

Appendix A

CBM Character Codes

This appendix contains the following tables:

- CBM BASIC keywords (Table A-1)
- CBM ASCII 7-bit codes (Table A-2)
- CBM screen memory 7-bit codes (Table A-3)
- CBM standard and alternate character set 8-bit codes (Table A-4)

Tables A-1, A-2 and A-3 are self-explanatory; they are referred to frequently throughout the book. The standard and alternate character sets illustrated in Table A-4 are also referred to frequently throughout the book; however, being unique to CBM computers, information presented in this table is summarized below.

The first two columns of Table A-4 show the standard and alternate character sets for the PET and CBM computers. The right three columns show each character's corresponding ASCII code and PEEK/POKE number. The characters are arranged in ascending sequence by their CBM ASCII code number. If the character does not have a CBM ASCII code number, as in the case of reverse characters, they are arranged in ascending sequence of PEEK/POKE number. Many characters appear twice because they have two CBM ASCII code numbers.

Standard Character Set. The standard character set is in effect when the PET 2001 computer is powered up, or when a value of 12 is poked into memory location 59468 by a POKE 59468,12 statement on a PET or CBM computer. The standard character set has upper-case alphabets, numbers, graphic characters and special symbols.

Alternate Character Set. The alternate character set is in effect when the CBM computer is powered up, or when a value of 14 is poked into memory location 59468 by a POKE 59468,14 statement on a PET or CBM computer. The alternate character set has upper- and lower-case alphabets, numbers, and some special symbols.

CBM ASCII Code. ASCII stands for American Standard Code for Information Interchange. Commodore Business Machines developed its own ASCII code for the CBM Computer in order to include its unique characters.

The ASCII code column (and Table A-2) shows both the decimal and hexadecimal CBM ASCII codes for each character. To find a character's ASCII code column, find the desired character in the character column, then look across the chart for the corresponding CBM ASCII code. When using the ASC() or CHR\$() function refer only to the decimal ASCII number.

The last portion of the chart, the reverse characters, do not have CBM ASCII codes. Therefore they are arranged by their PEEK/POKE numbers.

PEEK/POKE. The PEEK/POKE number is the number used when POKEing a character to the screen. It also represents the number of the character returned when PEEKing into memory to see what character is contained in a specified memory location. The PEEK/POKE numbers do not appear in strict ascending sequence until the reverse characters portion of the chart. At this point, the chart is arranged in ascending PEEK/POKE order because the reverse CBM characters lack CBM ASCII numbers, and can only be referenced with PRINT or PEEK/POKE statements.

Table A-1. CBM BASIC Keyboard Codes

Code (decimal)	Character/ Keyword	Code (decimal)	Character/ Keyword	Code (decimal)	Character/ Keyword	Code (decimal)	Character/ Keyword
0	End of line	70	F	141	GOSUB	181	INT
1-31	Unused	71	G	142	RETURN	182	ABS
32	space	72	H	143	REM	183	USR
33	!	73	I	144	STOP	184	FRE
34	"	74	J	145	ON	185	POS
35	#	75	K	146	WAIT	186	SQR
36	\$	76	L	147	LOAD	187	RND
37	%	77	M	148	SAVE	188	LOG
38	&	78	N	149	VERIFY	189	EXP
39	'	79	O	150	DEF	190	COS
40	(80	P	151	POKE	191	SIN
41)	81	Q	152	PRINT #	192	TAN
42	*	82	R	153	PRINT	193	ATN
43	+	83	S	154	CONT	194	PEEK
44	,	84	T	155	LIST	195	LEN
45	-	85	U	156	CLR	196	STR\$
46	.	86	V	157	CMD	197	VAL
47	/	87	W	158	SYS	198	ASC
48	0	88	X	159	OPEN	199	CHR\$
49	1	89	Y	160	CLOSE	200	LEFT\$
50	2	90	Z	161	GET	201	RIGHT\$
51	3	91	[162	NEW	202	MID\$
52	4	92	\	163	TAB(203	Unused
53	5	93]	164	TO	204	CONCAT†
54	6	94	↑	165	FN	205	DOPEN†
55	7	95	—	166	SPC(206	DCLOSE†
56	8	96-127	Unused	167	THEN	207	RECORD†
57	9	128	END	168	NOT	208	HEADER†
58	:	129	FOR	169	STEP	209	COLLECT†
59	;	130	NEXT	170	+	210	BACKUP†
60	<	131	DATA	171	-	211	COPY†
61	=	132	INPUT #	172	*	212	APPEND†
62	>	133	INPUT	173	/	213	DSAVE†
63	?	134	DIM	174		215	CATALOG†
64	@	135	READ	175	AND	216	RENAME†
65	A	136	LET	176	OR	217	SCRATCH†
66	B	137	GOTO	177	>	218	DIRECTORY†
67	C	138	RUN	178	=	219	?SYNTAX ERROR†
68	D	139	IF	179	<	220-254	Unused
69	E	140	RESTORE	180	SGN	255	π

