પરિશિષ્ટ I

દશાંકી સંખ્યાની તેને સંલગ્ન દ્વિઅંકી, અષ્ટાંકી અને સોળઅંકી સંખ્યાઓ સાથેની યાદી.

દશાંકી સંખ્યા (0,1,2,...,9)	દ્વિઅંકી સંખ્યા (0,1)	અષ્ટાંકી સંખ્યા (0,1,2,...,7)	સોળઅંકી સંખ્યા (0,1,2,...,9,A,B,C,D,E,F)
1	00001	1	1
2	00010	2	2
3	00011	3	3
4	00100	4	4
5	00101	5	5
6	00110	6	6
7	00111	7	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10
17	10001	21	11
18	10010	22	12
19	10011	23	13
20	10100	24	14

પરિશિષ્ટ II

અક્ષરોની આસ્કી (ASCII) કિંમતો

ASCII		ASCII		ASCII		ASCII	
કિંમત	અક્ષર	કિંમત	અક્ષર	કિંમત	અક્ષર	કિંમત	અક્ષર
000	NUL	032	blank	064	@	096	
001	SOH	033	!	065	A	097	a
002	STX	034	"	066	B	098	b
003	ETX	035	#	067	C	099	c
004	EOT	036	$	068	D	100	d
005	ENQ	037	%	069	E	101	e
006	ACK	038	&	070	F	102	f
007	BEL	039	'	071	G	103	g
008	BS	040	(	072	H	104	h
009	HT	041	)	073	I	105	i
010	LF	042	*	074	J	106	j
011	VT	043	+	075	K	107	k
012	FF	044	,	076	L	108	l
013	CR	045	-	077	M	109	m
014	SO	046	.	078	N	110	n
015	SI	047	/	079	O	111	o
016	DLE	048	0	080	P	112	p
017	DC1	049	1	081	Q	113	q
018	DC2	050	2	082	R	114	r
019	DC3	051	3	083	S	115	s
020	DC4	052	4	084	T	116	t
021	NAK	053	5	085	U	117	u
022	SYN	054	6	086	V	118	v
023	ETB	055	7	087	W	119	w
024	CAN	056	8	088	X	120	x
025	EM	057	9	089	Y	121	y
026	SUB	058	:	090	Z	122	z
027	ESC	059	;	091	[	123	{
028	FS	060	<	092	\	124	|
029	GF	061	=	093	]	125	}
030	RS	062	>	094	?	126	~
031	US	063	?	095	-	127	DEL

33 થી 126 અક્ષરો મુદ્રણ થઈ શકે તે પ્રકારના (printable) છે. અન્ય અક્ષરો નિયંત્રણ અક્ષરો છે, જે મુદ્રિત કરી શકતા નથી.

પરિશિષ્ટ III

સી પ્રક્રિયકોનો સારાંશ

પ્રક્રિયક	પ્રક્રિયાનો ઉપયોગ	સંબંધિતતા	અગ્રતા
( )	Function call	Left to Right	First
[ ]	Array expression		
→	Structure operator		
.	Structure operator		
+	Unary plus	Right to Left	Second
-	Unary minus		
++	Increment		
- -	Decrement		
!	Logical NOT		
~	Bitwise NOT		
*	Pointer operator		
&	Address operator		
sizeof( )	Size of operand		
*	Multiplication	Left to Right	Third
/	Division		
%	Modulo division		
+	Binary addition	Left to Right	Fourth
-	Binary subtraction		
<<	Left shift	Left to Right	Fifth
>>	Right shift		
<	Less than	Left to Right	Sixth
<=	Less than equal to		
>	Greater than		
>=	Greater than equal to		
= =	Equal to	Left to Right	Seventh
!=	Not equals to		
&	Bitwise AND	Left to Right	Eighth
^	Bitwise XOR	Left to Right	Ninth
|	Bitwise OR	Left to Right	Tenth
&&	Logical AND	Left to Right	Eleventh
||	Logical OR	Left to Right	Twelfth
? :	Conditional operator	Right to Left	Thirteenth
=, *=, - =, &=, +=,			
^=, |=, <<=, >>=	Assignment Operator	Right to Left	Fourteenth
,	Comma operator	Left to Right	Fifteenth

