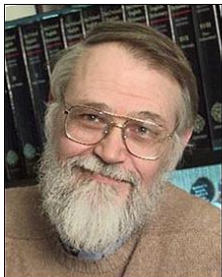


# Programozás C nyelven (1. ELŐADÁS)

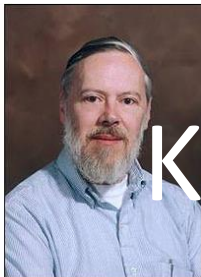
Sapientia EMTE

2020-21



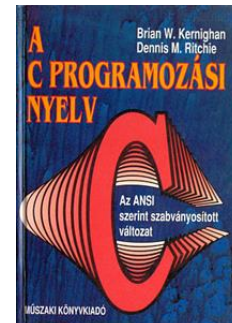


Brian Kernighan



Dennis Ritchie

# Kommunikáció a PC-vel



- Konzol (CON): billentyűzet (`stdin`), monitor (`stdout`)
- NYELV: gépi/emberi/programozási nyelv; Fordító programok
  - C nyelv: ANSI (C89, C99, C11); (tisztá / keverék)
  - Fejlesztői környezet: Visual Studio / **Code::Blocks**
- algoritmus
  - művelet-sor
- program / kód
  - utasítás-sor



# Feladatmegoldás a PC-vel (1)

- [matek] Legyen az  $x^2 + 5x + 6 = 0$  másodfokú egyenlet. Határozzuk meg a gyökeit.
- [programozás] Írj C programot, amely utasítja a számítógépet, hogy
  - **beolvassa** a billentyűzetről egy másodfokú egyenlet együtthatóit (**a, b, c**);
  - meghatározza a gyököket;
  - **kiírja** az eredményeket (**x1, x2**) a képernyőre.

Bemenő  
adatok

Kimenő  
adatok

INPUT: 1 5 6

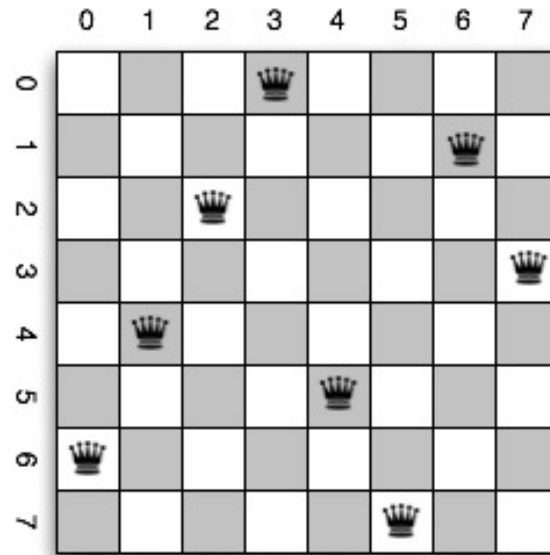
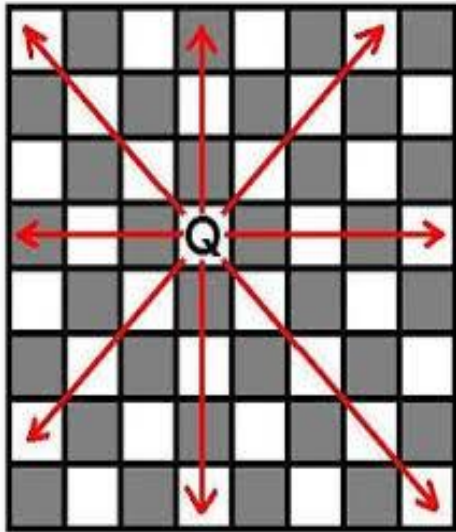
OUTPUT: -2 -3



## Feladatmegoldás a PC-vel (2)

- Helyezzünk el egy sakktáblán 8 királynőt úgy, hogy ne üssék egymást.
- Írj C programot, amely **beolvas** a billentyűzetről egy **n** értéket, generálja az összes lehetőséget, ahogy n királynő elhelyezhető az  $n \times n$  méretű sakktáblán úgy, hogy ne üssék egymást, majd **kiírja** a képernyőre a helyes elhelyezéseknek megfelelő **koordináta-pár-sorokat**.