† For BASIC 4.0 only

Table A-2. ASCII Standard 7-Bit Codes

<div> <div>Bit</div> <div>→</div> <div>6</div> <div>5</div> <div>4</div> <div>3</div> <div>2</div> <div>1</div> <div>0</div> </div>				0	0	0	0	1	1	1	1	1
				0	0	0	0	1	0	0	1	1
				0	1	0	1	0	0	1	0	1
0	0	0	0	NUL	DLE	SP	0	@	P	'	p	
0	0	0	1	SOH	DC1	!	1	A	Q	a	q	
0	0	1	0	STX	DC2	"	2	B	R	b	r	
0	0	1	1	ETX	DC3	#	3	C	S	c	s	
0	1	0	0	EOT	DC4	\$	4	D	T	d	t	
0	1	0	1	ENQ	NAK	%	5	E	U	e	u	
0	1	1	0	ACK	SYN	&	6	F	V	f	v	
0	1	1	1	BEL	ETB	.	7	G	W	g	w	
1	0	0	0	BS	CAN	(8	H	X	h	x	
1	0	0	1	HT	EM)	9	I	Y	i	y	
1	0	1	0	LF	SUB	*	:	J	Z	j	z	
1	0	1	1	VT	ESC	+	<	K	[k		
1	1	0	0	FF	FS	,	>	L	\	l	~	
1	1	0	1	CR	GS	-	=	M]	m	_	
1	1	1	0	SO	RS	.	>	N	^	n	o	
1	1	1	1	SI	US	/	?	O	_	o	DEL	
NUL				Null				FF				Form feed
SOH				Start of heading				CAN				End of transmission block
STX				Start of text				EM				Cancel
ETX				End of text				SUB				End of medium
EOT				End of transmission				ESC				Substitute
ENQ				Enquiry				FS				Escape
ACK				Acknowledge				GS				File separator
BEL				Bell, or alarm				RS				Group separator
BS				Backspace				US				Record separator
HT				Horizontal tabulation				SP				Unit separator
LF				Line feed				DEL				Space
VT				Vertical tabulation								Delete

Table A-3. CBM Screen Memory 7-Bit Codes

B	61	0	0	0	0	1	1	1	1
1	51	0	0	1	1	0	0	1	1
1	41	0	1	0	1	0	1	0	1
3210	1								
0000	1	Q	P		0	-	1		1
0001	1	A	Q	!	1	+	0		1
0010	1	B	R	"	2	-	1		1
0011	1	C	S	#	3	-	1		1
0100	1	D	T	\$	4	-	1		1
0101	1	E	U	%	5	-	1		1
0110	1	F	V	&	6	-	1		1
0111	1	G	W	.	7	-	1		1
1000	1	H	X	(8	-	1		1
1001	1	I	Y)	9	-	1		1
1010	1	J	Z	*	:	-	1		1
1011	1	K	[+	<	-	1		1
1100	1	L	\	,	>	-	1		1
1101	1	M]	-	=	-	1		1
1110	1	N	^	.	>	-	1		1
1111	1	O	_	/	?	-	1		1

Table A-4. PET/CBM Standard and Alternate Character Sets (Continued)

Standard Character Set		Alternate Character Set		ASCII		PEEK/ POKE	Standard Character Set		Alternate Character Set		ASCII		PEEK/ POKE
PET	CBM	PET	CBM	DEC	HEX		PET	CBM	PET	CBM	DEC	HEX	
				0	00		A	A	a	a	65	41	1
				1	01		B	B	b	b	66	42	2
				2	02		C	C	c	c	67	43	3
STOP	STOP			3	03		D	D	d	d	68	44	4
				4	04		E	E	e	e	69	45	5
				5	05		F	F	f	f	70	46	6
				6	06		G	G	g	g	71	47	7
				7	07		H	H	h	h	72	48	8
				8	08		I	I	i	i	73	49	9
				9	09		J	J	j	j	74	4A	10
				10	0A		K	K	k	k	75	4B	11
				11	0B		L	L	l	l	76	4C	12
				12	0C		M	M	m	m	77	4D	13
RETURN	RETURN			13	0D		N	N	n	n	78	4E	14
				14	0E		O	O	o	o	79	4F	15
				15	0F		P	P	p	p	80	50	16
				16	10		Q	Q	q	q	81	51	17
CRSR	CRSR			17	11		R	R	r	r	82	52	18
RVS	RVS			18	12		S	S	s	s	83	53	19
HOME	HOME			19	13		T	T	t	t	84	54	20
DELETE	DELETE			20	14		U	U	u	u	85	55	21
				21	15		V	V	v	v	86	56	22
				22	16		W	W	w	w	87	57	23
				23	17		X	X	x	x	88	58	24
				24	18		Y	Y	y	y	89	59	25
				25	19		Z	Z	z	z	90	5A	26
				26	1A		[[[[91	5B	27
				27	1B		\	\	\	\	92	5C	28
				28	1C]]]]	93	5D	29
CRSR←	CRSR←			29	1D		↑	↑	↑	↑	94	5E	30
				30	1E		←	←	←	←	95	5F	31
				31	1F						96	60	32
!	!	!	!	32	20	32	!	!	!	!	97	61	33
"	"	"	"	33	21	33	"	"	"	"	98	62	34
#	#	#	#	34	22	34	#	#	#	#	99	63	35
\$	\$	\$	\$	35	23	35	\$	\$	\$	\$	100	64	36
%	%	%	%	36	24	36	%	%	%	%	101	65	37
&	&	&	&	37	25	37	&	&	&	&	102	66	38
'	'	'	'	38	26	38	'	'	'	'	103	67	39
((((39	27	39	((((104	68	40
))))	40	28	40))))	105	69	41
*	*	*	*	41	29	41	*	*	*	*	106	6A	42
+	+	+	+	42	2A	42	+	+	+	+	107	6B	43
,	,	,	,	43	2B	43	,	,	,	,	108	6C	44
-	-	-	-	44	2C	44	-	-	-	-	109	6D	45
.	.	.	.	45	2D	45	110	6E	46
/	/	/	/	46	2E	46	/	/	/	/	111	6F	47
0	0	0	0	47	2F	47	0	0	0	0	112	70	48
1	1	1	1	48	30	48	1	1	1	1	113	71	49
2	2	2	2	49	31	49	2	2	2	2	114	72	50
3	3	3	3	50	32	50	3	3	3	3	115	73	51
4	4	4	4	51	33	51	4	4	4	4	116	74	52
5	5	5	5	52	34	52	5	5	5	5	117	75	53
6	6	6	6	53	35	53	6	6	6	6	118	76	54
7	7	7	7	54	36	54	7	7	7	7	119	77	55
8	8	8	8	55	37	55	8	8	8	8	120	78	56
9	9	9	9	56	38	56	9	9	9	9	121	79	57
:	:	:	:	57	39	57	:	:	:	:	122	7A	58
;	;	;	;	58	3A	58	;	;	;	;	123	7B	59
<	<	<	<	59	3B	59	<	<	<	<	124	7C	60
=	=	=	=	60	3C	60	=	=	=	=	125	7D	61
>	>	>	>	61	3D	61	>	>	>	>	126	7E	62
?	?	?	?	62	3E	62	?	?	?	?	127	7F	63
@	@	@	@	63	3F	63					128	80	64
				64	40	0							

