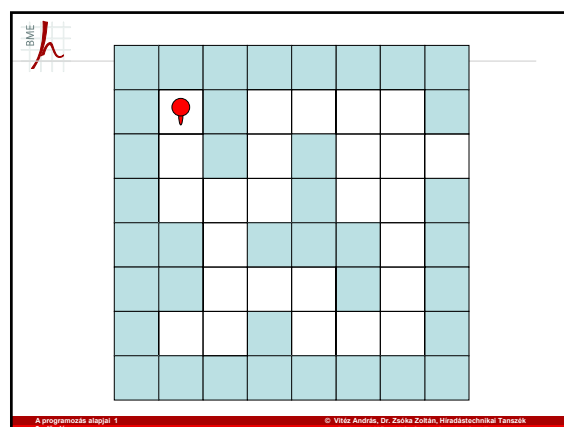
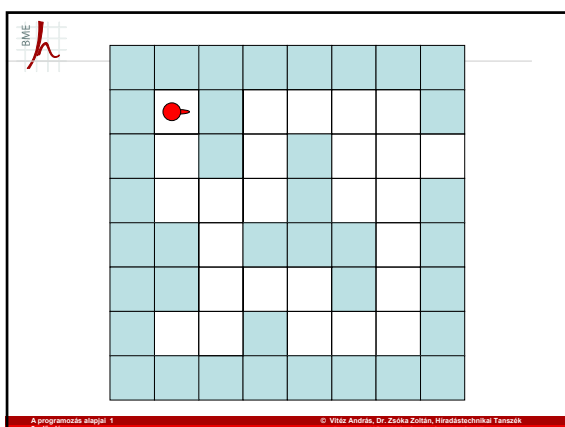
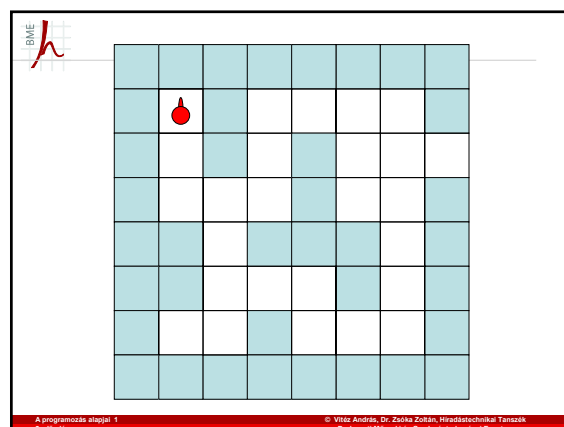
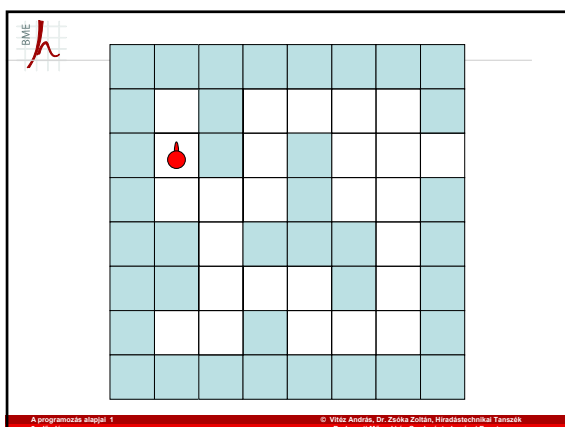


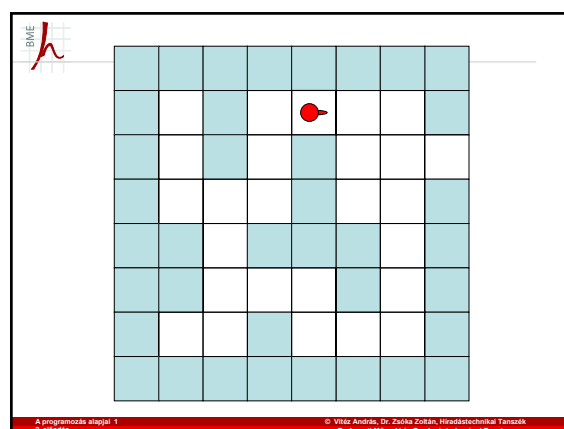
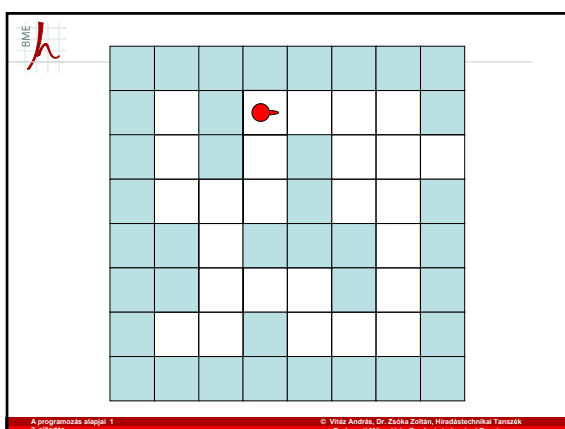
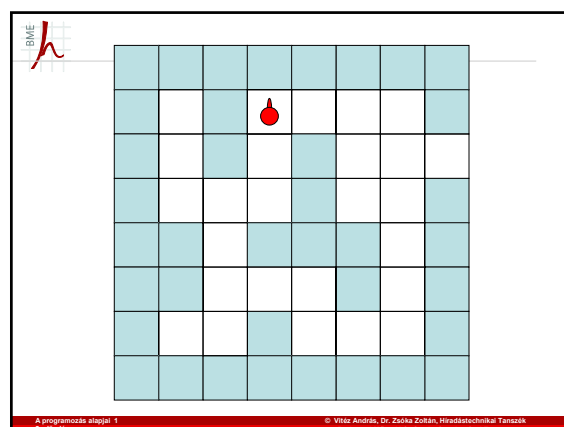
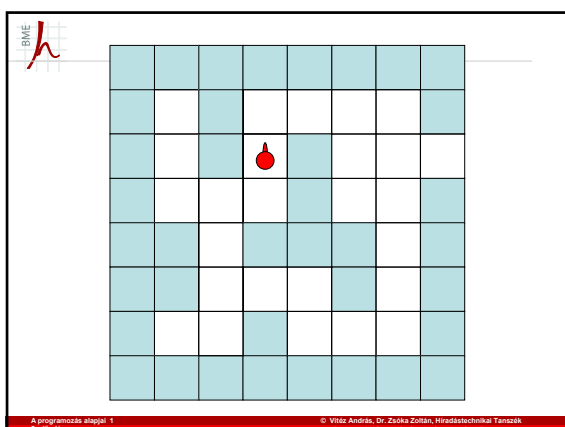
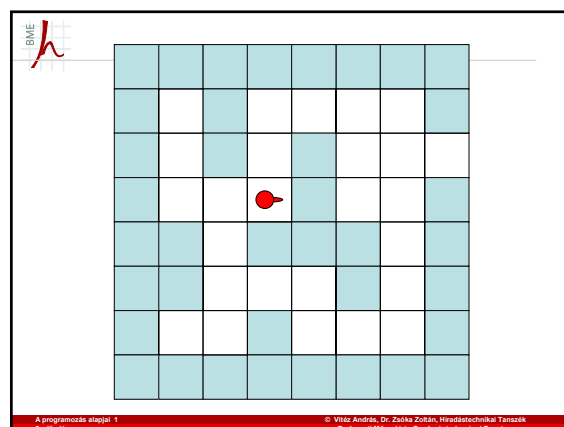
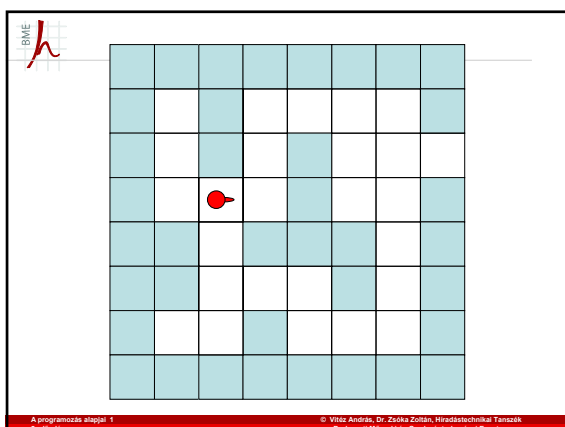


