

C programozás

2 óra

A C nyelv típusai

1. Előző óra, rövid összefoglalás

- **A C nyelvű programok feldolgozása**

forráskód (forrásprogram) C file (code.c)



C előfeldolgozó (preprocesszor)

+

C fordító (compiler)



tárgymodul (object modul/file) obj file (code.o)



összefűző-szerkesztő (linkage editor; linker)



végrehajtható program; futtatható állomány (code)

2. A C nyelv alapelemei

- **Alapelemek:** azonosítók, kulcsszavak, konstansok, sztringliterálok, operátorok, írásjelek.

Alapelemek=tokenek (a fordítás során a nyelv “tokenjeit” dolgozza fel)

A C nyelv jelkészlete:

A	B	C	D	E	F	G	H	I	J
K	L	M	N	O	P	Q	R	S	T
U	V	W	X	Y	Z				
a	b	c	d	e	f	g	h	i	j
k	l	m	n	o	p	q	r	s	t
u	v	w	x	y	z				
!	“	#	%	&	'	()	*	+
,	-	/	:	;	<	=	>	?	[
\]	^	_	{	 	}	~		

1. Azonosítók

A program bizonyos összetevőire **névvel** (**azonosító**) hivatkozunk

Első karakter: **_** vagy **betű** (csak ez lehet, kis és nagy betű számít !!!)

pl.: `szam`, `_alap`

2. Kulcsszavak

A kulcsszavak fixek, nem lehet átdefiniálni őket!!!

A C nyelv kulcsszavai:

<code>auto</code>	<code>double</code>	<code>int</code>	<code>struct</code>
<code>break</code>	<code>else</code>	<code>long</code>	<code>switch</code>
<code>case</code>	<code>enum</code>	<code>register</code>	<code>typedef</code>
<code>char</code>	<code>extern</code>	<code>return</code>	<code>union</code>
<code>const</code>	<code>float</code>	<code>short</code>	<code>unsigned</code>
<code>continue</code>	<code>for</code>	<code>signed</code>	<code>void</code>
<code>default</code>	<code>goto</code>	<code>sizeof</code>	<code>volatile</code>
<code>do</code>	<code>if</code>	<code>static</code>	<code>while</code>

3. Konstansok

Konstans alatt mindig számot értünk.

karakter, egész, felsorolt (enum) és lebegőpontos konstansok

egész konstansok: számjegyek sorozatából állnak (csak egész számok)

Decimális (10-es alapú) számrendszer:

pl.: 1787, -23, 0

Oktális (8-as alapú) számrendszer: első jegy 0

pl.: 0342, -030 0

Hexadecimális (16-os alapú) számrendszer: előtag 0x vagy 0X

pl.: 0x7ca, 0x21, 0

karakter konstansok:

egyszeres idézőjelek (') közé zárt egy vagy több karaktert tartalmazó
karaktersorozat

pl.: 'a', 'sghd', '%'

NAGYON FONTOS!!! egyetlen karaktert tartalmazó karakter konstansok által
képviselet számérték a karakter **kódja**

lebegőpontos konstansok: olyan decimális szám , amely (előjeles) valós számot
reprezentál.

a.) Egész-, törtrész közöttük **tizedespon**t (.)

b.) a 10 hatványkitevője az **e** vagy az **E** betűt követi

pl.: 10^{-23} -----> C nyelvben **1e-23** vagy **1E-23**

4. Sztringkonstansok (sztringliterálok):

kettős idézőjelek közé zárt karaktersorozat

pl.: "Ez egy sztringkonstans!" , "Ez egy sztringkonstans!\n"

Tárolás a memóriában: a sztringkonstans végén '\0' (záró karakter)

5. Operátorok

Olyan (egy vagy több karakterből álló) szimbólumok, amelyek előírják, hogyan kell feldolgozni az operandusokat.