Table A-4. PET/CBM Standard and Alternate Character Sets (Continued)

Standard Character Set		Alternate Character Set		ASCII		PEEK/POKE	Standard Character Set		Alternate Character Set		ASCII		PEEK/POKE
PET	CBM	PET	CBM	DEC	HEX		PET	CBM	PET	CBM	DEC	HEX	
RUN	RUN	RUN	RUN	129	81	65	↑	↑	A	A	193	C1	65
				130	82	66	↓	↓	B	B	194	C2	66
				131	83	67	—	—	C	C	195	C3	67
				132	84	68	—	—	D	D	196	C4	68
				133	85	69	—	—	E	E	197	C5	69
				134	86	70	—	—	F	F	198	C6	70
				135	87	71	—	—	G	G	199	C7	71
				136	88	72	—	—	H	H	200	C8	72
				137	89	73	—	—	I	I	201	C9	73
				138	8A	74	—	—	J	J	202	CA	74
Shifted RETURN	Shifted RETURN	Shifted RETURN	Shifted RETURN	139	8B	75	—	—	K	K	203	CB	75
				140	8C	76	—	—	L	L	204	CC	76
				141	8D	77	—	—	M	M	205	CD	77
				142	8E	78	—	—	N	N	206	CE	78
				143	8F	79	—	—	O	O	207	CF	79
				144	90	80	—	—	P	P	208	D0	80
				145	91	81	—	—	Q	Q	209	D1	81
				146	92	82	—	—	R	R	210	D2	82
				147	93	83	—	—	S	S	211	D3	83
				148	94	84	—	—	T	T	212	D4	84
CRSR RVS Off CLR Screen INSERT	CRSR RVS Off CLR Screen INSERT	CRSR RVS Off CLR Screen INSERT	CRSR RVS Off CLR Screen INSERT	149	95	85	—	—	U	U	213	D5	85
				150	96	86	—	—	V	V	214	D6	86
				151	97	87	—	—	W	W	215	D7	87
				152	98	88	—	—	X	X	216	D8	88
				153	99	89	—	—	Y	Y	217	D9	89
				154	9A	90	—	—	Z	Z	218	DA	90
				155	9B	91	—	—	[[219	DB	91
				156	9C	92	—	—	\	\	220	DC	92
				157	9D	93	—	—]]	221	DD	93
				158	9E	94	—	—	^	^	222	DE	94
CRSR—	CRSR—	CRSR—	CRSR—	159	9F	95	—	—	_	_	223	DF	95
				160	A0	96	—	—	a	a	224	E0	96
				161	A1	97	—	—	b	b	225	E1	97
				162	A2	98	—	—	c	c	226	E2	98
				163	A3	99	—	—	d	d	227	E3	99
				164	A4	100	—	—	e	e	228	E4	100
				165	A5	101	—	—	f	f	229	E5	101
				166	A6	102	—	—	g	g	230	E6	102
				167	A7	103	—	—	h	h	231	E7	103
				168	A8	104	—	—	i	i	232	E8	104
Shifted b	Shifted b	Shifted b	Shifted b	169	A9	105	—	—	j	j	233	E9	105
				170	AA	106	—	—	k	k	234		106
				171	AB	107	—	—	l	l	235		107
				172	AC	108	—	—	m	m	236		108
				173	AD	109	—	—	n	n	237		109
				174	AE	110	—	—	o	o	238		110
				175	AF	111	—	—	p	p	239		111
				176	B0	112	—	—	q	q	240		112
				177	B1	113	—	—	r	r	241		113
				178	B2	114	—	—	s	s	242		114
CRSR—	CRSR—	CRSR—	CRSR—	179	B3	115	—	—	t	t	243		115
				180	B4	116	—	—	u	u	244		116
				181	B5	117	—	—	v	v	245		117
				182	B6	118	—	—	w	w	246		118
				183	B7	119	—	—	x	x	247		119
				184	B8	120	—	—	y	y	248		120
				185	B9	121	—	—	z	z	249		121
				186	BA	122	—	—	[[250		122
				187	BB	123	—	—	\	\	251		123
				188	BC	124	—	—]]	252		124
CRSR—	CRSR—	CRSR—	CRSR—	189	BD	125	—	—	^	^	253		125
				190	BE	126	—	—	_	_	254		126
				191	BF	127	—	—	a	a	255		127
				192	CO	64	—	—	b	b			