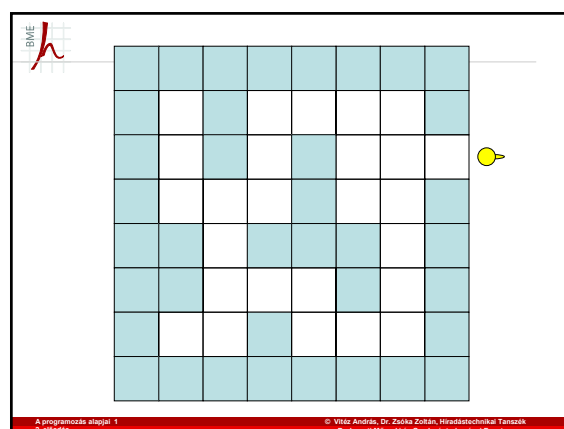
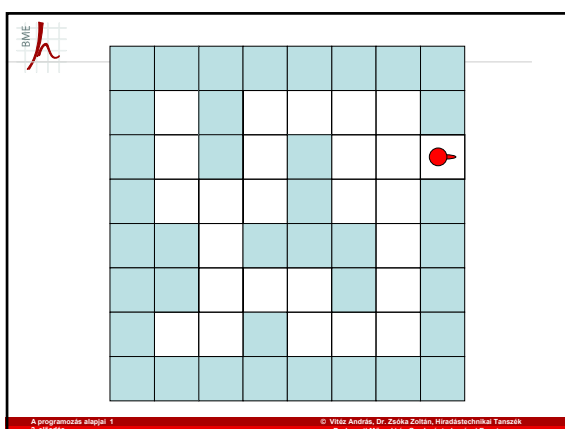
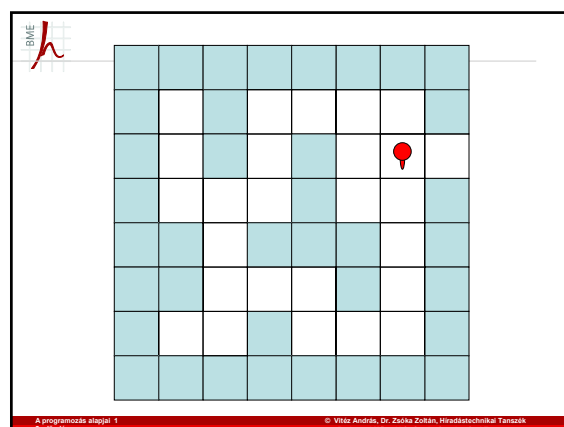
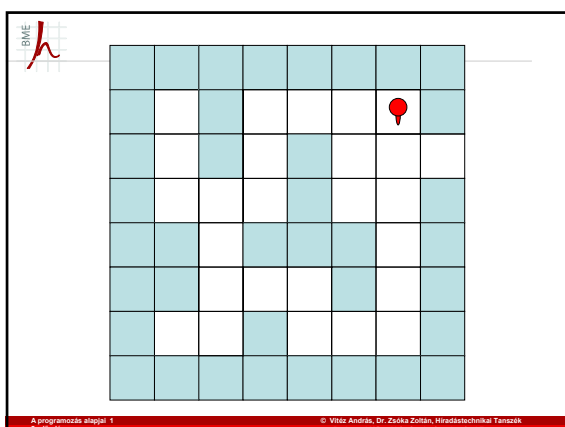
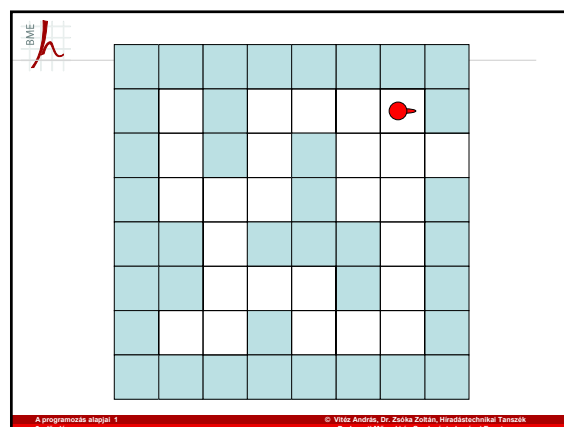
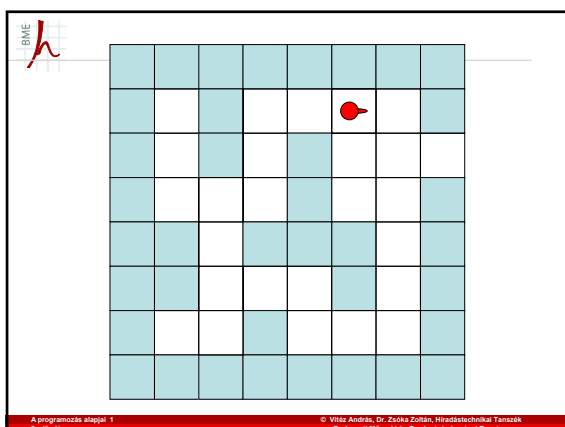
A programozás alapjai 1
2 előadás

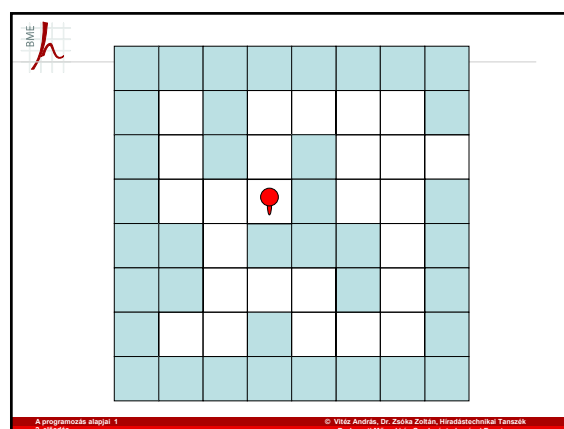
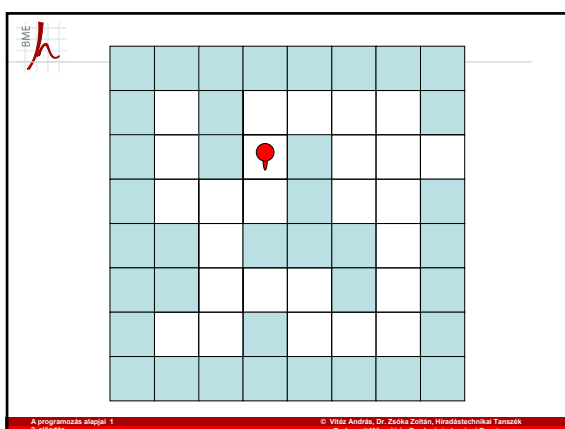
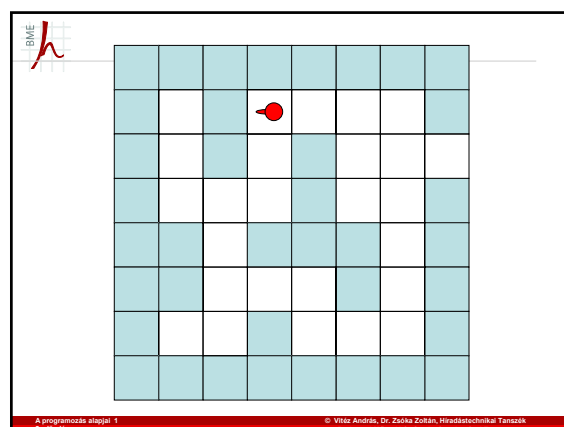
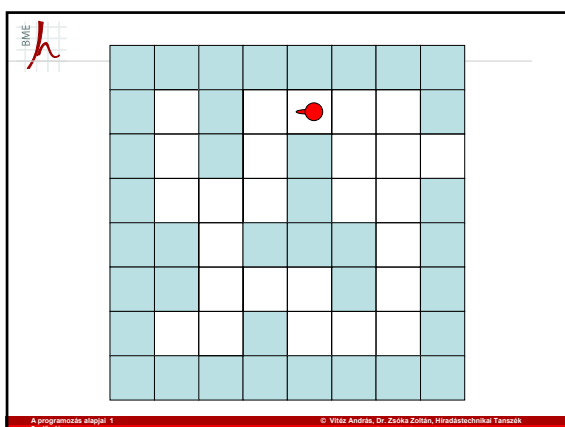
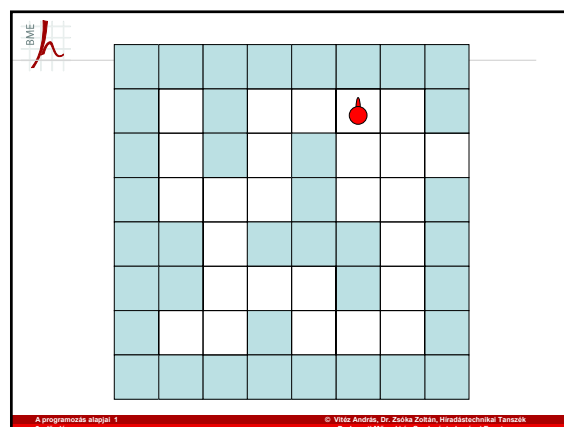
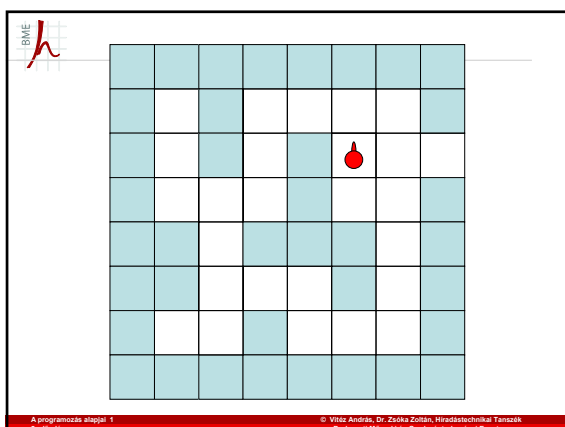
© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Rudolfi Miklós és Károlyi Juditoktól Emlékező Alapítvány