# Feladatmegoldás a PC-vel (2)



INPUT: 8

OUTPUT:

(0, 3)

(1, 6)

(2, 2)

(3, 7)

(4, 1)

(5, 4)

(6, 0)

(7, 5)

# Memória-változók (tárolók)

- STACK/HEAP/ADATSZEGMENS

- Jellemzők:

- (1) azonosító, (2) típus, (3) cím, (4) érték

- azonosító: **a..z**, **A..Z**, **0..9**, **\_**

- típus:

- egész: **char**, **int**

- valós: **float**, **double**

- **void**

- típus módosítók

- **unsigned**, **signed**, **short**, **long**

- **<típus>** **<azonosító>**,

Memória

|            |          |
|------------|----------|
| 0          | 00000000 |
| 1          | 00000000 |
| 2          | 00000000 |
| .          | 00000000 |
| .          | 00000000 |
| .          | 00000000 |
| 1000       | 00000000 |
| cím → 1001 | 00100100 |
| 1002       | 00011101 |
| 1003       | 00000000 |
| .          | 00000000 |
| .          | 00000000 |
| .          | 00000000 |

```
short int a = 7460;
int x1,x2,x3;
int szam1 = 0, szam2;
double z, pi = 3.14;
short u;
unsigned long v;
```

# Egyszerű típusok

`1 = sizeof(char) ≤ sizeof(short) ≤ sizeof(int) ≤ sizeof(long)`  
`sizeof(float) ≤ sizeof(double)`

| Típus          | Méret | Értéktartomány                                  |
|----------------|-------|---|
| char           | 8     | -128 .. 127                                     |
| unsigned char  | 8     | 0 .. 255  |
| short          | 16    | -32768 .. 32767                                 |
| unsigned short | 16    | 0 .. 65535                                      |
| int            | 16    | -32768 .. 32767                                 |
| int            | 32    | -2147483648 .. 2147483647                       |
| unsigned int   | 16    | 0 .. 65535                                      |
| unsigned int   | 32    | 0 .. 4294967295                                 |
| long           | 32    | -2147483648 .. 2147483647                       |
| unsigned long  | 32    | 0 .. 4294967295                                 |
| float          | 32    | $3.4 \cdot 10^{-38}$ .. $3.4 \cdot 10^{38}$     |
| double         | 64    | $1.7 \cdot 10^{-308}$ .. $1.7 \cdot 10^{308}$   |
| long double    | 80    | $3.4 \cdot 10^{-4932}$ .. $1.1 \cdot 10^{4932}$ |

# Olvasás/írás (ANSI C) billentyűzetről/képernyőre (`scanf`/`printf`)

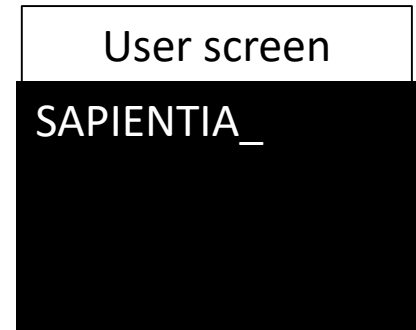
- **`scanf`**: olvasás a standard inputról (`stdin`)
  - A standard input implicite a billentyűzet
- **`printf`**: írás a standard outputra (`stdout`)
  - A standard output implicite a képernyő



# OLVASÁS / ÍRÁS

## billentyűzetről / képernyőre

```
#include <stdio.h>
int main()
{
    printf("SAPIENTIA");
    return 0;
}
```



# OLVASÁS / ÍRÁS

billentyűzetről / képernyőre

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

```
int main()
```

```
{
```

```
int a,b,c;
```

```
scanf("%i", &a);
```

```
scanf("%i", &b);
```

```
c = a+b;
```

```
printf("%i", c);
```

```
return 0;
```

```
}
```