Table A-4. PET/CBM Standard and Alternate Character Sets (Continued)

Standard Character Set		Alternate Character Set		ASCII		PEEK/ POKE	Standard Character Set		Alternate Character Set		ASCII		PEEK/ POKE
PET	CBM	PET	CBM	DEC	HEX		PET	CBM	PET	CBM	DEC	HEX	
□	□	□	□			128	■	■	■	■			192
␣	␣	a	a			129	␣	␣	␣	␣			193
␢	␢	b	b			130	␢	␢	␢	␢			194
␣	␣	c	c			131	␣	␣	␣	␣			195
␣	␣	d	d			132	␣	␣	␣	␣			196
␣	␣	e	e			133	␣	␣	␣	␣			197
␣	␣	f	f			134	␣	␣	␣	␣			198
␣	␣	g	g			135	␣	␣	␣	␣			199
␣	␣	h	h			136	␣	␣	␣	␣			200
␣	␣	i	i			137	␣	␣	␣	␣			201
␣	␣	j	j			138	␣	␣	␣	␣			202
␣	␣	k	k			139	␣	␣	␣	␣			203
␣	␣	l	l			140	␣	␣	␣	␣			204
␣	␣	m	m			141	␣	␣	␣	␣			205
␣	␣	n	n			142	␣	␣	␣	␣			206
␣	␣	o	o			143	␣	␣	␣	␣			207
␣	␣	p	p			144	␣	␣	␣	␣			208
␣	␣	q	q			145	␣	␣	␣	␣			209
␣	␣	r	r			146	␣	␣	␣	␣			210
␣	␣	s	s			147	␣	␣	␣	␣			211
␣	␣	t	t			148	␣	␣	␣	␣			212
␣	␣	u	u			149	␣	␣	␣	␣			213
␣	␣	v	v			150	␣	␣	␣	␣			214
␣	␣	w	w			151	␣	␣	␣	␣			215
␣	␣	x	x			152	␣	␣	␣	␣			216
␣	␣	y	y			153	␣	␣	␣	␣			217
␣	␣	z	z			154	␣	␣	␣	␣			218
␣	␣	␣	␣			155	␣	␣	␣	␣			219
␣	␣	␣	␣			156	␣	␣	␣	␣			220
␣	␣	␣	␣			157	␣	␣	␣	␣			221
␣	␣	␣	␣			158	␣	␣	␣	␣			222
␣	␣	␣	␣			159	␣	␣	␣	␣			223
␣	␣	␣	␣			160	␣	␣	␣	␣			224
␣	␣	␣	␣			161	␣	␣	␣	␣			225
␣	␣	␣	␣			162	␣	␣	␣	␣			226
␣	␣	␣	␣			163	␣	␣	␣	␣			227
␣	␣	␣	␣			164	␣	␣	␣	␣			228
␣	␣	␣	␣			165	␣	␣	␣	␣			229
␣	␣	␣	␣			166	␣	␣	␣	␣			230
␣	␣	␣	␣			167	␣	␣	␣	␣			231
␣	␣	␣	␣			168	␣	␣	␣	␣			232
␣	␣	␣	␣			169	␣	␣	␣	␣			233
␣	␣	␣	␣			170	␣	␣	␣	␣			234
␣	␣	␣	␣			171	␣	␣	␣	␣			235
␣	␣	␣	␣			172	␣	␣	␣	␣			236
␣	␣	␣	␣			173	␣	␣	␣	␣			237
␣	␣	␣	␣			174	␣	␣	␣	␣			238
␣	␣	␣	␣			175	␣	␣	␣	␣			239
␣	␣	␣	␣			176	␣	␣	␣	␣			240
␣	␣	␣	␣			177	␣	␣	␣	␣			241
␣	␣	␣	␣			178	␣	␣	␣	␣			242
␣	␣	␣	␣			179	␣	␣	␣	␣			243
␣	␣	␣	␣			180	␣	␣	␣	␣			244
␣	␣	␣	␣			181	␣	␣	␣	␣			245
␣	␣	␣	␣			182	␣	␣	␣	␣			246
␣	␣	␣	␣			183	␣	␣	␣	␣			247
␣	␣	␣	␣			184	␣	␣	␣	␣			248
␣	␣	␣	␣			185	␣	␣	␣	␣			249
␣	␣	␣	␣			186	␣	␣	␣	␣			250
␣	␣	␣	␣			187	␣	␣	␣	␣			251
␣	␣	␣	␣			188	␣	␣	␣	␣			252
␣	␣	␣	␣			189	␣	␣	␣	␣			253
␣	␣	␣	␣			190	␣	␣	␣	␣			254
␣	␣	␣	␣			191	␣	␣	␣	␣			255

Appendix B

CBM Error Messages

Error messages may be displayed in response to just about anything you key in at the CBM keyboard or when your program is running. Both the CBM BASIC interpreter and the operating system issue error messages, listed separately below.

