

# Jazyk C (ANSI)

## Štruktúra programu/Funkcií

```
#include <filename>    vloží knižničné súbory
#include "filename"    vloží užívateľské súbory
#define meno value     definícia makra/konštanty
typ fnc(typ1,...)     deklarácia funkcie
typ meno              deklarácia externej premennej
main() {
    declarations      deklarácie lokálnych premenných
    prikazy
}
typ fnc(arg1,...) {   definícia funkcie
    declarations      deklarácie lokálnych premenných
    prikazy
    return value;
}
/* */                  komentár
main(int argc, char *argv[]) main s argumentmi
```

## Dátové Typy/Deklarácie

znak (1 byte)	char
celé číslo	int
pohyblivá rádová čiarka (malá presnosť)	float
pohyblivá rádová čiarka (dvojitá presnosť)	double
krátke (16 bit) číslo	short
dlhé (32 bit) číslo	long
kladné aj záporné	signed
iba kladné	unsigned
smerník na int, float,...	*int, *float,...
vymenovanie konštant	enum
konštantná (nemenná) hodnota	const
deklarácia externej premennej	extern
registrová premenná	register
lokálna v jednom súbore	static
bez hodnoty	void
štruktúra	struct
pomenovaný dátový typ	typedef typ meno
veľkosť objektu (typ je size_t)	sizeof object
veľkosť typu (typ je size_t)	sizeof(typmeno)

## Konštanty

long (prípona)	L or l
float (prípona)	F or f
vedecký tvar	e
oktálne (prefix nula)	0
hexadecimálne (prefix nula-x)	0x or 0X
znaková konštantá	'a'
znaková konštantá (oktálne)	'\ooo'
znaková konštantá (hexadecimálne)	'\xhh'
newline, cr, tab, backspace	\n, \r, \t, \b
špeciálne znaky	\\, \?, \', \"
znaková konštantá (končí '\0')	"abc...de"

## Operátory

prvok štruktúry *meno.clen*  
smerník do štruktúry *pointer->clen*  
inkrementácia, dekrementácia ++, --  
plus, minus, logical not, bitwise not +, -, !, ~  
hodnota cez smerník, adresa \*pointer, &meno  
typová konverzia (typ) vyraz  
zistenie veľkosti sizeof  
násobenie, delenie \*, /, %  
sčítanie, odčítanie +, -  
bitový posun vľavo, vpravo <<, >>  
porovnanie >, >=, <, <=  
porovnanie ==, !=  
bitové AND &  
bitové XOR ^  
bitové OR |  
logické AND &&  
logické OR ||  
podmienený výraz vyraz1 ? vyraz2 : vyraz3  
operátory priradenia +=, -=, \*=, ...  
oddeľovač výrazov ,  
Unárne operátory, podmienený výraz a operátory priradenie sa združujú zprava do ľava; ostatné zľava doprava.

## Smerníky a polia

objekt, na ktorý ukazuje smerník *pointer \*pointer*  
adresa objektu *meno &meno*  
deklarácia smerníka *typ \*meno*  
pole *meno[dim]*  
viacrozmerné pole *meno[dim1,...]*  
beztypový smerník *void \**  
smerník do štruktúry *pointer->clen*  
**Deklarácie smerníkov**  
smerník na *typ typ \*meno*  
funkcia *f* vracajúca smerník na *typ typ \*f()*  
smerník pf na funkciu vracajúcu *typ typ (\*pf)()*

## Štruktúry

```
struct tag {
    declarations
};
```

predpis štruktúry  
deklarácia členov

```
struct tag meno
meno.clen
pointer -> clen
```

Príklad. (\*p).x a p->x značia to isté

```
union
clen : b
```

## Inicializácia

```
premennej typ meno=value
pola typ meno[]={value1,...}
reťazca char meno[]="string"
```

## Tok riadenia

```
ukončenie príkazu ;
oddelenie bloku { }
komentáre /* */
opustenie switch, while, do, for break
ďalšia iterácia while, do, for continue
návrat z funkcie return vyraz
ukončenie vykonávania exit(arg)
Tvorba toku riadenia (if/while/for/do/switch)
```

```
if (vyraz)
    prikaz
else if (vyraz)
    prikaz
else
    prikaz
while (vyraz)
    prikaz
for (vyraz1; vyraz2; vyraz3)
    prikaz
do
    prikaz
while(vyraz);
switch (vyraz) {
    case const1:
        prikaz1
        break;
    case const2:
        prikaz2
        break;
    default:
        prikaz
        break;
}
```