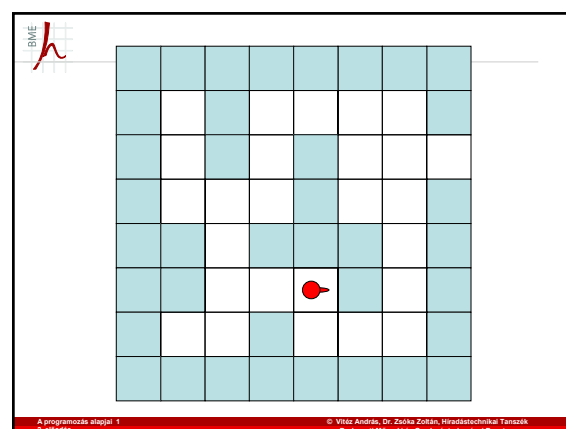
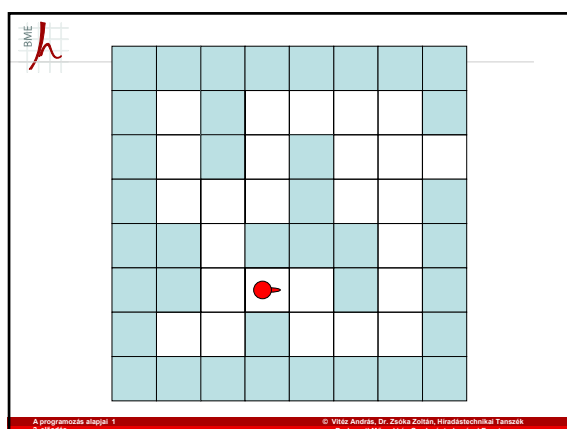
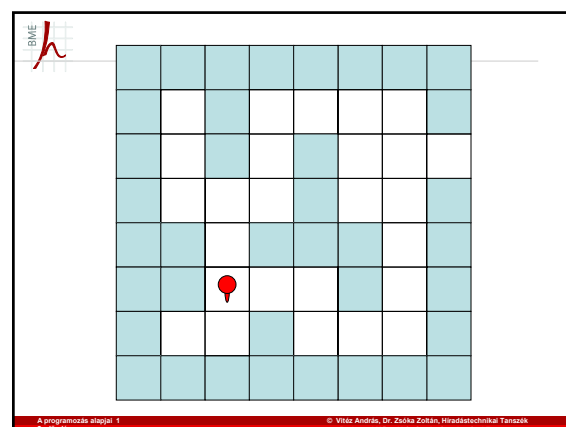
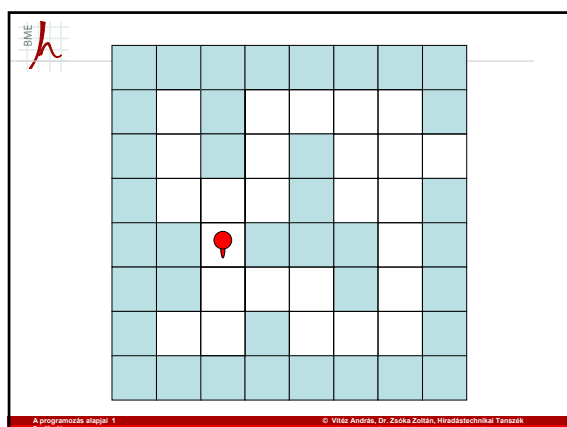
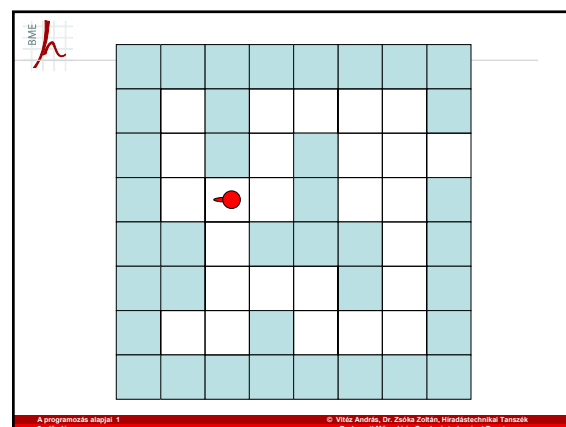
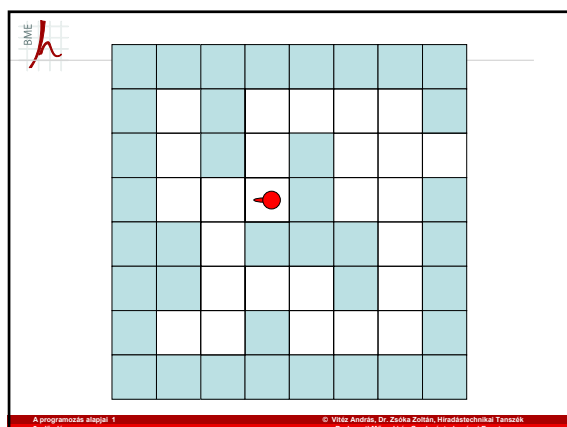


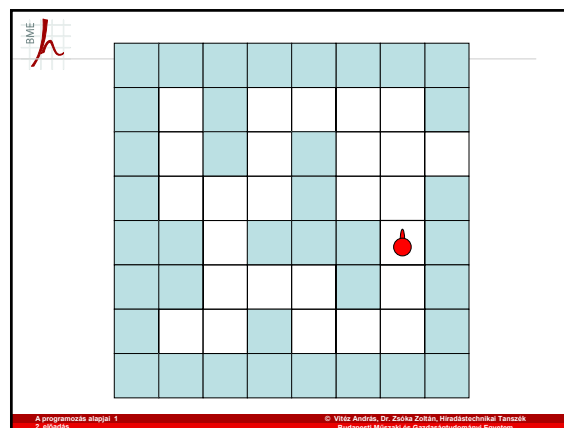
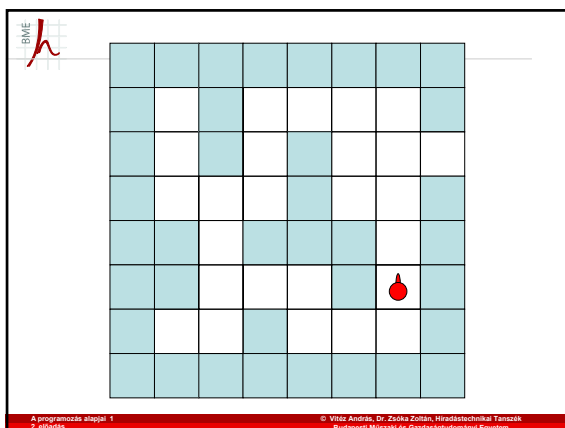
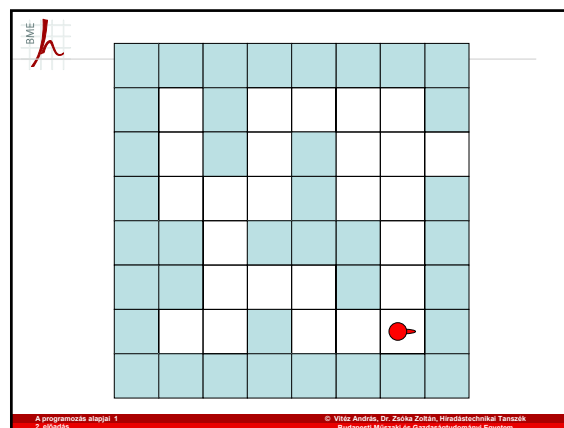
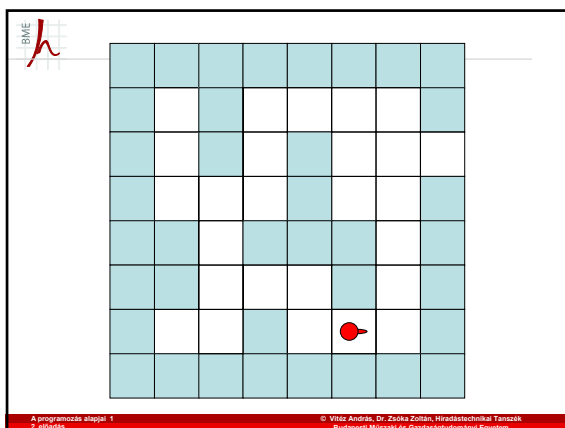
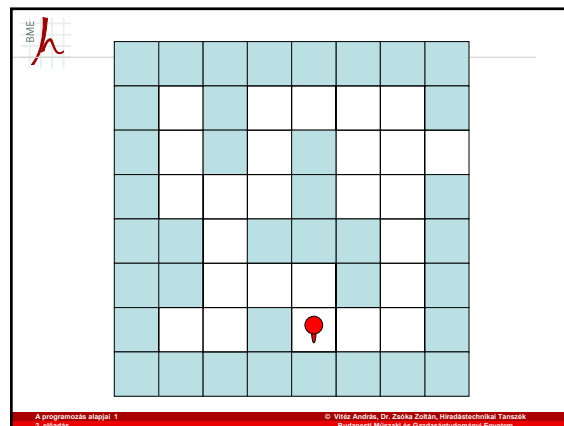
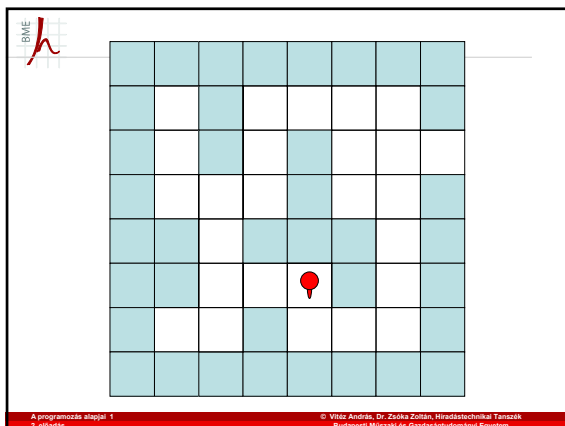