Whenever the CBM BASIC interpreter detects an error, it displays a diagnostic message, headed by a question mark, in the general form:

?message ERROR IN LINE number

where message is the type of error (listed alphabetically below) and number is the line number in the program where the error occurred (not present in immediate mode). Following any error message, BASIC returns to immediate mode and gives the READY prompt.

CBM BASIC error messages are listed below, with two descriptive paragraphs: The first describes the cause of the error, and the second discusses possible ways of correcting the error.

BASIC ERROR MESSAGES

Error Message	Cause and Suggested Remedies
BAD SUBSCRIPT	An attempt was made to reference an array element that is outside the dimensions of the array. This may happen by specifying the wrong number of dimensions (different from the DIM statement), using a subscript larger than specified in the DIM statement or using a subscript larger than 10 for a non-dimensioned array.

Correct the array element number to remain within the original dimensions, or change the array size to allow more elements.

CAN'T CONTINUE

A CONT command was issued, but program execution cannot be resumed because the program has been altered, added to or cleared in immediate mode, or execution was stopped by an error. Program execution cannot be continued past an error message.

Correct the error. The most prudent course is to type RUN and start over. However, you can attempt to reenter the program at the point of interruption by a directed GOTO.

DIVISION BY ZERO

An attempt was made to perform a division operation with a divisor of zero. Dividing by zero is not allowed.

Check the values of variables (or constants!) in the indicated line number. Change the program so that the divisor can never be evaluated to zero or add a check for zero before performing the division.

FORMULA TOO COMPLEX

This is not a program error but indicates that a string expression in the program is too intricate for CBM BASIC to handle.

Break the indicated expression into two or more parts and rerun the program (this will also tend to improve program readability).

ILLEGAL DIRECT

A command was given in immediate mode that is valid only in program mode. The following are invalid in immediate mode: DATA, DEF FN, GET, GET#, INPUT, INPUT#.

Enter the desired operation as a (short) program and RUN it.

ILLEGAL QUANTITY

A function is passed one or more parameters that are out of range. This message also occurs if the USR function is referenced before storing the subroutine address at memory locations 1 and 2.

Check the ranges given in Chapter 8 for the function in question. Change the program to be sure that the argument will always be within range, or add a check before the function reference to make sure that the argument is allowed. If USR error, insert statements to POKE the subroutine address before the USR reference.

NEXT WITHOUT FOR

A NEXT statement is encountered that is not tied to a preceding FOR statement. Either there is no FOR statement or the variable in the NEXT statement is not in a corresponding FOR statement.

OUT OF DATA

The FOR part of a FOR-NEXT loop must be inserted or the offending NEXT statement deleted. Be sure that the index variables are the same at both ends of the loop.

A READ statement is executed but all of the DATA statements in the program have already been read. For each variable in a READ statement, there must be a corresponding DATA element.

Add more DATA elements or restrict the number of READs to the current number of DATA elements. Insert a RESTORE statement to reread the existing data. Or add a flag at the end of the last DATA statement (any value not used as a DATA element may be used for the flag value) and stop READING when the flag has been read.

OUT OF MEMORY

The user program area of memory has been filled and a request is given to put more in, e.g., add a line to the program. This message may also be caused by multiple FOR - NEXT and/or GOSUB nestings that fill up the Stack; this is the case if ?FRE(0) shows considerable program area storage left.

Simplify the program. Pay particular attention to reducing array sizes. It may be necessary to restructure the program into overlays.

OVERFLOW

A calculation has resulted in a number outside the allowable range, i.e., the number is too big. The largest number allowed is 1.70141184E+38.

Check your calculations. It may be possible to eliminate this error just by changing the order in which the calculations are programmed.

REDIM'D ARRAY

An array name appears in more than one DIM statement. This error also occurs if an array name is used (given a default size of 11) and later appears in a DIM statement.

Place DIM statements near the beginning of the program. Check to see that each DIM statement is executed only once. DIM must not appear inside a FOR - NEXT loop or in a subroutine where either may be executed more than once.

REDO FROM START

This is a diagnostic message during an INPUT statement operation and is not a fatal error. It indicates that the wrong type of data (string for numeric or vice versa) was entered in response to an INPUT request.

Reenter the correct type data. INPUT will continue prompting until an acceptable response is entered.

RETURN WITHOUT GOSUB

A RETURN statement was encountered without a previous matching GOSUB statement being executed.

Insert a GOSUB statement or delete the RETURN statement. The error may be caused by dropping into the subroutine code inadvertently. In this case correct the program flow. An END or STOP statement placed just ahead of the subroutine serves as a debugging aid.

STRING TOO LONG

An attempt was made by use of the concatenation operator (+) to create a string longer than 255 characters.

Break the string into two or more shorter strings as part of the program operation. Use the LEN function to check string lengths before concatenating them.

SYNTAX

There is a syntax error in the line just entered (immediate mode) or scanned for execution (program mode). This is the most common error message, and is caused by such things as misspellings, incorrect punctuation, unmatched parentheses, extraneous characters, etc.

