

Programozás C nyelven (13. ELŐADÁS)

Sapientia EMTE

2015-16



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

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

C++

C

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

Olvasás/ Írás scanf/ printf

```
int x;
cin >> x;
```

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

```
int x,y;
cin >> x >> y;
```

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

```
int x; float y;
cin >> x >> y;
```

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

```
char c;
cin >> c;
```

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

```
char szo[10];
cin >> szo;
```

```
char szo[10];
scanf("%s", szo);
```

```
char mondat[100];
cin.getline(mondat, 100);
```

```
char mondat[100];
scanf("%[^\\n]\\n", mondat);
```

```
scanf("<formázó sor>", <változó cím lista>);
<formázó sor>:
%-al bevezetett formázó karakterek
```

Átugorja a '\\n'-t

Mindent olvas, kivéve a '\\n'-t

Formázó-karakterek

Típus	Formázókarakter
char	c
int	d vagy i (10-es számrendszerben) o (8-as számrendszerben) 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

Olvasás/ Írás

scanf/ printf

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

<formázósor>:

„Normál” karakterek

%-al bevezetett formázó karakterek

\-el bevezetett escape szekvenciák

C++

C

```
int x = 7;  
cout << x+1;
```

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

```
cout << "sapi";
```

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

```
cout << "sapi\ninfo\n";
```

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

```
int x = 3; float y = 3.14;  
cout << "x= " << x << '\n';  
cout << "y= " << y << '\n';
```

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

```
char szo[10] = "Neumann";  
cout << szo;
```

```
char szo[10] = "Neumann";  
printf("%s", szo);
```

```
char mondat[100] = "en te";  
cout << mondat;
```

```
char mondat[100] = "en te";  
printf("%s", mondat);
```

PÉLDA (scanf/printf)

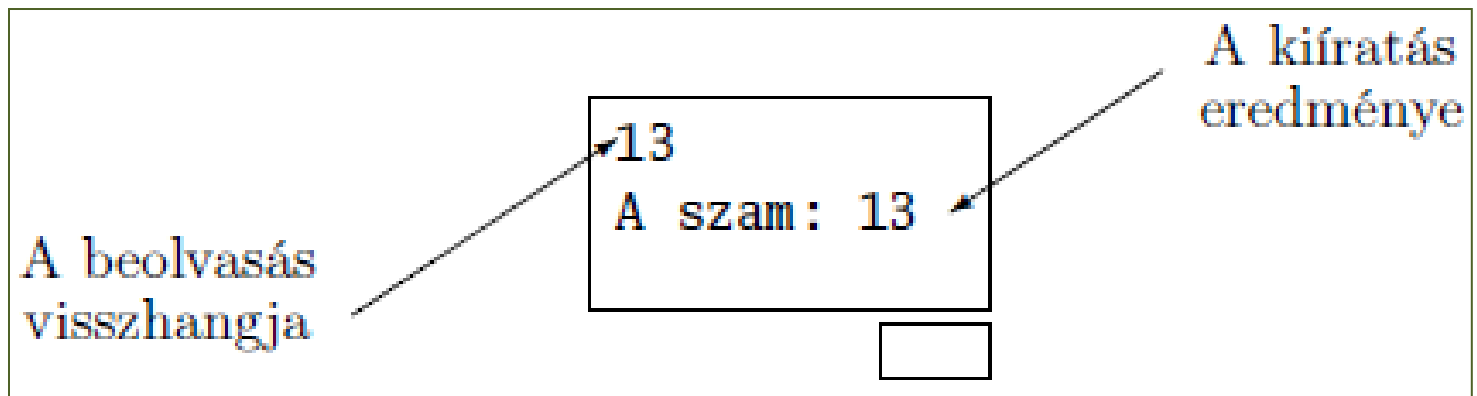
```
long a;  
float b;  
double c;  
long double d;  
scanf("%ld%f%lf%Lf", &a, &b, &c, &d);  
  
printf("a=%ld\nb=%f\nc=%lf\nd=%Lf\n", a, b, c, d);
```

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

?

PÉLDA (scanf/printf)

```
#include <stdio.h>
main()
{
    int a;
    scanf("%d", &a);
    printf("A szám:%d", a);
    return 0;
}
```



Olvassunk be a billentyűzetről egy $n < 100$ elemű számsorozatot, majd írassuk ki a képernyőre fordított sorrendben.

```
int n,a[100],i;
scanf("%i",&n);
for( i = 0 ; i < n ; ++i){
    scanf("%i",&a[i]); // scanf("%i", a+i);
}
printf("A fordított sorozat:\n");
for( i = n-1 ; i >= 0 ; --i){
    printf("%i ", a[i]);
}
```

```
5
13 7 -5 6 11
A fordított sorozat:
11 6 -5 7 13_
```


PÉLDA (printf)

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

?

x	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);
```

?

y	12947
	1352947
	77

		1	2	9	4	7
1	3	5	2	9	4	7
					7	7

12947

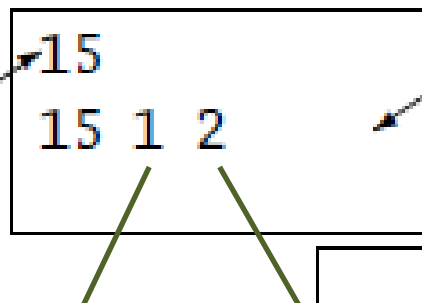
1352947

77

Mit térítenek vissza a scanf/printf függvények?