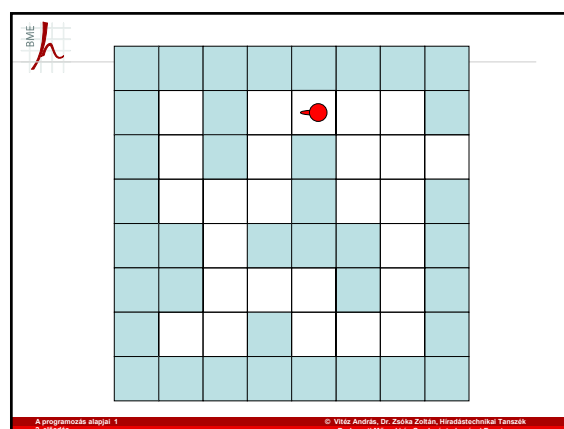
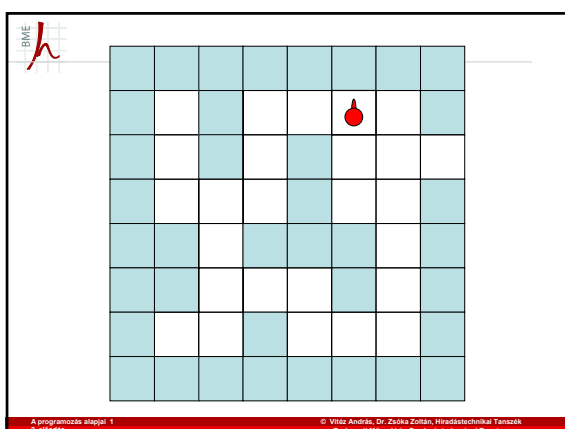
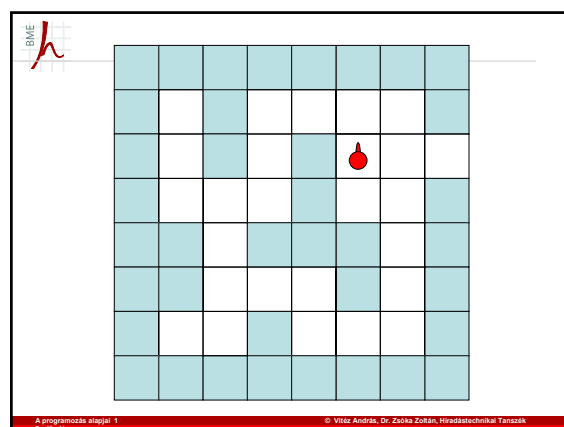
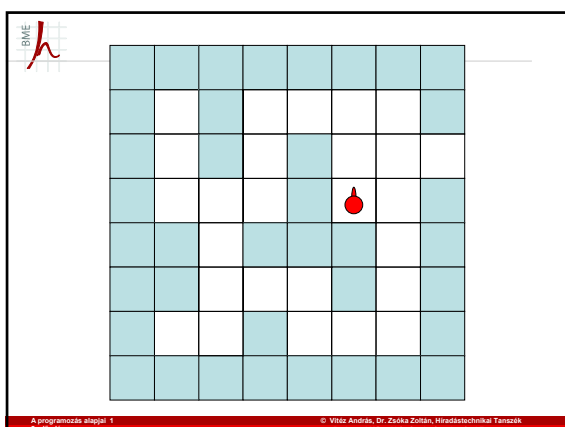
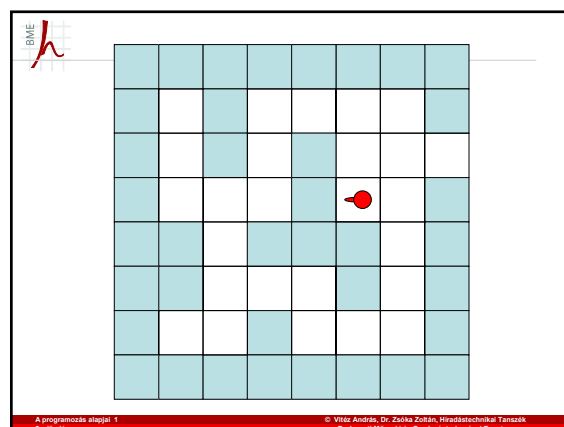
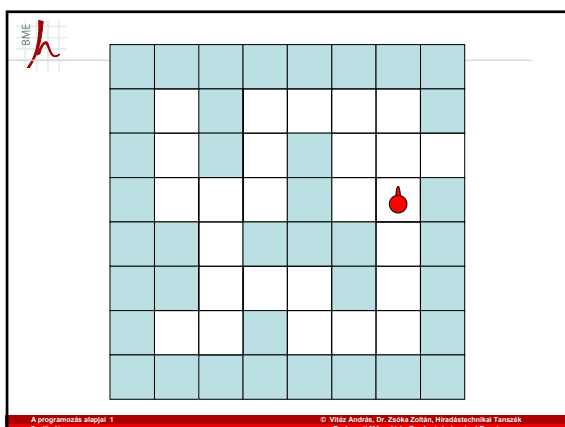


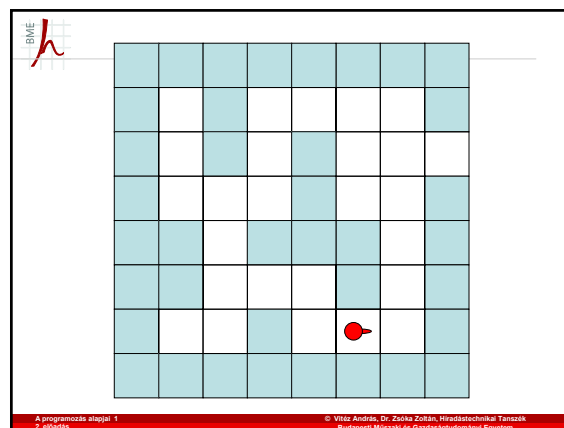
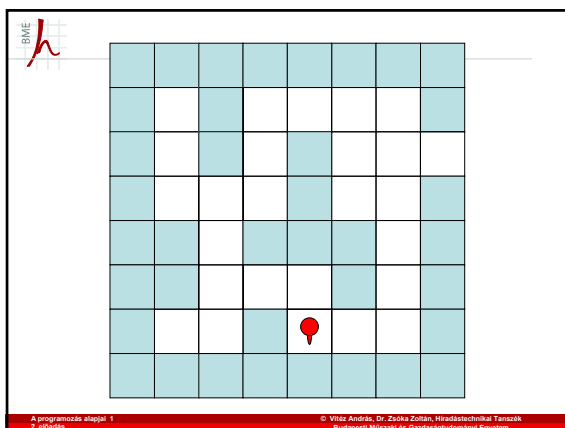
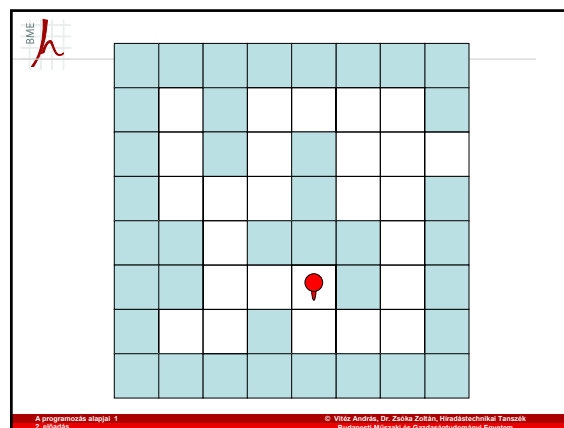
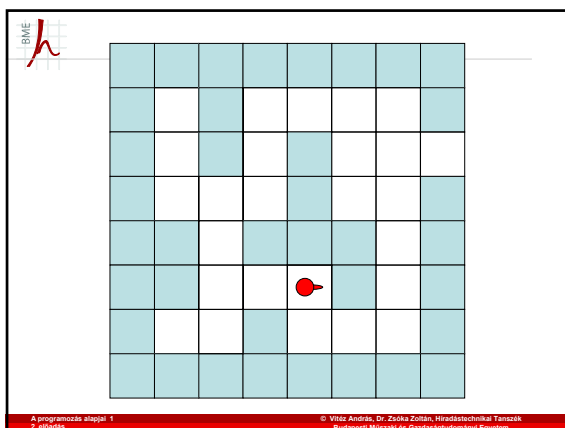
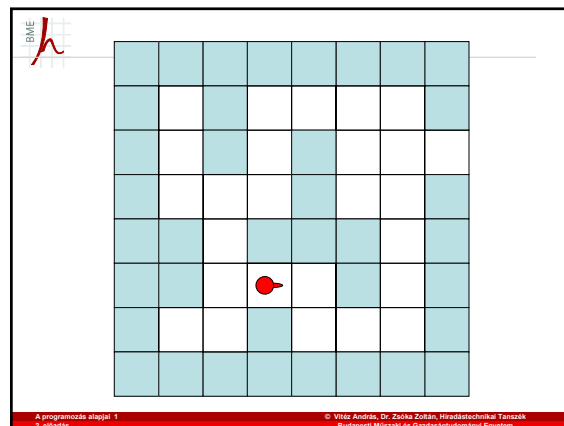
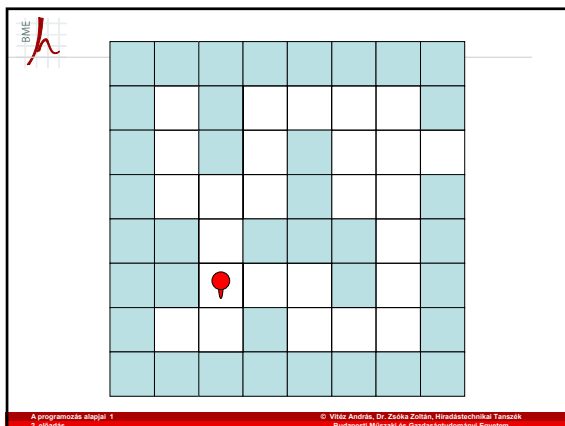