Hivatkozás a változó címére

|   |    |
|---|----|
| — | 13 |
| — | —  |

|    |    |
|----|----|
| 13 | 13 |
| —  | 25 |
| —  | —  |

|    |     |
|----|-----|
| 13 | 13  |
| 25 | 25  |
| —  | 38_ |

É  
R  
T  
É  
K  
A  
D  
Á  
S

# OLVASÁS / ÍRÁS

```
#include <stdio.h>
int main()
{
    int a,b;
    printf("Ird be az 1. szamot: ");
    scanf("%i", &a);
    printf("Ird be a 2. szamot: ");
    scanf("%i", &b);
    printf("Osszeg: %i\n", a+b);
    return 0;
}
```

Ird be az 1. szamot: \_

Ird be az 1. szamot: 13

Ird be az 1. szamot: 13  
Ird be a 2. szamot: \_

Ird be az 1. szamot: 13  
Ird be a 2. szamot: 25

Ird be az 1. szamot: 13  
Ird be a 2. szamot: 25  
Osszeg: 38

# Formázó-karakterek

---

| Típus              | Formázókarakter   |
|--------------------|---|
| char               | c   |
| int                | d vagy i (10-es számrendszerben)<br>o (8-as számrendszerben)<br>x,X (16-os számrendszerben) |
| unsigned int       | u   |
| short int          | hd vagy hi  |
| unsigned short int | hu  |
| long int           | ld vagy li  |
| unsigned long int  | lu  |
| float              | f   |
| double             | lf  |
| long double        | Lf  |
| karakterlánc       | s   |
| pointer            | p   |

---

```
scanf("<formázósor>", <változócímlista);
```

<formázósor>:

%-al bevezetett formázó karakterek

```
int x;  
scanf("%i", &x);
```

```
int x,y;  
scanf("%i%i", &x, &y);
```

```
int x; float y;  
scanf("%i%f", &x, &y);
```

```
char c;  
scanf("%c", &c);
```

```
int x = 7;  
printf("%i", x+1);
```

```
printf("sapi");
```

```
printf("sapi\ninfo\n");
```

```
int x = 3; float y = 3.14;  
printf("x= %i\n", x);  
printf("y= %f\n", y);
```

```
printf("<formázósor>", <kifejezéslista);
```

<formázósor>:

„Normál” karakterek

%-al bevezetett formázó karakterek

\-el bevezetett escape szekvenciák

# PÉLDA (scanf/printf)

```
long a;  
float b;  
double c;  
long double d;  
scanf("%ld%f%lf%Lf", &a, &b, &c, &d);  
  
printf("a=%ld\nb=%f\nc=%lf\nd=%Lf\n", a, b, c, d);
```

A `printf("%d, %o, %x, %X ", 10, 10, 10, 10);` utasítás nyomán a képernyőn a következő jelenik meg: 10, 12, a, A.

?

# PÉLDA (printf)

```
printf("%5.2f", x);
```

?

|          |           |
|----------|-----------|
| <b>x</b> | 13.47     |
|          | 1352.4712 |
|          | 3.4777    |
|          | .5        |
|          | 0.        |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
|   |   | 1 | 3 | . | 4 | 7 |
| 1 | 3 | 5 | 2 | . | 4 | 7 |
|   |   |   | 3 | . | 4 | 8 |
|   |   |   | 0 | . | 5 | 0 |
|   |   |   | 0 | . | 0 | 0 |

13.47

1352.47

3.48

0.50

0.00

```
printf("%5li", y);
```

?

|          |         |
|----------|---------|
| <b>y</b> | 12947   |
|          | 1352947 |
|          | 77      |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
|   |   | 1 | 2 | 9 | 4 | 7 |
| 1 | 3 | 5 | 2 | 9 | 4 | 7 |
|   |   |   |   |   | 7 | 7 |