પરિશિષ્ટ IV

નિયમિત ઉપયોગમાં લેવાતી કેટલીક હેડર ફાઈલ

આપણે જાણીએ છીએ કે, સી ભાષામાં વિધેયો ઘણા જ મહત્ત્વના છે. વિવિધ પ્રક્રિયાઓના અમલ માટે સી ભાષા આંતરસ્થાપિત (inbuilt) વિધેયોનો સંગ્રહ ધરાવે છે. આ વિધેયોનો સંગ્રહ હેડર ફાઈલમાં જૂથ બનાવીને કરવામાં આવ્યો છે. આવી હેડર ફાઈલોના સંગ્રહને સી લાઈબ્રેરી કહે છે. પ્રોગ્રામર ઉપયોગમાં લઈ શકે તે પ્રકારની કેટલીક હેડર ફાઈલોની યાદી નીચે આપવામાં આવી છે.

ફાઈલનું નામ	ઉદ્દેશ
<stdio.h>	પ્રમાણભૂત નિવેશ/નિર્ગમ માટેના વિધેયો
<ctype.h>	અક્ષરની ચકાસણી અને રૂપાંતરણ માટેના વિધેયો
<math.h>	ગાણિતિક વિધેયો
<stdlib.h>	અક્ષરોની હારમાળા (સ્ટ્રિંગ)નું રૂપાંતરણ, મેમરીની ફાળવણી અને યાદચ્છિક અંકોના નિર્માણ માટે ઉપયોગમાં લેવામાં આવતા વિવિધ પ્રકારના વિધેયો
<string.h>	અક્ષરોની હારમાળા (સ્ટ્રિંગ)ને લગતા વિધેયો

ઉપરોક્ત હેડર ફાઈલમાં સમાવવામાં આવેલા કેટલાક વિધેયો નીચે દર્શાવવામાં આવ્યા છે. વિધેયના ઉપયોગ માટે નિવેશ (input) જરૂરી છે. ઈનપુટ એ વિધેયની વ્યાખ્યાનો એક ભાગ છે અને તેને આર્ગ્યુમેન્ટ (argument) તરીકે ઓળખવામાં આવે છે. આર્ગ્યુમેન્ટના ડેટા પ્રકાર જુદા જુદા હોઈ શકે છે. આર્ગ્યુમેન્ટને ઓળખવા માટે નીચે આપેલ અક્ષરોના કોડનો ઉપયોગ કરવામાં આવે છે.

c	- char પ્રકારની આર્ગ્યુમેન્ટ
d	- double પ્રકારની આર્ગ્યુમેન્ટ
f	- file પ્રકારની આર્ગ્યુમેન્ટ
i	- int પ્રકારની આર્ગ્યુમેન્ટ
l	- long પ્રકારની આર્ગ્યુમેન્ટ
p or *	- pointer પ્રકારની આર્ગ્યુમેન્ટ
s	- string પ્રકારની આર્ગ્યુમેન્ટ
u	- unsigned પ્રકારની આર્ગ્યુમેન્ટ

કેટલાક વિધેયો જુદા-જુદા પ્રકારની કિંમતો પરત કરવાની ક્ષમતા પણ ધરાવે છે. જુદા જુદા વિધેયો, તેની આર્ગ્યુમેન્ટ અને પરત કિંમતના પ્રકાર વિશે નીચે મુજબ વિગતો આપવામાં આવી છે.

Functions in <stdio.h> file		
વિધેય	પરત કિંમત	સમજૂતી
getc(f)	int	ફાઈલ f માંથી એક અક્ષર ઉમેરવા માટે
getchar(void)	int	પ્રમાણભૂત ઈનપુટ સાધનની મદદથી એક અક્ષર ઉમેરવા માટે
gets(s)	char*	પ્રમાણભૂત ઈનપુટ સાધનની મદદથી અક્ષરોની હારમાળા (સ્ટ્રિંગ) ઉમેરવા માટે
printf(...)	int	પ્રમાણભૂત આઉટપુટ સાધન તરફ વિગતો મોકલવા માટે.
putc(c,f)	int	ફાઈલ f માં એક અક્ષર મોકલવા માટે.
putchar(c)	int	પ્રમાણભૂત આઉટપુટ સાધન તરફ એક અક્ષર મોકલવા માટે

puts(s)	int	પ્રમાણભૂત આઉટપુટ સાધન તરફ અક્ષરોની હારમાળા (સ્ટ્રિંગ) મોકલવા માટે
scanf(...)	int	પ્રમાણભૂત ઈનપુટ સાધનની મદદથી વિગતો ઉમેરવા માટે.