Examine the line carefully and make corrections. Note that syntax errors in a program are diagnosed at run time, not at the time the lines are entered from the keyboard. You can eliminate many syntax error messages by carefully scrutinizing newly entered program lines before running the program.

TYPE MISMATCH

An attempt was made to enter a string into a numeric Assignment variable or vice versa, or an incorrect type was given as a function parameter.

Change the offending item to correct type. Refer to Chapter 8 for acceptable parameter types.

UNDEF'D STATEMENT

An attempt was made to branch to a nonexistent line number.

Insert a statement with the necessary line number or branch to another line number.

UNDEF'D FUNCTION

Reference was made to a user defined function that has not previously been defined by appearing in a DEF FN statement. The definition must precede the function reference.

Define the function. Place DEF FN statements near the beginning of the program.

OPERATING SYSTEM ERROR MESSAGES**BAD DATA**

String data was input when numeric data was expected.

BAD DISK

Correct the input data to numeric, or change the program to accept string input.

A media failure on a **HEADER** command, due to either the diskette missing from the drive, a write protect tab, or a defective magnetic surface.

Check the disk drive to see if a diskette is properly inserted. Remove write protect tab if present. If magnetic surface is defective, use a different diskette (**BASIC 4.0**).

DEVICE NOT PRESENT

No device on the **IEEE 488 Bus** was present to handshake an attention sequence. The Status function will have a value of 2, indicating a timeout. This message may occur for any I/O command.

If the device identification is in error, correct the **OPEN** (or other) statement. If the statement is correct, especially if it has worked before, check the addressed device for malfunction, misconnection, or power off.

FILE ALREADY EXISTS

The name of the source file being copied with the **COPY** statement already exists on the destination diskette.

Delete the file on the destination diskette before attempting to **COPY**, or use a different diskette as the destination diskette.

FILE NOT FOUND

The filename given in the **LOAD** or **OPEN** statement was not found on the specified device.

Check that you have the correct tape or diskette in the device. Check the filenames on the tape or diskette for possible spelling error in the program statement.

FILE NOT OPEN

An attempt was made to access a file that was not opened via the **OPEN** statement.

Open the file.

FILE OPEN

An attempt was made to open a file that has already been opened via a previous **OPEN** statement.

Check the logical file number (first parameter in the **OPEN** statement) to be sure a different number is used for each file. Insert a **CLOSE** statement if you want to reopen the same file for a different I/O operation.

LOAD

An unacceptable number of tape errors were accumulated on a tape load (more than 31) that were not cleared on reading the redundant block. This message is issued in connection with the **LOAD** command (see Chapter 4).

NOT INPUT FILE

An attempt was made to read from a tape file that has been opened for output only.

Check the READ# and OPEN statement parameters for correctness. Reading requires a zero as the third parameter of the OPEN statement (this is the default option).

NOT OUTPUT FILE

An attempt was made to write to a tape file that has been opened for input only.

Check the PRINT# and OPEN statement parameters for correctness. Writing to a file requires a 1 (or a 2 if you want an EOT at the end of the file) as the third parameter in the OPEN statement.

VERIFY ERROR

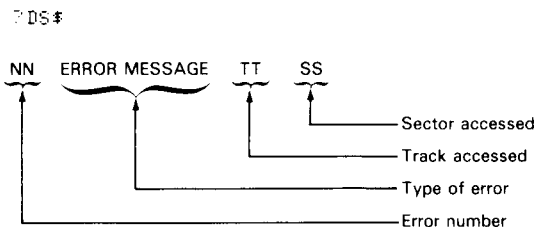
The program in memory and the specified file do not compare. This message is issued in connection with the VERIFY command (see Chapter 8).

DOS ERROR MESSAGES

REQUESTING ERROR MESSAGES

To request error messages under BASIC 4.0, execute a PRINT statement to display numeric variable DS or string variable DS\$.

DS\$ displays four parameters as follows:



Using BASIC <3.0 you cannot access variables DS or DS\$. To examine error status, you must OPEN a logical file specifying physical unit 8 with secondary address 15. You must then input four string variables and display them. This may be illustrated as follows:

```

10 OPEN 1,8,15
20 INPUT #1, A$, B$, C$, D$
30 PRINT A$, B$, C$, D$
40 CLOSE 1
  
```

A\$, is the error message number, B\$ is the error message, C\$ is the track number, and D\$ is the sector number.

Table B-1 includes the track number and sector number for all DOS errors.