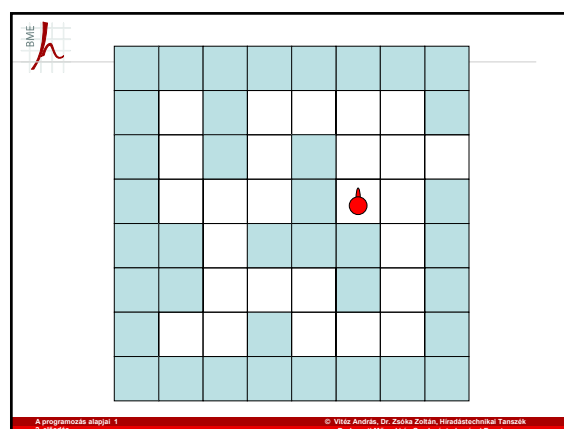
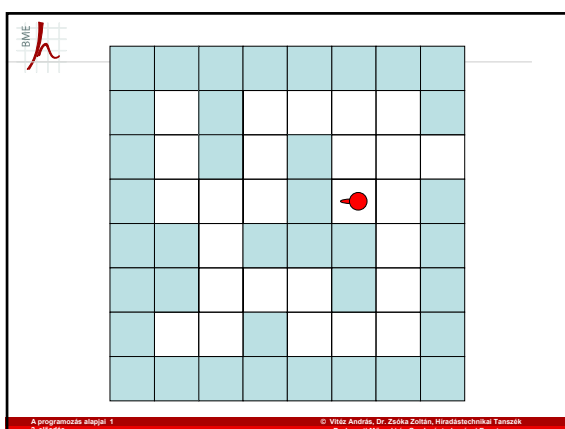
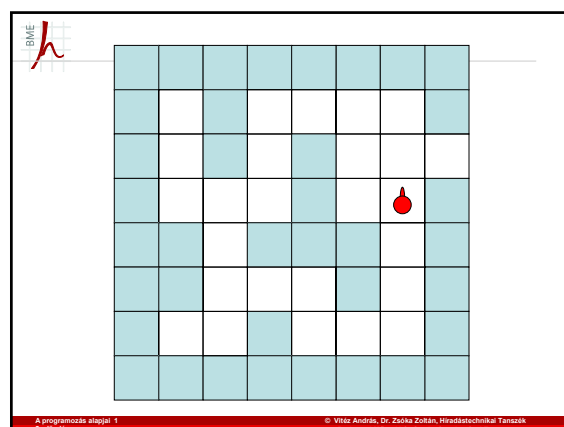
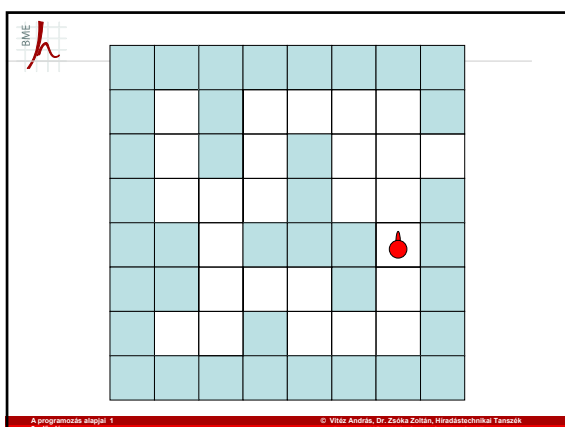
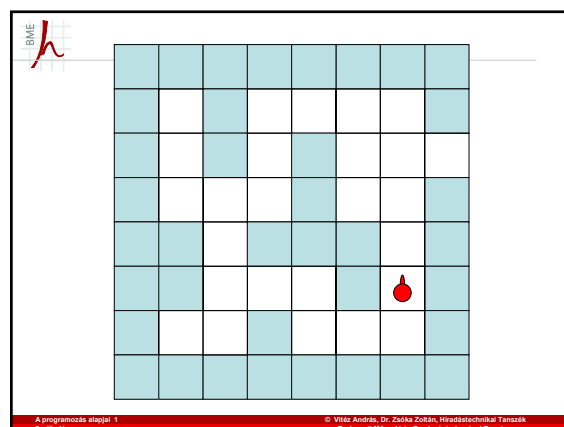
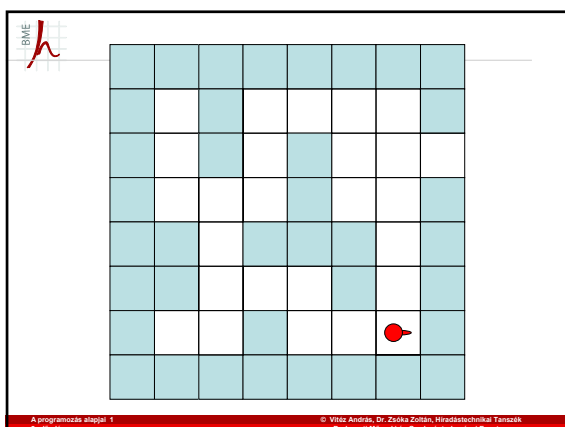


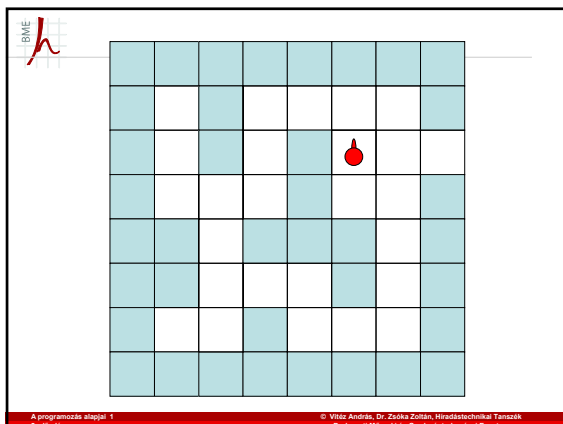












Algoritmus fogalma

A kezelhetetlen probléma definíciója

- Azok a problémák amelyek a futási idő tekintetében a feladat méretének exponenciális függvényével jellemezhetők, vagy
- Memória szükséglet tekintetében polinomiálisak

számítástechnikai szempontból kezelhetetlenek

Algoritmus fogalma

Példa: Isten algoritmusa

21626001637244900000 feldolgozandó állapot
Ha másodpercenként 1 000 000-t dolgozunk fel,
kb. 685 756 évig tart.
Az emberiség eddigi története < 10 000 év

Programozási nyelvek osztályozása

Algoritmusok kódolása:

STRUKTÚRÁTLAN és **STRUKTÚRÁLT** eszközökkel

Programozási nyelvek osztályozása

Gépközei programozás:

Struktúrálatlan és struktúrált eszközök egyaránt megengedettek

Programozási nyelvek osztályozása

Magas szintű nyelvi programozás:

Struktúrálatlan eszközök **kerülendő**

A nyelvek használatukat **lehetővé teszik**
(= tartalmaznak olyan elemeket, amelyek segítségével szükség esetén megvalósíthatók)

Programozási nyelvek osztályozása

Magas szintű nyelvi programozás:

Struktúrált eszközökből építkezünk

A nyelvek használatukat támogatják
(= olyan elemeket tartalmaznak, amelyek a megvalósítást közvetlenül elősegítik)

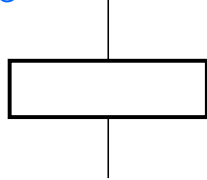
A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Programok építőkövei

A programok építőkövei:

ELEMI TEVÉKENYSÉG



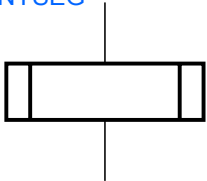
A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Programok építőkövei

A programok építőkövei:

ÖSSZETETT TEVÉKENYSÉG



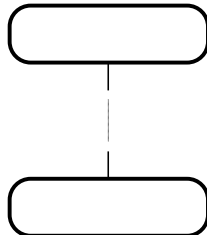
A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Programok építőkövei

A programok építőkövei:

ÖSSZETETT TEVÉKENYSÉG KIFEJTÉSE



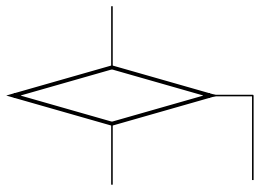
A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Programok építőkövei

A programok építőkövei:

DÖNTÉSI PONT



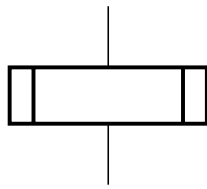
A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Programok építőkövei

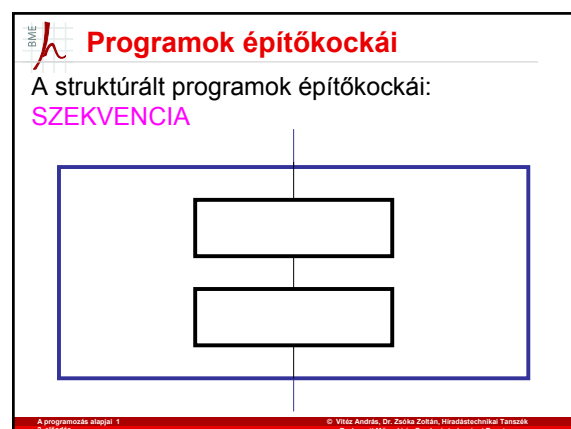
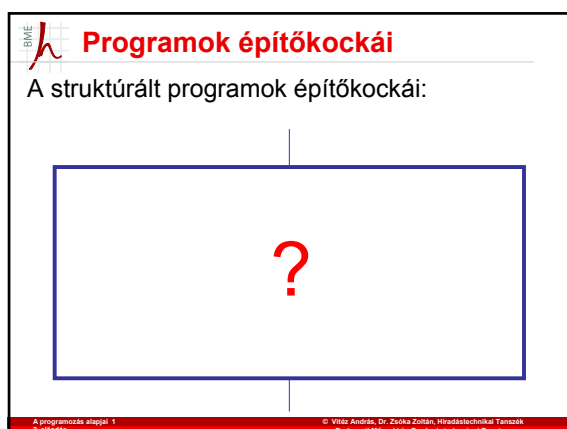
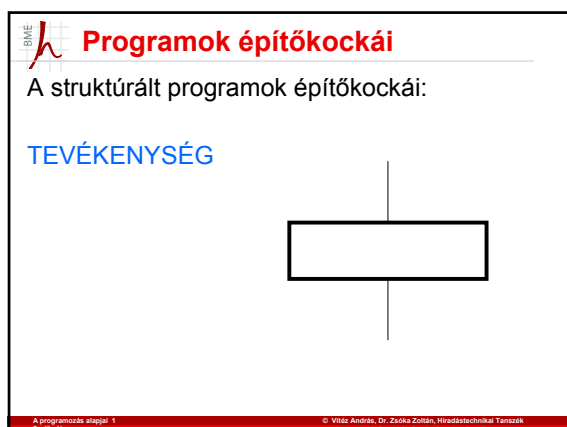
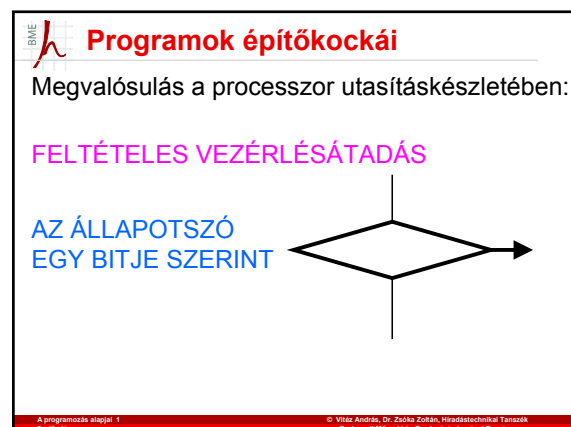
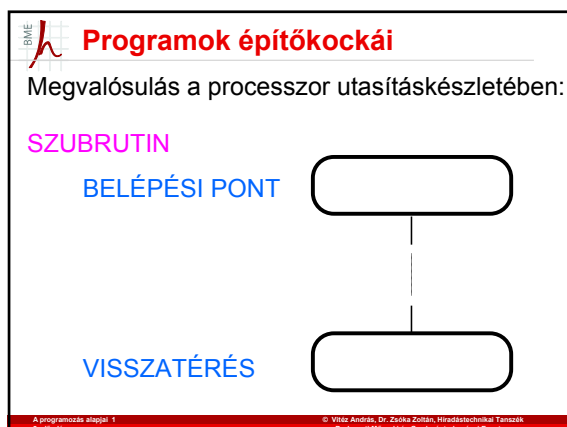
Megvalósulás a processzor utasításkészletében:

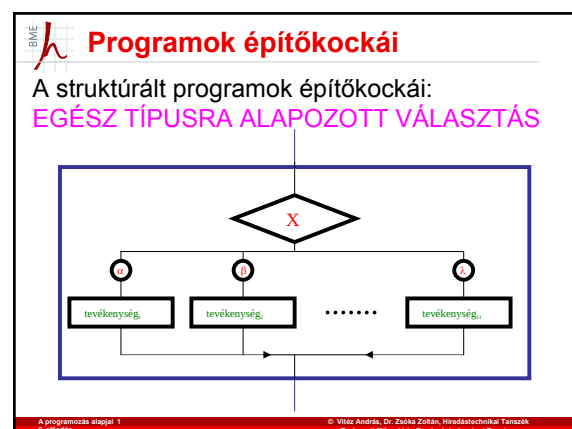
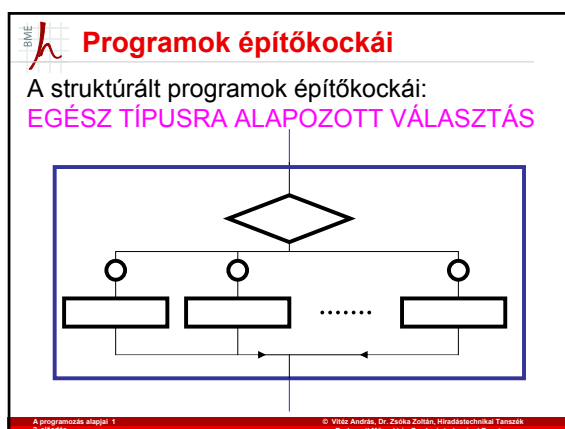
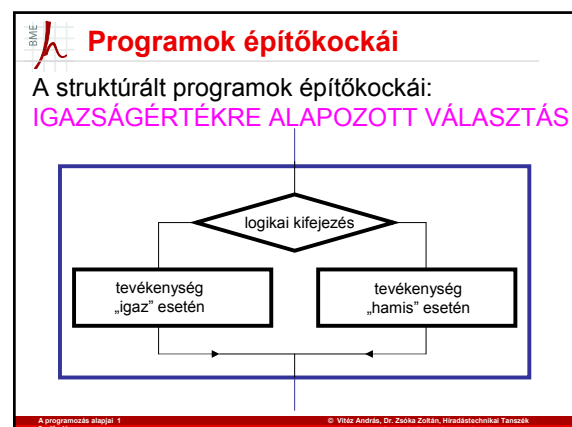
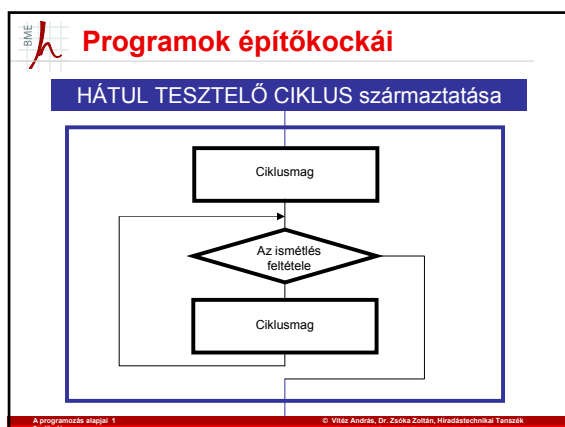
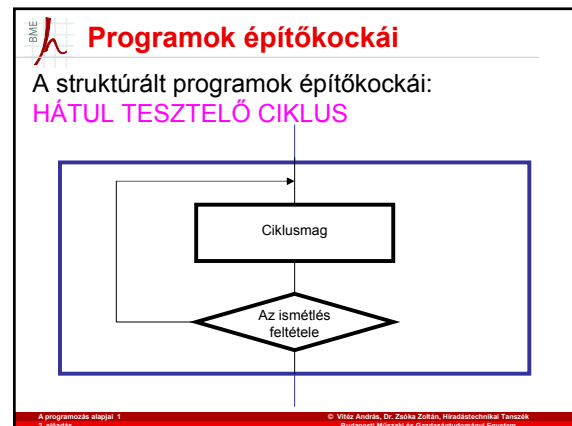
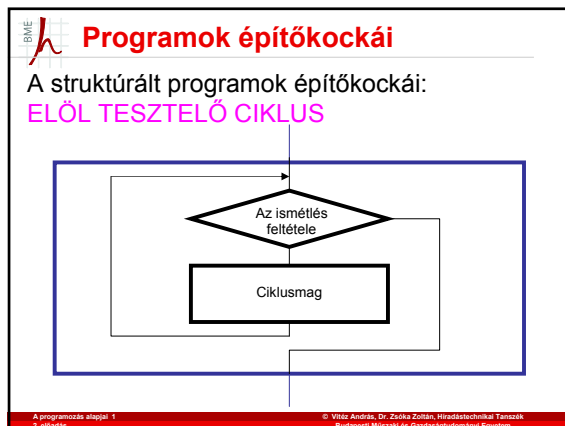
SZUBRUTIN HÍVÁS

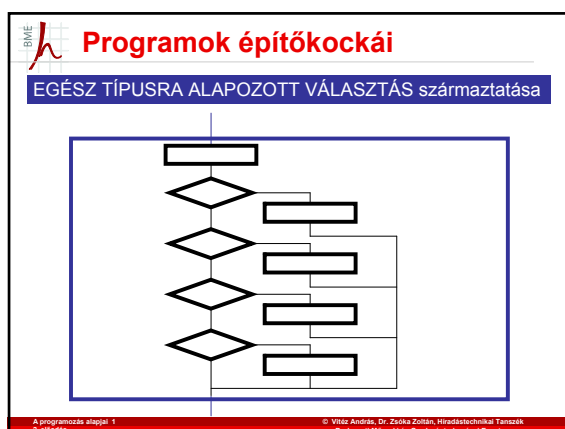
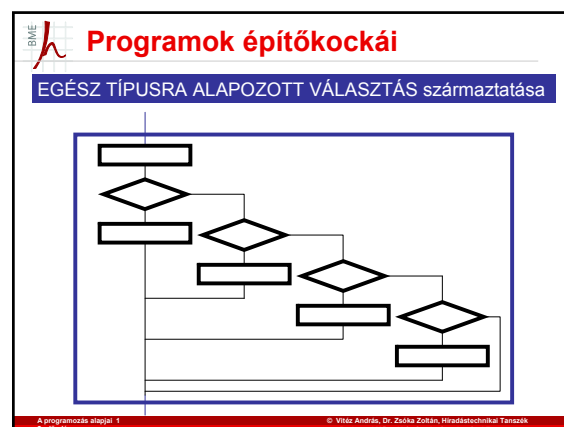
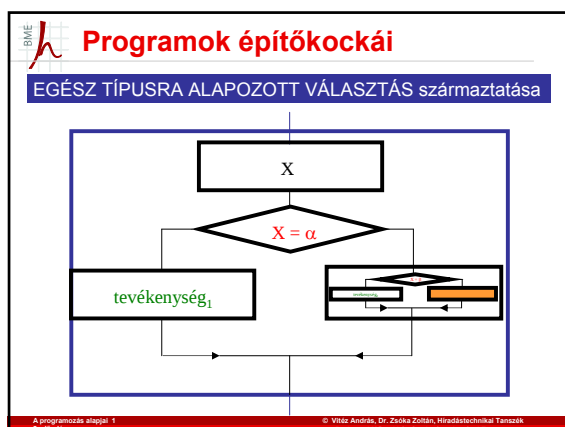
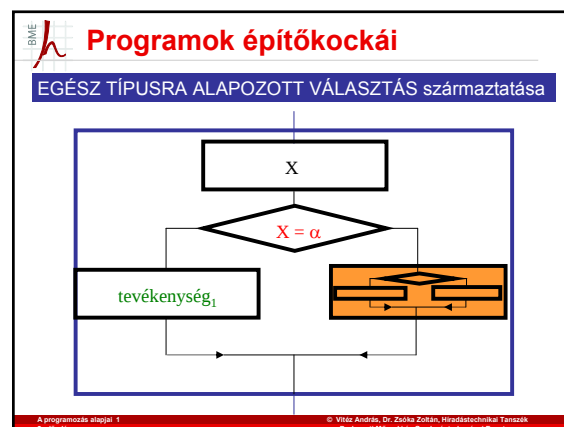
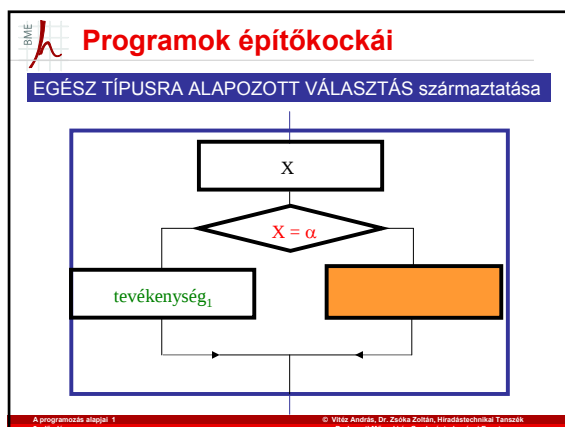