12947

1352947

77

# OLVASÁS / ÍRÁS állományból/ba

```
#include <stdio.h>
int main()
{
    freopen("be.txt", "r", stdin);
    freopen("ki.txt", "w", stdout);
    double x1,x2,atlag;
    scanf("%lf%lf", &x1, &x2);
    atlag = (x1 + x2) / 2;
    printf("Atlag: %lf\n", atlag);
    freopen("CON", "r", stdin);
    freopen("CON", "w", stdout);
    return 0;
}
```

| be.txt |
|--------|
| 12.5   |
| 24.3   |

Aritmetikai operátorok:  
+, -, \*, /, %

| ki.txt      |
|-------------|
| Atlag: 18.4 |



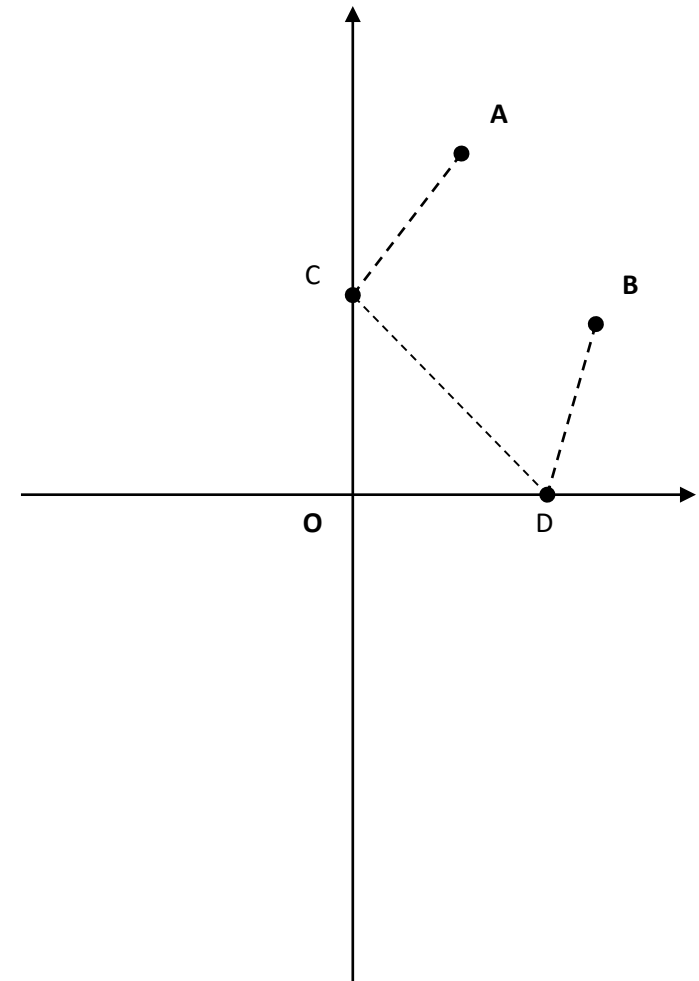
# Olvasás/írás *szöveges* állományból/ba (`freopen`)

- **`freopen`**: lehetővé teszi, hogy standard inputtá/outputtá tegyük a bementi/kimeneti állományunkat
  - `freopen("be.txt", "r", stdin);`
  - `freopen("ki.txt", "w", stdout);`
- A fenti `freopen`-ek nyomán, a `scanf/printf` függvény a `be.txt/ki.txt` állományból/ba olvas/ír
- A billentyűzet/monitor visszaállítása standard inputtá/outputtá
  - `freopen("CON", "r", stdin);`
  - `freopen("CON", "w", stdout);`