!	!=	%	%=	&	&&	&=	()	*	*=
+	++	+=	,	-	--	-=	->	.	/
/=	<	<=	<<	<<=	=	==	>	>=	>>
>>=	?:	[]	^	^=	sizeof		=		~

6. Írásjelek

Az írásjelek a C nyelvben olyan szimbólumokat jelölnek amelyeknek csak szintaktikai szerepe van (azonosítók elkülönítése; kód egyes részeinek kijelölése), DE semmilyen műveletet sem definiálnak. Viszont néhány írásjel egyben operátor is!

- []** tömb kijelölése, méretének megadása
- ()** paraméter- és az argumentum lista kijelölése
- { }** blokkok vagy függvény behatárolása
- *** mutató típus jelölése a deklarációban
- ,** függvény argumentumok elválasztása
- :** címke elválasztása
- ;** utasítás végének jelölése
- #** előfordító direktíva jelölése

3. A C nyelv típusai

1. Deklaráció és definíció

- a.) Minden névről meg kell mondani mire szeretnénk használni, mert így a fordító nem tud mit kezdeni az adott névvel.  **DEKLARÁCIÓ**

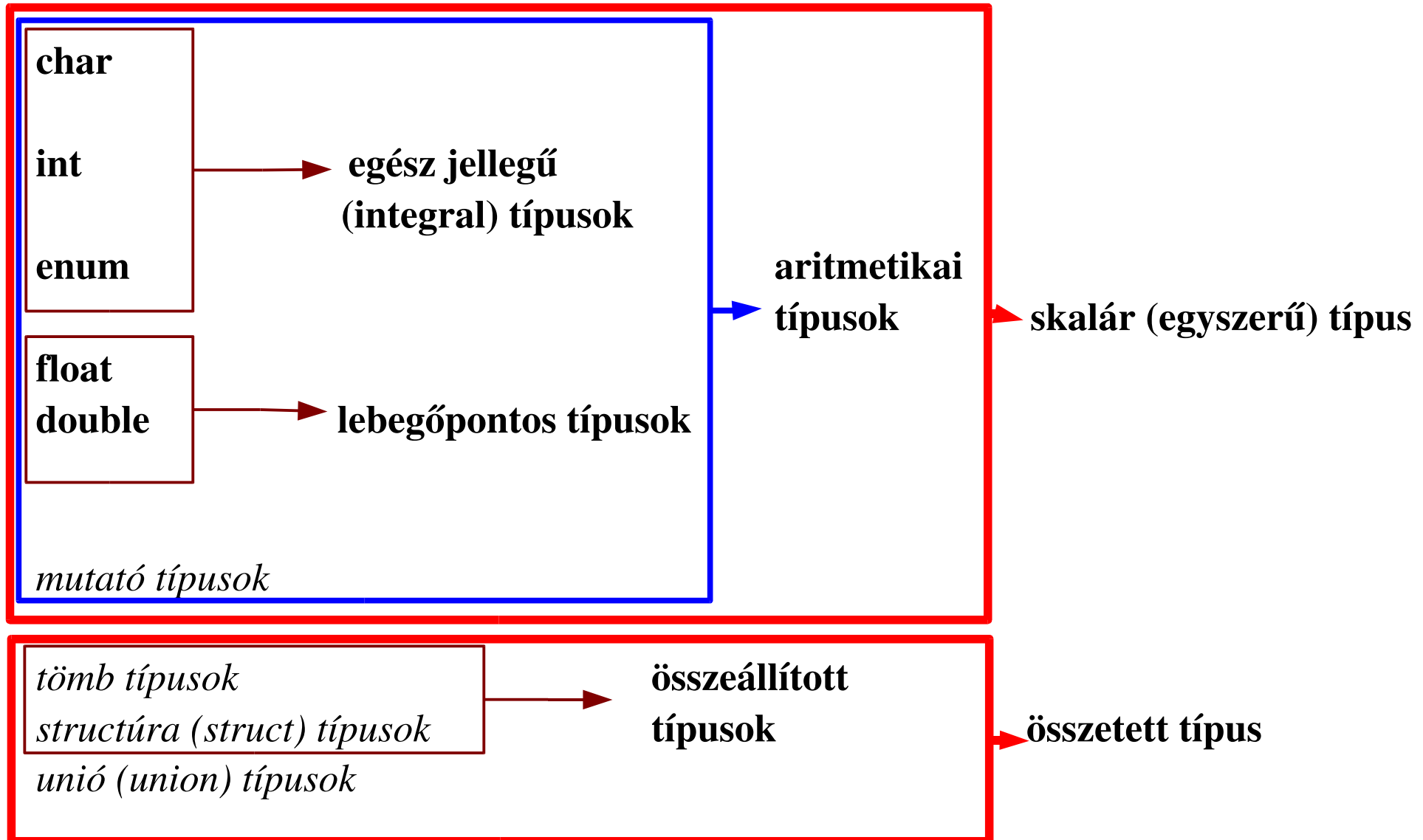
A név tulajdonságait (**típus, tárolási osztály, láthatóság**) közöljük a fordítóval.