A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem







A strukturált programozás

A strukturált programozás alaptétele:

Minden algoritmusnak létezik strukturált programja

A programozás 1. 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A struktúrált programozás

Az algoritmus leírásában megengedett elemek:

- elemi tevékenység
- szekvencia
- elágazás (döntés függvényében)
- becsatlakozás

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A struktúrált programozás

A struktúrált program leírásában megengedett elemek:

- elemi tevékenység
- szekvencia
- ciklus (elől- és hátultesztelő)
- választás (igazságértékre és egész típusra alapozott)

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A struktúrált programozás

Ekvivalens átalakításokkal minden egy bemenetű és egy kimenetű gráf megengedett alakra hozható.

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A struktúrált programozás

Matematikai értelemben nem teljesen korrekt, de elfogadott a következő szemléletes bizonyítás:

- A problémát a struktúráltan vezérlés-átadások okozzák.
- Vezessünk be egy segédváltozót!
- A program kezdetekor állítsuk nullára!
- Minden becsatlakozást számozzunk meg!
- Amikor számozott ponthoz érkezünk, állítsuk be a változó értékét,
- és a tevékenységet hagyjuk abba!
- Ciklusban keressük a folytatás helyét,
- és mindig a kívánt ágot hajtsuk végre!

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A struktúrált programozás

Példa: $N \leftarrow Y$

A programozás alapjai 1
2. előadás

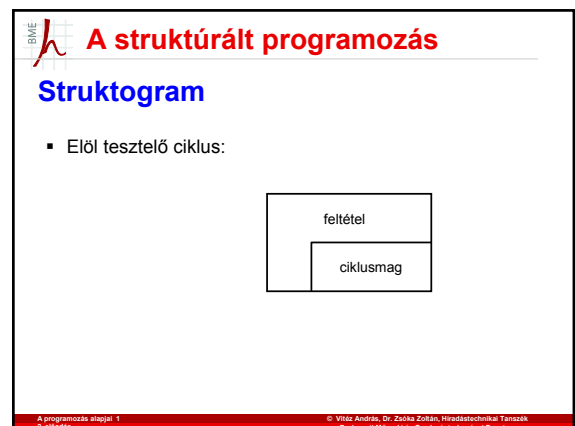
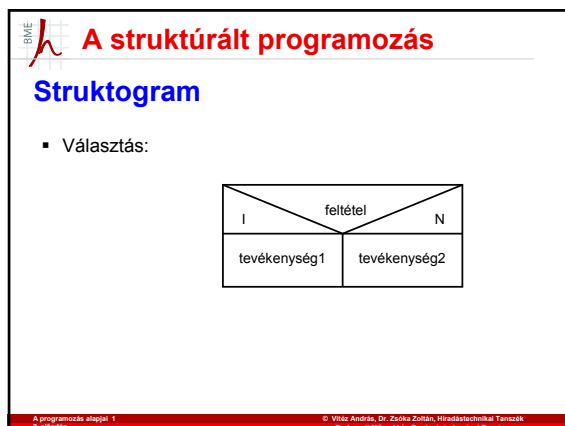
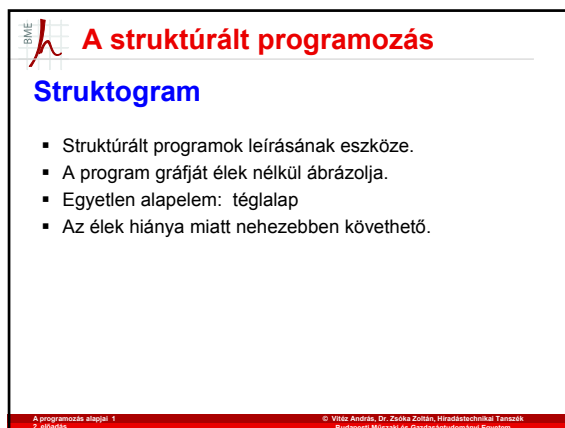
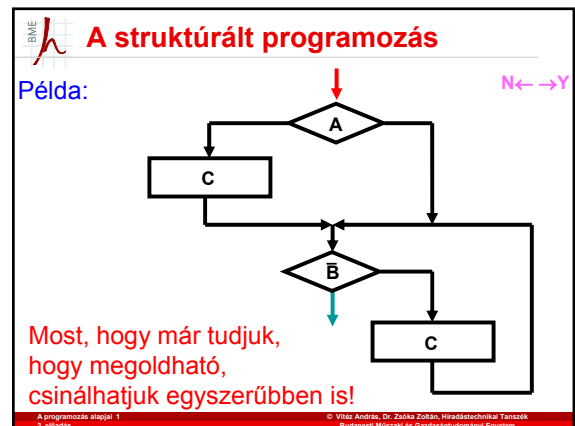
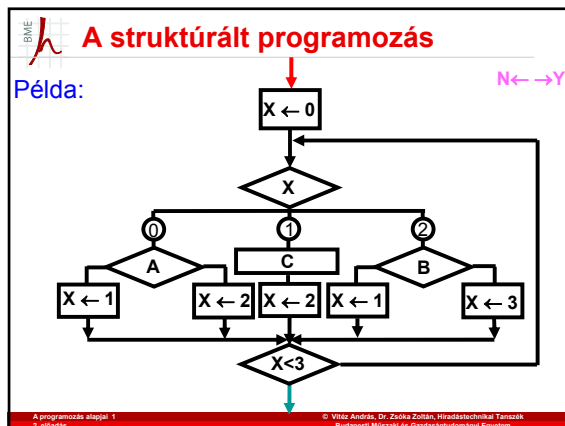
© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A struktúrált programozás

Példa: $N \leftarrow Y$

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem





A struktúrált programozás

Struktogram

- Hátul tesztelő ciklus:



A programozás alapjai 1
2. előadás

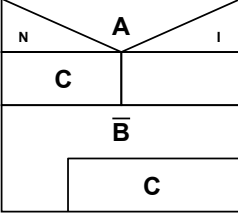
© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A struktúrált programozás

Struktogram

- Az előbbi példánk:



A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



Minimális követelmények

Mit várunk egy magas szintű nyelvtől ?

- Értékkadás
- Szegmens definíció
- Eljáráshívás
- Igazságértékre alapozott választás
- Elöl tesztelő ciklus

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv alaki szabályai

Néhány általános szabály

- Kis- és nagybetű különbözik
- Sorokra tördelhető és tabulálható
- Megjegyzés beszűrhető
- A nyelv utasításai nem használhatók azonosítóként
- A nyelv operátorai azonosítók belsejében sem fordulhatnak elő
- Ahol utasítás állhat, ott blokk is állhat:

```

{
    deklarációk
    utasítások
}

```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C utasításai

Kifejezés igazságértéke

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C utasításai

Kifejezés igazságértéke

HAMIS

ha a kifejezés értékének belső ábrázolása csupa 0 értékű bitből áll, függetlenül az ábrázolás hosszától

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C utasításai

Kifejezés igazságértéke

HAMIS
ha a kifejezés értékének belső ábrázolása csupa 0 értékű bitből áll, függetlenül az ábrázolás hosszától

IGAZ
minden más esetben

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

üres utasítás

- Alakja
;
- Jelentése:
bárhol állhat, ahol utasítás állhat, de semmit sem akarunk tenni

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

Kifejezés utasítás

Kiváltja

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

Kifejezés utasítás

Kiváltja

- az értékadást
- az eljáráshívást (ezért C-ben csak függvény van!)

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

Kifejezés utasítás

Példák:

```
a+5;
b=3*a+1;
printf("Abrakadabra");
s=sin(9*b);
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

IF utasítás

- Alakja:
if (kifejezés) utasítás
vagy
if (kifejezés) utasítás else utasítás
- Jelentése:
igazságértékre alapozott választás

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

IF utasítás

Példák:

```
if(a>3) b=2;
if(a>3) b=2; else c=9;
if (a>3) ; else c=9;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

IF utasítás

Alkalmazási példa:

Vegyük az a változó abszolút értékét!

Ha a negatív,
írjuk át az ellentettjére,
egyébként ne csináljunk semmit!

```
if (a<0) a = -a;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

IF utasítás

Alkalmazási példa:

Vegyük az a változó abszolút értékét!

Ha a negatív,
írjuk át az ellentettjére,
egyébként ne csináljunk semmit!

```
if (a<0) a = -a;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

IF utasítás

Alkalmazási példa:

Vegyük az a változó abszolút értékét!