```
int a, b;  
a = scanf("%d", &n);  
b = printf("%d ", n);  
printf("%d %d", a, b);
```

A scanf
visszhangja



A printf-ek
„nyoma”

Beolvasott
értékek
száma

Kiírt
karakterek
száma

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);`

Olvassunk be állományból egy $n \times m$ méretű legfentebb kétszámjegyű számokat tartalmazó mátrixot, majd írassuk ki állományba, táblázatosan.

```
int n,m,b[100][100],i,j;
freopen("be.txt", "rt", stdin);
freopen("ki.txt", "wt", stdout);

scanf("%i%i", &n, &m);
for( i = 0 ; i < n ; ++i){
    for( j = 0 ; j < m ; ++j){
        scanf("%i", &b[i][j]);
    }
}
for( i = 0 ; i < n ; ++i){
    for( j = 0 ; j < m ; ++j){
        printf("%3i", b[i][j]);
    }
    printf("\n");
}

freopen("CON", "rt", stdin);
freopen("CON", "wt", stdout);
```

be.txt			
3	4		
7	12	5	3
55	22	45	77
9	6	4	7

ki.txt			
7	12	5	3
55	22	45	77
9	6	4	7

Olvasás/írás *szöveges* állományból/ba (**fscanf**/**fprintf**)

```
FILE *fin; //állománypointer (választhatsz más nevet is)
fin = fopen("<állománynév>", "rt");
...
fscanf(fin, "<formázósor>", <változócímlista>);
...
fclose(fin);
```

```
FILE *fout; //állománypointer (választhatsz más nevet
is)
fout = fopen("<állománynév>", "wt");
...
fprintf(fout, "<formázósor>", <kifejezéslista>);
...
fclose(fout);
```

Olvassunk be állományból egy $n \times m$ méretű, legfentebb kétszámjegyű számokat tartalmazó mátrixot, majd írassuk ki állományba, táblázatosan.

```
int n,m,b[100][100],i,j;
FILE *fin, *fout;
fin = fopen("be.txt", "rt"); if(!fin){...}
fout = fopen("ki.txt", "wt"); if(!fout){...}

fscanf(fin, "%i%i", &n, &m);
for( i = 0 ; i < n ; ++i){
    for( j = 0 ; j < m ; ++j){
        fscanf(fin, "%i", &b[i][j]);
    }
}
for( i = 0 ; i < n ; ++i){
    for( j = 0 ; j < m ; ++j){
        fprintf(fout, "%3i", b[i][j]);
    }
    fprintf(fout, "\n");
}

fclose(fin);
fclose(fout);
```

be.txt			
3	4		
7	12	5	3
55	22	45	77
9	6	4	7

ki.txt			
7	12	5	3
55	22	45	77
9	6	4	7

Szöveget tartalmazó állomány feldolgozása karakterenként

Hány karaktert tartalmaz a „szoveg.txt” állomány?

```
#include <stdio.h>
int main(){
    int i; char c;
    FILE *fin;
    fin = fopen("szoveg.txt", "rt");
    if(!fin){
        printf("Sikertelen allomanymegnyitas");
        return 0;
    }
    i = 0;
    while( fscanf(fin, "%c", &c) != EOF ){
        ++i;
    }
    printf("Karakterek szama: %i", i);
    fclose(fin);
    return 0;
}
```

szoveg.txt

Kiss Elemer
Szakkollegium

—

Karakterek szama: 26

Sikertelen olvasás
esetén az fscanf az EOF
konstant téríti vissza

Szöveget tartalmazó állomány feldolgozása szavanként

Hány szót tartalmaz a „szoveg.txt” állomány?

```
#include <stdio.h>
int main(){
    int i; char szo[30];
    FILE *fin;
    fin = fopen("szoveg.txt", "rt");
    if(!fin){
        printf("Sikertelen allomanymegnyitas");
        return 0;
    }
    for( i = 0 ; fscanf(fin, "%s", szo) != EOF ; ++i ){
        ;
    }
    printf("Szavak szama: %i", i);
    fclose(fin);
    return 0;
}
```

szoveg.txt

Kiss Elemer
Szakkollegium
—

Szavak szama: 3

Szöveget tartalmazó állomány feldolgozása soronként

Hány sort tartalmaz a „szoveg.txt” állomány?

```
#include <stdio.h>
int main(){
    int i; char sor[100];
    FILE *fin;
    fin = fopen("szoveg.txt", "rt");
    if(!fin){
        printf("Sikertelen allomanymegnyitas");
        return 0;
    }
    for( i = 0 ; fscanf(fin, "%[^\n]\n", sor) != EOF ; ++i ){
        ;
    }
    printf("Sorok szama: %i", i);
    fclose(fin);
    return 0;
}
```

szoveg.txt

Kiss Elemer
Szakkollegium

—

Sorok szama: 2



Összefoglalás

További részletek
végett lásd a jegyzet
11.2 alfejezetét

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("<állománynév>", "r", stdin);`
 - `scanf`
- `freopen("<állománynév>", "w", stdout);`
 - `printf`
- `FILE *fin, *fout;`
 - `fin = fopen("<állománynév>", "rt");`
 - `fscanf(fin, ...); ... fclose(fin);`
 - `fout = fopen("<állománynév>", "wt");`
 - `fprintf(fout, ...); ... fclose(fout);`