## C Preprocesor

```
vloží knižničné súbory #include <filename>
vloží užívateľské súbory #include "filename"
nahradenie textu #define meno text
nahradenie makra #define meno(var) text
Príklad. #define max(A,B) ((A)>(B) ? (A) : (B))
zrušenie definície #undef meno
podmienený preklad #if, #else, #elif, #endif
je meno definované, nedefinované? #ifdef, #ifndef
meno definované? defined(meno)
pokračovanie makra na viac riadkoch \
```

## Štandardné ANSI knižnice

```
<assert.h> <ctype.h> <errno.h> <float.h> <limits.h>
<locale.h> <math.h> <setjmp.h> <signal.h> <stdarg.h>
<stddef.h> <stdio.h> <stdlib.h> <string.h> <time.h>
```

July 1998 v1.0. Copyright © 1998 Joseph H. Silverman

Permission is granted to make and distribute copies of this card provided the copyright notice and this permission notice are preserved on all copies.

Send comments and corrections to J.H. Silverman, Math. Dept., Brown Univ., Providence, RI 02912 USA. (jhs@math.brown.edu)

# C Reference Card (ANSI)

## Vstup/Výstup <stdio.h>

### Štandardný Vstup/Výstup

štandardný vstupný súbor	stdin
štandardný výstupný súbor	stdout
štandardný chybový súbor	stderr
koniec súboru	EOF
načítaj znak	getchar()
vypíš znak	putchar( <i>chr</i> )
formátovaný výpis	printf("format", <i>arg</i> <sub>1</sub> ,...)
vpíš do reťazca <i>s</i>	sprintf( <i>s</i> , "format", <i>arg</i> <sub>1</sub> ,...)
načítaj formátované dáta	scanf("format", & <i>meno</i> <sub>1</sub> ,...)
načítaj z reťazca <i>s</i>	sscanf( <i>s</i> , "format", & <i>meno</i> <sub>1</sub> ,...)
riadok do reťazca <i>s</i> (< max znakov)	gets( <i>s</i> , max)
vypíš reťazec <i>s</i>	puts( <i>s</i> )

### Vstup/Výstup do/zo súboru

deklarácia smerníka súboru	FILE * <i>fp</i>
otvorenie súboru	fopen("meno", "mode")
módy: r (čítanie), w (zápis), a (doplnenie)	
načítaj znak	getc( <i>fp</i> )
zapiš znak	putc( <i>chr</i> , <i>fp</i> )
zapiš do súboru	fprintf( <i>fp</i> , "format", <i>arg</i> <sub>1</sub> ,...)
čítaj zo súboru	fscanf( <i>fp</i> , "format", <i>arg</i> <sub>1</sub> ,...)
uzavri súbor	fclose( <i>fp</i> )
nenulová hodnota, ak nastala chyba	ferror( <i>fp</i> )
nenulová hodnota, ak nastal koniec súboru	feof( <i>fp</i> )
načítaj riadok do reťazca <i>s</i> (< max znakov)	fgets( <i>s</i> , max, <i>fp</i> )
napíš reťazec <i>s</i>	fputs( <i>s</i> , <i>fp</i> )

### Formáty pre Vstup/Výstup: "%-+ 0w.pc"

-	ľavé zarovnanie
+	výpis so znamienkom
<i>space</i>	vypíš medzeru ak nie je znamienko
0	blok vedúcich núl
<i>w</i>	minimálna šírka poľa
<i>p</i>	presnosť
<i>c</i>	znak konverzie:
<i>d, i</i>	celé číslo
<i>u</i>	neznamienkové
<i>c</i>	jediný znak
<i>s</i>	reťazec znakov
<i>f</i>	float/double
<i>e, E</i>	číslo s exponentom
<i>o</i>	osmičkové číslo
<i>x, X</i>	šestnástkové číslo
<i>p</i>	smerník
<i>n</i>	počet napísaných znakov
<i>g, G</i>	rovnaké ako <i>f</i> alebo <i>e, E</i> závisí od exponentu