Ha a negatív,
írjuk át az ellentettjére,
egyébként ne csináljunk semmit!

```
if (a<0) a = -a;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

IF utasítás

Alkalmazási példa:

Vegyük az a változó abszolút értékét!

Ha a negatív,
írjuk át az ellentettjére,
egyébként ne csináljunk semmit!

```
if (a<0) a = -a;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

SWITCH utasítás

- Alakja:


```
switch (egész kifejezés) {
case konstans: utasítások
...
case konstans: utasítások
default: utasítások
}
```
- Jelentése: az egyező címkehez tartozó utasításoknál lép be a vezérlés

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

- A default címke nem kötelező
- A default címke bárhol állhat

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Példa:

```
switch (3*a-2){
    case 4: b=2*b;
           c=b-1;
    default: b=6;
           d=4;

    case 1:
    case 7: c=0;
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!

Abban az esetben, ha a jegy=1, írjuk ki, elégtelen, és ezzel befejeztük!
 Abban az esetben, ha a jegy=2, írjuk ki, elégséges, és ezzel befejeztük!
 Abban az esetben, ha a jegy=3, írjuk ki, közepes, és ezzel befejeztük!
 Abban az esetben, ha a jegy=4, írjuk ki, jó, és ezzel befejeztük!
 Minden más esetben, írjuk ki, jeles!

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!

Abban az esetben, ha a jegy=1, írjuk ki, elégtelen, és ezzel befejeztük!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!

Abban az esetben, ha a jegy=1, írjuk ki, elégtelen, és ezzel befejeztük!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!
Abban az esetben, ha a jegy=2, írjuk ki, elégséges, és ezzel befejeztük!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!
Abban az esetben, ha a jegy=3, írjuk ki, közepes, és ezzel befejeztük!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!
Abban az esetben, ha a jegy=4, írjuk ki, jó, és ezzel befejeztük!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!
Minden más esetben, írjuk ki, jeles!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

SWITCH utasítás

Alkalmazási példa:

Írjunk ki egy osztályzatot szóvegesen!
Minden más esetben, írjuk ki, jeles!

```
switch (jegy){
    case 1: printf("elégtelen"); break;
    case 2: printf("elégséges"); break;
    case 3: printf("közepes"); break;
    case 4: printf("jó"); break;
    default: printf("jeles");
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

WHILE utasítás

- Alakja: **while** (kifejezés) utasítás
- Jelentése: elől tesztelő ciklus

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Informatikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

WHILE utasítás

Példa:

```
while (a>50) a=a-4;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

WHILE utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

Legalább 1 bit kell.
Amíg $a > 1$, felezzük meg, és növeljük a bitek számát!

```
b=1; while (a>1){ a = a/2; b = b+1; }
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

WHILE utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

Legalább 1 bit kell.
Amíg $a > 1$, felezzük meg, és növeljük a bitek számát!

```
b=1; while (a>1){ a = a/2; b = b+1; }
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

WHILE utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

Legalább 1 bit kell.
Amíg $a > 1$, felezzük meg, és növeljük a bitek számát!

```
b=1; while (a>1){ a = a/2; b = b+1; }
```

Első tesztelő ciklus kell, mert lehet, hogy egyszer sem kell feleznünk!

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

DO utasítás

- Alakja: `do utasítás while (kifejezés);`
- Jelentése: hátul tesztelő ciklus

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem



A C nyelv utasításai

DO utasítás

Példa:

```
do a=a+4; while (a>50);
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

DO utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

A bitek számlálását kezdjük 0-tól!
 Növeljük a bitek számát, és felezzük a -t, amíg $a > 1$!

```
b=0; do {b = b+1; a = a/2;} while
```

A programozás alapjai 1
 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
 Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

DO utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

A bitek számlálását kezdjük 0-tól!
 Növeljük a bitek számát, és felezzük a -t, amíg $a > 1$!

```
b=0; do {b = b+1; a = a/2;} while (a>1);
```

A programozás alapjai 1
 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
 Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

DO utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

A bitek számlálását kezdjük 0-tól!
 Növeljük a bitek számát, és felezzük a -t, amíg $a > 1$!

```
b=0; do {b = b+1; a = a/2;} while
```

A programozás alapjai 1
 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
 Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

FOR utasítás

- Alakja: **for** (*kif1*; *kif2*; *kif3*) *utasítás*
- Jelentése:
 - elől tesztelő ciklus
 - kif1*: inicializáló kifejezés
 - kif2*: feltétel kifejezés
 - kif3*: utótevékenység kifejezés

A programozás alapjai 1
 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
 Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

FOR utasítás

- Mindhárom kifejezés hiányozhat
- szándékos végtelen ciklus

```
for (;;) utasítás
```

A programozás alapjai 1
 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
 Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

FOR utasítás

Példák:

```
for(i=0; i<3; i++) printf("%d\n", i);
for(i=2; i<9;) {i=i+3; b=2*i;};
for(;;) jelzolampat_allit();
```

A programozás alapjai 1
 2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
 Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

FOR utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

Legalább 1 bit kell.

Amíg $a > 1$, felezzük meg, és növeljük a bitek számát!

```
for (b=1; a>1; b = b+1) a = a/2;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

FOR utasítás

Alkalmazási példa:

Határozzuk meg, hány biten írható fel az a szám!

Legalább 1 bit kell.

Amíg $a > 1$, felezzük meg, és növeljük a bitek számát!

```
for (b=1; a>1; b = b+1) a = a/2;
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

BREAK utasítás

- Alakja: **break;**
- Jelentése: kilép az öt közvetlenül tartalmazó while, do, for vagy switch utasításból

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

BREAK utasítás

Kilóg a struktúrált szerkezetből, mégis használjuk, mert

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

Példa BREAK használatára

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

CONTINUE utasítás

- Alakja: **continue;**
- Jelentése: átlépi az öt közvetlenül tartalmazó while, do vagy for ciklus magjának még hátralévő utasításait
- Vigyázat! for esetében az utótevékenység következik

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

CONTINUE utasítás

Kilóg a struktúrált szerkezetből,
mégis használjuk, mert

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

Példa CONTINUE használatára

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

RETURN utasítás

- Alakja: **return;**
vagy **return kifejezés;**
- Jelentése: az adott kifejezés értékével visszatér a függvényből

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

RETURN utasítás

- Nem kell, hogy a függvény utolsó utasítása legyen.
- Nem kell, ha nincs visszatérési érték, és egyébként utolsó utasítás volna.

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

RETURN utasítás

Példa:

```
int hexa_value(char c){
    if (c<='9')
        return c-'0';
    if (c<='F')
        return c-'A'+10;
    return c-'a'+10;
}
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

GOTO utasítás

- Alakja: **goto címke;**
- Jelentése: feltétel nélkül átadja a vezérlést a címkével jelzett utasításra

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

A C nyelv utasításai

GOTO utasítás
Kilóg a struktúrált szerkezetből.

NEM használjuk !!!

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét ϵ pontossággal
Ezt láttuk a gyakorlaton:

```

graph TD
    Start([gyökhely]) --> Cond1{p-n > epsilon}
    Cond1 -- N --> End([ ])
    Cond1 -- I --> Calc1[k <- (n+p)/2]
    Calc1 --> Cond2{f(k) < 0}
    Cond2 -- I --> Calc2[n <- k]
    Cond2 -- N --> Calc3[p <- k]
    Calc2 --> Cond1
    Calc3 --> Cond1
  
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét ϵ pontossággal
Ugyanez struktogrammal ábrázolva:

```

graph TD
    Header[p-n > epsilon]
    Body[k <- (n+p)/2]
    Body --> Cond{f(k) < 0}
    Cond -- I --> Calc1[n <- k]
    Cond -- N --> Calc2[p <- k]
    Calc1 --> Header
    Calc2 --> Header
  
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét ϵ pontossággal

```

/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > epsilon)
{
    k = (n+p) / 2;
    if (f(k) < 0)
        n = k;
    else
        p = k;
}
  
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét ϵ pontossággal

```

/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > epsilon)    ← elől tesztelő ciklus
{
    k = (n+p) / 2;
    if (f(k) < 0)
        n = k;
    else
        p = k;
}
  
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét ϵ pontossággal

```

/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > epsilon)    ← elől tesztelő ciklus
                           az ismétlés feltétele
{
    k = (n+p) / 2;
    if (f(k) < 0)
        n = k;
    else
        p = k;
}
  
```

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```
/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}
```

← elől tesztelő ciklus
← az ismétlés feltétele
← a ciklus magja

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```
/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}
```

← elől tesztelő ciklus
← az ismétlés feltétele
← a ciklus magja utasítás helyett blokk

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```
/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}
```

← két utasításból álló szekvencia

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```
/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}
```

← kifejezés utasítás
← két utasításból álló szekvencia

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```
/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}
```

← kifejezés utasítás
← két utasításból álló szekvencia
← igazságértékre alapozott választás

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```
/*
f(x) egy szigorúan monoton növekvő függvény
f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}
```

← feltétel kifejezés

A programozás alapjai 1
2. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```

/*
f(x) egy szigorúan monoton növekvő függvény
    f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}

```

feltétel kifejezés

az „igen” ágon kifejezés utasítás

A programozás alapjai 1
9. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem

Példa utasítások beágyazására

Keressük az $f(x)$ függvény gyökhelyét e pontossággal

```

/*
f(x) egy szigorúan monoton növekvő függvény
    f(n) < 0 és f(p) > 0
*/
while (p-n > e)
{
    k=(n+p)/2;
    if (f(k)<0)
        n=k;
    else
        p=k;
}

```

feltétel kifejezés

az „igen” ágon kifejezés utasítás

a „nem” ágon kifejezés utasítás

A programozás alapjai 1
9. előadás

© Vitéz András, Dr. Zsóka Zoltán, Híradástechnikai Tanszék
Budapesti Műszaki és Gazdaságtudományi Egyetem