Table B-1. DOS Error Messages

	Error Number	Error Message	Track	Sector
Status Messages	00	OK	00	00
	01	FILES SCRATCHED	# FILES	00
Read Errors	20	READ ERROR (Block header not found)	T	S
	21	READ ERROR (No synch character)	T	S
	22	READ ERROR (Data block not present)	T	S
	23	READ ERROR (Checksum error in data block)	T	S
	24	READ ERROR (Byte decoding error)	T	S
	27	READ ERROR (Checksum error in header)	T	S
Write Errors	25	WRITE ERROR (Write-verify error)	T	S
	26	WRITE PROTECT ON	T	S
	28	WRITE ERROR (Long data block)	T	S
	29	DISK ID MISMATCH	T	S
Syntax Errors	30	SYNTAX ERROR (General syntax)	00	00
	31	SYNTAX ERROR (Invalid command)	00	00
	32	SYNTAX ERROR (Long line)	00	00
	33	SYNTAX ERROR (Invalid file name)	00	00
	34	SYNTAX ERROR (No file given)	00	00
	39	SYNTAX ERROR (Invalid DOS command)	00	00
	50	SYNTAX ERROR (Record not present)	00	00
	51	SYNTAX ERROR (Overflow in record)	T	S
	52	SYNTAX ERROR (File too large)	T	S
File Errors	60	WRITE FILE OPEN	00	00
	61	FILE NOT OPEN	00	00
	62	FILE NOT FOUND	00	00
	63	FILE EXISTS	00	00
	64	FILE TYPE MISMATCH	00	00
	65	NO BLOCK	T	S
	66	ILLEGAL TRACK AND SECTOR	T	S
	67	ILLEGAL SYSTEM TRACK AND SECTOR	T	S
System Errors	70	NO CHANNEL	00	00
	71	DIR ERROR	00	00
	72	DISK FULL	00	00
	73	DOS MISMATCH	00	00
	74	DRIVE NOT READY	00	00

READ ERRORS

Error Message Error Message
Number

Cause of Error

20 Block header
not found

The disk controller is unable to locate the header of the requested data block. Caused by an illegal sector number, or the header has been destroyed.

21 No synch
character

The disk controller is unable to detect a synch mark on the desired track. Caused by misalignment of the read/write head or no diskette is present. Can also indicate a hardware failure.

22	Data block not present	The disk controller has been requested to read or verify a data block that was not properly written. This error message occurs in conjunction with the BLOCK commands and indicates an illegal track and/or sector request.
23	Checksum error in data block	This error message indicates that there is an error in one or more of the data bytes. The data has been read into the DOS memory, but the checksum over the data is in error. This message may also indicate grounding problems.
24	Byte decoding error	The data or header has been read into the DOS memory, but a hardware error has been created due to an invalid bit pattern in the data byte. This message may also indicate grounding problems.
27	Checksum error in header	The controller has detected an error in the header of the requested data block. The block has not been read into the DOS memory. This message may also indicate grounding problems.

WRITE ERRORS

Error Message Number	Error Message	Cause of Error
25	Write-verify error	This message is generated if the controller detects a mismatch between the written data and the data in the DOS memory.
26	WRITE PROTECT ON	This message is generated when the controller has been requested to write a data block while the write protect switch is depressed. Typically, this is caused by using a diskette with a write protect tab over the notch.
28	Long data block	The controller attempts to detect the synch mark of the next header after writing a data block. If the synch mark does not appear within a pre-determined time, the error message is generated. The error is caused by a bad diskette format (the data extends into the next block), or by hardware failure.
29	DISK ID MISMATCH	This message is generated when the controller has been requested to access a diskette which has not been initialized. This message can also occur if a diskette has a bad header.

SYNTAX ERRORS

Error Message Number	Error Message	Cause of Error
30	General syntax	The DOS cannot interpret the command sent to the command channel. Typically, this is caused by an illegal number of file names, or patterns are illegally used. For example, two file names may appear on the left side of the COPY command.
31	Invalid command	The DOS does not recognize the command. The command must start in the first position.
32	Long line	The command sent is longer than 40 characters.
33	Invalid file name	Pattern matching is invalidly used in the OPEN or SAVE command.
34	No file given	The file name was left out of a command or the DOS does not recognize it as such. Typically, a quotation mark (") or colon (:) has been left out of the command.
39	Invalid DOS Command	An unrecognizable disk operating system command was received.
50	Record not present	An INPUT# or GET# statement selected a record beyond the current end of file. This is an error if you are attempting to read a record; it is not necessarily an error if you are positioning to the end of a file in order to add new records to an old file.
51	Overflow in Record	A PRINT# statement attempted to write more than the allowed number of characters to a relative file. The terminating carriage return is counted as one character when computing record length.
52	File too large	The current record position will result in disk overflow on the next write-to-disk operation.

FILE ERRORS

Error Message Number	Error Message	Cause of Error
60	WRITE FILE OPEN	This message is generated when a write file that has not been closed is being opened for reading.
61	FILE NOT OPEN	This message is generated when a file is being accessed that has not been opened in the DOS. Sometimes, in this case, a message is not generated; the request is simply ignored.
62	FILE NOT FOUND	The requested file does not exist on the indicated drive.
63	FILE EXISTS	The file name of the file being created already exists on the diskette.
64	FILE TYPE MISMATCH	The file type does not match the file type in the directory entry for the requested file.
65	NO BLOCK	This message occurs in conjunction with the B-A command. It indicates that the block to be allocated has been previously allocated. The parameters indicate the next higher in number available track and sector. If the parameters are zero (0), then all blocks higher in number are in use.
66	ILLEGAL TRACK AND SECTOR	An attempt has been made to access a sector that does not physically exist. The track and/or sector number specified is outside of the allowed range for the current diskette. Unless you are using random access files, you should never see this error code.
67	ILLEGAL SYSTEM TRACK AND SECTOR	When accessing program or data files, an attempt has been made to access a sector that is reserved for use by the disk operating system.