<ctype.h> ફાઈલમાં આવેલ વિધેયો

વિધેય	પરત કિંમત	સમજૂતી
isalnum(c)	int	આર્ગ્યુમેન્ટ અક્ષર કે અંક પ્રકારની (alphanumeric) છે કે નહીં તે ચકાસવા માટે. True માટે શૂન્યેતર અને False માટે શૂન્ય પરત કરશે
isaplpha(c)	int	આર્ગ્યુમેન્ટ અક્ષર (alphabet) છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isacii(c)	int	આર્ગ્યુમેન્ટ ASCII અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
iscntrl(c)	int	આર્ગ્યુમેન્ટ ASCII નિયંત્રણ અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isdigit(c)	int	આર્ગ્યુમેન્ટ દશાંકી પ્રકારનો અંક છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
islower(c)	int	આર્ગ્યુમેન્ટ નાના (small/lower) અક્ષરમાં છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isodigit(c)	int	આર્ગ્યુમેન્ટ અષ્ટાંકી પ્રકારનો અંક છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isprint(c)	int	આર્ગ્યુમેન્ટ મુદ્રણક્ષમ (printable) ASCII અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
ispunct(c)	int	આર્ગ્યુમેન્ટ વિરામચિહ્ન (punctuation) પ્રકારનો અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isspace(c)	int	આર્ગ્યુમેન્ટ વ્હાઈટ-સ્પેસ પ્રકારનો અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isupper(c)	int	આર્ગ્યુમેન્ટ મોટા (capital/upper) અક્ષર છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
isxdigit(c)	int	આર્ગ્યુમેન્ટ સોળઅંકી પ્રકારનો અંક છે કે નહીં તે ચકાસશે. True માટે શૂન્યેતર સંખ્યા અને False માટે શૂન્ય પરત કરશે.
toascii(c)	int	આર્ગ્યુમેન્ટની કિંમતને ASCIIમાં ફેરવવા માટે.
tolower(c)	int	અક્ષરને લોઅર કેસમાં ફેરવવા માટે.
toupper(c)	int	અક્ષરને અપર કેસમાં ફેરવવા માટે.

<math.h> ફાઈલમાં આવેલ વિધેયો

વિધેય	પરત કિંમત	સમજૂતી
acos(d)	double	d ની આર્ક કોસાઈન કિંમત પરત કરવા માટે.
asin(d)	double	dની આર્ક સાઈન કિંમત પરત કરવા માટે.
atan(d)	double	d ની આર્ક ટેન્જન્ટ કિંમત પરત કરવા માટે.
atan2(d1, d2)	double	d1/d2 ની આર્ક ટેન્જન્ટ કિંમત પરત કરવા માટે.
ceil(d)	double	આપેલ સંખ્યાથી વધુ હોય તેવી સૌથી નજીકની પૂર્ણાંક સંખ્યા પરત કરવા માટે
cos(d)	double	d ની કોસાઈન કિંમત પરત કરવા માટે
cosh(d)	double	d ની અતિપરવલય (hyperbolic) કોસાઈન કિંમત પરત કરવા માટે
exp(d)	double	d નો e ઘાતાંક પરત કરવા માટે