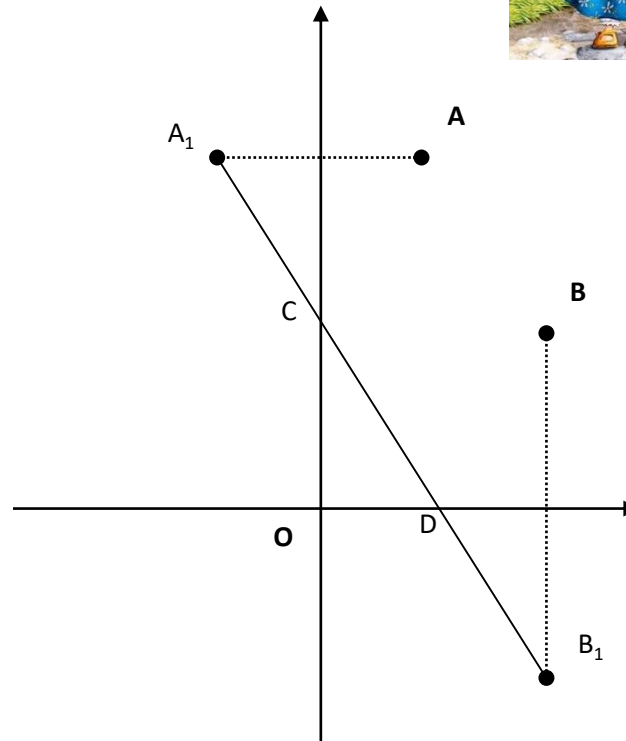
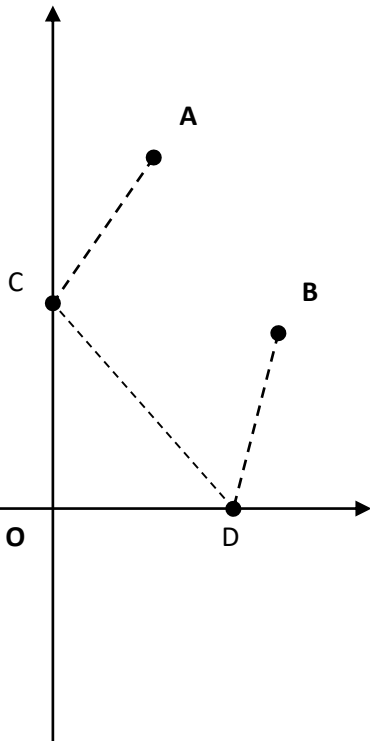
# Feladat (Piroska és a farkas)



- Írjunk programot, amely a billentyűzetről beolvassa két pont (A és B) koordinátáit (az euklideszi sík első negyedébe esnek; A első szögfelező felett, B első szögfelező alatt), és kiírja a képernyőre az A-ból B-be vezető azon legrövidebb út hosszát, amely érinti az  $Ox$  és  $Oy$  egyeneseket.



# Feladat (Piroska és a farkas)



```
double xA, yA, xB, yB, mintav;  
double xA1, yA1, xB1, yB1;  
scanf("%lf%lf%lf%lf", &xA, &yA, &xB, &yB);  
xA1 = -xA; yA1 = yA;  
yB1 = -yB; xB1 = xB;  
mintav = sqrt( (xB1-xA1) * (xB1-xA1) + (yA1-yB1) * (yA1-yB1) );  
printf("%lf", mintav);
```

# ISMÉTLÉS

- **VÁLTOZÓK:** `char, int, float, double`
  - `unsigned, signed, short, long`
  - `<típus> <azonosító>;`



## Billentyűzetről/Képernyőre

- `scanf("<formázó sor>", <változó címlista>);`
- `printf("<formázó sor>", <kifejezés lista>);`
  - Formázó-karakterek:
    - `d/i,o,x,X; u; hd/hi,hu; ld/li,lu; f; lf; Lf; c; s; p`

## Állományból/ba

- `freopen("<állomány név>", "r", stdin);`
  - `scanf`
- `freopen("<állomány név>", "w", stdout);`
  - `printf`