## Testovanie Druhu Znakú <ctype.h>

alfa-numerický?	isalnum( <i>c</i> )
abecedný?	isalpha( <i>c</i> )
kontrolný znak?	isctrl( <i>c</i> )
desiatinná číslica?	isdigit( <i>c</i> )
viditeľný znak?	isgraph( <i>c</i> )
malé písmeno?	islower( <i>c</i> )
viditeľný znak?	isprint( <i>c</i> )
interpunkcia?	ispunct( <i>c</i> )
medzera, koniec riadku, tab...?	isspace( <i>c</i> )
veľké písmeno?	isupper( <i>c</i> )
šestnástková číslica?	isxdigit( <i>c</i> )
konvertuj na malé písmeno	tolower( <i>c</i> )
konvertuj na veľké písmeno	toupper( <i>c</i> )

## Operácie s reťazcami <string.h>

<i>s, t</i> sú reťazce, <i>cs, ct</i> sú reťazcové konštanty	
dĺžka <i>s</i>	strlen( <i>s</i> )
kopíruj <i>ct</i> do <i>s</i>	strcpy( <i>s, ct</i> )
až po <i>n</i> znakov	strncpy( <i>s, ct, n</i> )
zlúč <i>ct s s</i>	strcat( <i>s, ct</i> )
až po <i>n</i> znakov	strncat( <i>s, ct, n</i> )
porovnaj <i>cs s ct</i>	strcmp( <i>cs, ct</i> )
iba prvých <i>n</i> znakov	strncmp( <i>cs, ct, n</i> )
smerník ku prvému <i>c</i> v <i>cs</i>	strchr( <i>cs, c</i> )
smerník k poslednému <i>c</i> v <i>cs</i>	strrchr( <i>cs, c</i> )
kopíruj <i>n</i> znakov z <i>ct</i> do <i>s</i>	memcpy( <i>s, ct, n</i> )
kopíruj <i>n</i> znakov z <i>ct</i> do <i>s</i>	memmove( <i>s, ct, n</i> )
porovnaj <i>n</i> znakov z <i>cs s ct</i>	memcmp( <i>cs, ct, n</i> )
smerník ku prvému <i>c</i> v prvých <i>n</i> znakov	memchr( <i>cs, c, n</i> )
vlož <i>c</i> do prvých <i>n</i> znakov z <i>cs</i>	memset( <i>s, c, n</i> )

## Štand. univerzálne funkcie <stdlib.h>

absolútna hodnota z <i>int n</i>	abs( <i>n</i> )
absolútna hodnota z <i>long n</i>	labs( <i>n</i> )
podiel a zvyšok delenia <i>int n, d</i>	div( <i>n, d</i> )
vracia štruktúru s <i>div_t.quot</i> a <i>div_t.rem</i>	
podiel a zvyšok delenia <i>long n, d</i>	ldiv( <i>n, d</i> )
vracia štruktúru s <i>ldiv_t.quot</i> and <i>ldiv_t.rem</i>	
pseudo-náhodné celé číslo [0, RAND_MAX]	rand()
nastav náhodné počiatočné číslo <i>n</i>	srand( <i>n</i> )
ukonči vykonávanie programu	exit( <i>status</i> )
pošli reťazec <i>s</i> systému na vykonanie	system( <i>s</i> )
<b>Konverzie</b>	
preveď reťazec <i>s</i> float/double	atof( <i>s</i> )
preveď reťazec <i>s</i> int	atoi( <i>s</i> )
preveď reťazec <i>s</i> long	atol( <i>s</i> )
preveď časť reťazca <i>s</i> na double	strtod( <i>s, endp</i> )
preveď časť reťazca <i>s</i> (base <i>b</i> ) na long	strtoul( <i>s, endp, b</i> )
rovnako, ale unsigned long	strtoul( <i>s, endp, b</i> )

### Alokácia Pamäte

alokuj pamäť	calloc( <i>nobj, size</i> )
alokuj pamäť	malloc( <i>size</i> )
zmeň veľkosť objektu	realloc( <i>pts, size</i> )
uvolnenie miesta	free( <i>ptr</i> )

### Funkcie Polí

hľadaj v poli heslo	bsearch( <i>key, array, n, size, cmp()</i> )
zoraď pole vzostupne	qsort( <i>array, n, size, cmp()</i> )