- b.) Ha az a cél, hogy a deklarációnak megfelelő objektum is létrejöjjön a memóriában akkor **DEFINÍCIÓT** kell alkalmazni.

Tehát a **DEFINÍCIÓ** olyan **DEKLARÁCIÓ**, amely helyfoglalással jár.

Objektum: olyan memóriaterület jelöl, amely egy vagy több értéket tartalmaz.

2. A C nyelv típusai



3. Típuselőírások, típusmódosítók, típusminősítők

Típusnevek felépítése: típuselőírás

alaptípusok: **char**
int
float
double
enum
struct
union
void (típusnév hiánya; üres típus)

típusmódosítók: **short, long** *a módosítók tárolási hosszát szabályozza*
signed, unsigned *az előjel értelmezését szabályozza*

típusminősítők: **const** *olyan objektum definiálható, amely nem változatható meg (olvasható objektum)*

típuselőírás: (alaptípus, típusmódosítók, típusminősítők)

pl.: signed long int

FONTOS!!! Önmagában a típusmódosító is szerepelhet, mint típuselőírás

pl.: short, short int *azonos típust definiál*

4.Típusok

a.) Egész típusok (int)

short <= int <= long

Az értékészleteket a <limits.h> file adja meg az adott implementációhoz.

Előjeles egész

Típus nevek, minimális és maximális értékek:

	minimális hossz [bit]	típus-megadás alakja	minimális érték <limits.h> -ban	maximális érték <limits.h> -ban
rövid	16	short short int signed short signed short int	SHRT_MIN	SHRT_MAX
normál	16	int signed int	INT_MIN	INT_MAX
hosszú	32	long long int signed long signed long int	LONG_MIN	LONG_MAX

Szám-konstans alakja:

Típus	Számrendszer	Példák
int	decimális	0 123 -33
	oktális	012 0177777
	hexadeimális	0x1a 0x7fff 0xAa1bB
long		15344398l 17558765L

Előjel nélküli egész

Típus nevek, minimális és maximális értékek:

	minimális hossz [bit]	típus-megadás alakja	minimális érték	maximális érték <limits.h> -ban
rövid	16	unsigned short unsigned short int	0	USHRT_MAX
normál	16	unsigned unsigned int	0	UINT_MAX
hosszú	32	unsigned long unsigned long int	0	ULONG_MAX

Szám-konstans alakja:

unsigned	<mint int > u <mint int > U
unsigned long	<mint int > ul <mint int > UL <mint int > lu <mint int > IU ...