fabs(d)	double	d ની નિરપેક્ષ કિંમત પરત કરવા માટે
floor(d)	double	આપેલ સંખ્યાથી ઓછી હોય તેવી સૌથી નજીકની પૂર્ણાંક સંખ્યા પરત કરવા માટે
fmod(d1,d2)	double	d1/d2ની શેષ પરત કરવા માટે (d1ની નિશાની સાથે).
labs(1)	long int	1 ની નિરપેક્ષ (absolute) કિંમત પરત કરવા માટે
log(d)	double	d નો કુદરતી લઘુગુણક (natural logarithm) પરત કરવા માટે
log10(d)	double	d નો (10 આધારિત) લઘુગુણક પરત કરવા માટે.
pow(d1,d2)	double	d1 ના d2 ઘાતાંક જેટલી કિંમત પરત કરવા માટે.
sin(d)	double	d ની સાઈન કિંમત પરત કરવા માટે.
sinh(d)	double	d ની અતિપરવલય (hyperbolic) સાઈન કિંમત પરત કરવા માટે.
sqrt(d)	double	d નું વર્ગમૂળ પરત કરવા માટે.
tan(d)	double	d ની ટેન્જન્ટ કિંમત પરત કરવા માટે.
tanh(d)	double	d ની અતિપરવલય (hyperbolic) ટેન્જન્ટ કિંમત પરત કરવા માટે
<stdlib.h> ફાઈલમાં આવેલ વિધેયો		
વિધેય	**પરત કિંમત**	**સમજૂતી**
abs(i)	int	i ની નિરપેક્ષ (absolute) કિંમત પરત કરવા માટે.
atof(s)	double	સ્ટ્રિંગને બમણી ચોકસાઈ ધરાવતી (double-precision) કિંમતમાં ફેરવવા માટે
atoi(s)	int	સ્ટ્રિંગને પૂર્ણાંક સંખ્યામાં ફેરવવા માટે.
atol(s)	long	સ્ટ્રિંગને લોંગ પૂર્ણાંક સંખ્યામાં ફેરવવા માટે
calloc(u1,u2)	void*	u1 ઘટકોને u2 બાઈટ લંબાઈ ધરાવતા મેમરીના ખંડ ફાળવે છે. ફાળવણીની શરૂઆતના ખંડનું પોઈન્ટર પરત કરશે.
exit(u)	void	તમામ ફાઈલો અને બફર બંધ કરી પ્રોગ્રામ બંધ કરવા માટે. (પ્રોગ્રામના અંતની સ્થિતિ દર્શાવવા વિધેય દ્વારા u ની કિંમત ઉમેરવામાં આવશે.)
free(p)	void	p થી શરૂઆત થતી હોય તેવા મેમરીના ફાળવેલા ખંડને મુક્ત કરવા માટે.
malloc(u)	void*	u બાઈટ જેટલી મેમરી ફાળવવા માટે ફાળવવામાં આવેલ જગ્યાની શરૂઆતનું પોઈન્ટર પરત કરશે.
rand(void)	int	યાદચ્છિક (random) ધન પૂર્ણાંક પરત કરશે.
realloc(p,u)	void*	પોઈન્ટર ચલ p ને u બાઈટ જેટલી નવી મેમરી ફાળવશે. નવી મેમરી જગ્યાની શરૂઆતનું પોઈન્ટર પરત કરશે.
<string.h> ફાઈલમાં આવેલ વિધેયો		
વિધેય	**પરત કિંમત**	**સમજૂતી**
strcmp(s1, s2)	int	બે સ્ટ્રિંગને શબ્દકોશ રચના પ્રમાણે (lexicographically) સરખાવશે. જો s1 < s2 હોય તો ઋણ સંખ્યા, s1 અને s2 સરખા હોય તો 0 અને s1 > s2 હોય તો ધન સંખ્યા પરત કરશે.
strcmpi(s1,s2)	int	બે સ્ટ્રિંગના કેસને અવગણી શબ્દકોશ રચના પ્રમાણે (lexicographically) સરખાવશે. જો s1 < s2 હોય તો ઋણ સંખ્યા, s1 અને s2 સરખા હોય તો 0 અને s1 > s2 હોય તો ધન સંખ્યા પરત કરશે.
strcpy(s1, s2)	char*	s2 સ્ટ્રિંગની નકલ s1માં કરશે.
strlen(s)	int	s સ્ટ્રિંગમાં આવેલ અક્ષરોની સંખ્યા પરત કરશે.