## Matematické funkcie <math.h>

argumenty a vrátené hodnoty sú typu double	
trigonometrické funkcie	sin( <i>x</i> ), cos( <i>x</i> ), tan( <i>x</i> )
prevrátené trigonometrické funkcie	asin( <i>x</i> ), acos( <i>x</i> ), atan( <i>x</i> )
arctan( <i>y/x</i> )	atan2( <i>y, x</i> )
hyperbolické trigon. funkcie	sinh( <i>x</i> ), cosh( <i>x</i> ), tanh( <i>x</i> )
exponenciály & logaritmy	exp( <i>x</i> ), log( <i>x</i> ), log10( <i>x</i> )
Prevod mantisa/exponent	ldexp( <i>x, n</i> ), frexp( <i>x, *e</i> )
delenie & zvyšok	modf( <i>x, *ip</i> ), fmod( <i>x, y</i> )
mocniny	pow( <i>x, y</i> ), sqrt( <i>x</i> )
zaokrúhlenie	ceil( <i>x</i> ), floor( <i>x</i> ), fabs( <i>x</i> )

## Limity Typov Celých Čísel <limits.h>

CHAR_BIT	8	bitov v char
CHAR_MAX		max. hodnota char
CHAR_MIN	0	max. hodnota char
INT_MAX	+2147483647	max. hodnota int
INT_MIN	-2147483647	min. hodnota int
LONG_MAX	+2147483647	max. hodnota long
LONG_MIN	-2147483647	min. hodnota long
SCHAR_MAX	+127	max. hodnota char
SCHAR_MIN	-127	min. hodnota char
SHRT_MAX	+32767	max. hodnota short int
SHRT_MIN	-32767	min. hodnota short int
UCHAR_MAX	255	max. hodnota unsigned char
UINT_MAX	4294967295	max. hodnota unsigned int
ULONG_MAX	4294967295	max. hodnota unsigned long
USHRT_MAX	65536	max. hodnota unsigned short

## Funkcie Času a Dátumu <time.h>

procesorový čas použitý programom	clock()
<i>Príklad.</i> clock()/CLOCKS_PER_SEC je čas v sekundách	
aktuálny kalendárny čas	time()
čas <sub>2</sub> -čas <sub>1</sub> v sekundách (double)	difftime( <i>time</i> <sub>2</sub> , <i>time</i> <sub>1</sub> )
typy na reprezentácie času	clock_t, time_t
typ štruktúry pre formát kalendárneho času	tm
tm_sec	sekundy po minúte
tm_min	minúty po hodine
tm_hour	hodiny od polnoci
tm_mday	deň v mesiaci
tm_mon	mesiace od januára
tm_year	roky od 1900
tm_wday	dni od nedele
tm_yday	dni od 1. januára
tm_isdst	Indikácia letného času

preveď lokálny čas na kalendárny čas	mktime( <i>tp</i> )
preveď čas v <i>tp</i> na reťazec	asctime( <i>tp</i> )
preveď kalendárny čas v <i>tp</i> na lokálny čas	asctime( <i>tp</i> )
preveď kalendárny čas na GMT	gmtime( <i>tp</i> )
preveď kalendárny čas na lokálny čas	localtime( <i>tp</i> )
formátuj info dátumu a času	strftime( <i>s, smax, "format", tp</i> )
<i>tp</i> je smerník na štruktúru typu <i>tm</i>	

## Limity Typu Float <float.h>

FLT_RADIX	2	základ exponentu
FLT_ROUNDS		zaokrúhľovací mód rádovej čiarky
FLT_DIG	6	desiatinné číslice presnosti
FLT_EPSILON	1E-5	najmenšie <i>x</i> tak $1.0 + x \neq 1.0$
FLT_MANT_DIG		počet číslic v mantise
FLT_MAX	1E+37	maximálne číslo s rádovou čiarkou
FLT_MAX_EXP		maximálny exponent
FLT_MIN	1E-37	minimálne číslo s rádovou čiarkou
FLT_MIN_EXP		minimálny exponent
DBL_DIG	10	desiatinné číslice presnosti
DBL_EPSILON	1E-9	najmenšie <i>x</i> so $1.0 + x \neq 1.0$
DBL_MANT_DIG		počet číslic v mantise
DBL_MAX	1E+37	max double číslo s rádovou čiarkou
DBL_MAX_EXP		maximálny exponent
DBL_MIN	1E-37	min double číslo s rádovou čiarkou
DBL_MIN_EXP		minimálny exponent