b.) Lebegőpontos típusok

float <= double <= long double

Típus	Konstans alakja	Min. abs. érték	Max. abs. érték	pontoság [dec. jegy]
float	12.3f 0.12F 12.F .5f 1E-3f 1.8e5f	FLT_MIN 3.4e-38	FLT_MAX 3.8e38	FLT_DIG >= 6
double	12.3 0.12 12. .5 1E-3 1.8e5	DBL_MIN <= 1e-308	DBL_MAX >= 1e+308	DBL_DIG >= 15
long double	12.3L 0.12l 12.l .5L 1E-3l 1.8e5L	3.4e-4932	3.4e+4932	19

az alapvető “valós” típus a double

c.) Karakter típusok (szinte mint egy egész típus)

Ábrázolás: tipikusan 1 darab 8 bites byte -on.

Hogy egy egyszerű char előjeles-e, az nem definiált, a megvalósítás lehet előjeles és előjel nélküli is, sőt keverheti is a kettőt :egyszer így, máskor úgy!

Alak	Min. értékkészlet	Értékhatárok <limits.h> -ban
signed char	-128 ... 127	SCHAR_MIN ... SCHAR_MAX
unsigned char	0 ... 255	0 ... UCHAR_MAX
char	akár -128 ... 127 akár 0 ... 255	CHAR_MIN ... CHAR_MAX

Karakterkonstansok alakja, speciális értékek:

Karakter konstans (literál)	Alakja	Jelentése
egyszerű karakterek:	'a'	kis a betű
	':'	kettőspont
spec. karakter escape kdja	'\"'	apoztróf : ‘
	'\a'	alarm hangjelzés
	'\b'	backspace visszatörlés
	'\f'	form feed lapdobás
	'\n'	new line újsor jel
	'\r'	carriage return kocsi vissza
	'\t'	horizontális tabulátor
	'\v'	vertikális tabulátor
	'\"'	backslash \
	'\''	idézőjel
	'\?'	kérdőjel
numerikus escape karakterek	'\0'	nulla értékű karakter
	'\10'	oktális 10 8 decimális ért. kar.
	'\x10'	hexadecimális kar. 16 dec. ért.

Számításnál a ... char típusú értéket előbb automatikusan mindig ... int típusúvá alakítja, és azzal számol.

d.) Felsorolt típusok

enum *azonosító {felsorolás}*

pl.: **enum** evszak {tavasz, nyar, osz, tel}

automatikusan 0 1 2 3

pl.: **enum** evszak {tavasz, nyar=4, osz, tel}

 0 4 5 6

vagy akar

pl.: **enum** {tavasz, nyar=4, osz, tel}

5. Egyszerű változók definiálása

egyszerű változó: egy egész vagy lebegőpontos érték tárolására alkalmas

<tárolási osztály> típus <típus...> változónév <kezdőérték> <, ...>;

pl.:

```
int alpha;
```

```
int beta=45;
```

```
int delta;
```

```
int alpha, beta=45, delta;
```

A deklarációban a típust megelőzheti néhány alapszó: **auto**, **register**, **static**, **extern** objektum tárolásával kapcsolatban tartalmaz előírásokat. -----> **tárolási osztály**

objektum **elhelyezkedése, láthatósága, élettartalma;**



5. Saját típus létrehozása

unsigned long double

típusnév, ami meghatároz egy típust

unsigned long double gamma;

definíció, itt a gamma kezdőérték nélküli változó

Speciális tárolási osztály

typedef

Érvényes típusokhoz szinonim nevet rendel hozzá!!!

typedef unsigned long double gorog;
gorog gamma;

6. Konstansok definíciója

a.) **const**

pl.: **const int b;**
 b=3;

const int b=3;

b.) **#define** makrók

pl.: **#define EGY 1**

c.) **enum** típus

6. Értékek

értékadás:

objektum = érték; =értékadás operátora

(kifejezés, változó)

a bal oldal azt az objektumot jelöli ki (címzi meg) a memóriában, ahova a jobb oldalon megadott kifejezés értékét be kell tölteni

balérték (lvalue), illetve jobbérték (rvalue)