SYSTEM ERRORS

Error Message Number	Error Message	Cause of Error
70	NO CHANNEL (available)	The requested channel is not available, or all channels are in use. A maximum of five sequential files may be opened at one time to the DOS. Direct access channels may have six opened files.
71	DIR(ectory) ERROR	The BAM does not match the internal count. There is a problem in the BAM allocation or the BAM has been overwritten in DOS memory. To correct this problem, reinitialize the diskette to restore the BAM in memory. Some active files may be terminated by the corrective action.
72	DISK FULL	Either the blocks on the diskette are used or the directory is at its limit (152 entries).
73	DOS MISMATCH	Data written to a diskette using any one version of DOS may be read using any other version of DOS. However, you must write to a diskette using the same DOS version with which the diskette was initialized. Error 73 is reported if you attempt to write to a diskette using a different version of DOS from the one which created and initialized the diskette.
74	DRIVE NOT READY	An attempt has been made to access the 8050 diskette unit with the selected drive.

Appendix C

BASIC Bibliography

- Advanced BASIC.* James S. Coan, Hayden Book Co., Rochelle Park, New Jersey.
- BASIC.* Albrecht, Finkle, and Brown, Peoples Computer Company, Menlo Park, California, 1967.
- BASIC: A Computer Programming Language.* C. Pegels, Holden-Day, Inc., 1973.
- Basic BASIC.* James S. Coan, Hayden Book Company, Rochelle Park, New Jersey.
- BASIC Programming.* J. Kemeny and T. Kurtz, Peoples Computer Company, Menlo Park, California, 1967.
- Entering BASIC.* J. Sack and J. Meadows, Science Research Associates, 1973.
- A Guided Tour of Computer Programming in BASIC.* T. Dwyer, Houghton Mifflin Company, 1973.
- Hands-On BASIC with a PET.* Herbert D. Peckham, McGraw-Hill Book Company, New York, 1979.
- Programming Time Shared Computers in BASIC.* Eugene H. Barnett, Wiley-Interscience, Library of Congress #72-175789.
- What to Do After You Hit Return.* Peoples Computer Company, Menlo Park, California 94025.

Appendix D

CBM Newsletters and References

This appendix contains a listing of CBM-related publications for CBM users who want to seek out continuing sources of information on the CBM computer. Many of these sources contain notices of PET/CBM user groups and activities. No endorsement of these publications is implied.

Periodicals

Calculators/Computers Magazine, Box 310, Menlo Park, California 94025. Bimonthly. \$10.00 year. A magazine that has several PET articles in each issue.

Commodore PET Users Club Newsletter, Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, California 95051. Monthly. \$15.00 year U.S., \$25.00 year foreign. Official Commodore newsletter in U.S.

Commodore PET Users Club Newsletter, Commodore Systems, 360 Eusten Rd., London, England NW1 3BL. Bimonthly. £10. Official Commodore newsletter in Europe.

COMPUTE!, P.O. Box 5406, Greensboro, North Carolina, 27403. Monthly. \$16.00 year U.S., \$18.00 year Canada, \$20.00 year elsewhere. Each issue has a regular section on PET/CBM products.

CURSOR, P.O. Box 550, Goleta, California 93017. Monthly. \$33.00 year. A cassette magazine — you receive a tape cassette of programs that can be loaded into the CBM. Each cassette comes with a 2-page newsletter/program description.

MICRO, The 6502 Journal, 8 Fourth Lane, South Chelmsford, Massachusetts 01824. Bimonthly. Single copies \$1.50, \$6.00 year. A magazine that has several CBM articles in each issue. For the experienced CBM user.

People's Computers, 1263 El Camino Real, Box E, Menlo Park, California 94025. Bimonthly. Single copies \$1.50, \$8.00 year. A magazine that has several CBM articles in each issue.

PET Users Group Newsletter, Lawrence Hall of Science, University of California, Berkeley, California 94720. Monthly. \$4.50 for 6 integral issues, checks payable to Regents of the University of California. Highly recommended.

Purser's Reference List of Computer Cassettes. Quarterly. Single copy \$4.00 domestic, \$5.00 foreign. \$12.00 year domestic, \$16.00 year foreign. Extensive list of CBM programs available on cassette.

Reference Manuals

CBM Floppy Disk User Manual Model 2040, Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, California 95051, 1979. By the manufacturers of the CBM computer.

CBM Printer User Manual Models 2022 & 2023, Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, California 95051, 1979. By the manufacturers of the CBM computer.

Commodore Business Computer User's Guide Series 8000, Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, California 95051, 1980. By the manufacturers of the CBM computer.

MCS6500 Microcomputer Family Programming Manual, MOS Technology, Inc., 950 Rittenhouse Road, Norristown, Pennsylvania 19401. \$10.00 (price may vary with location). By the manufacturers of the 6502 microprocessor.

MCS6500 Microcomputer Family Hardware Manual, MOS Technology, Inc., 950 Rittenhouse Road, Norristown, Pennsylvania 19401. \$10.00 (price may vary with location). By the manufacturers of the 6502 microprocessor.

PET and the IEEE 488 Bus (GPIB), E. Fisher and C. W. Jensen, Osborne/McGraw-Hill, 630 Bancroft Way, Berkeley, California 94710, 1980. \$15.99.

PET 2001-8 Personal Computer User Manual, Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, California 95051. (8K system). \$9.95. By the manufacturers of the PET computer.

PET 2001-16, 16N, 32, 32N Personal Computer User Manual, Commodore Business Machines, Inc., 3330 Scott Blvd., Santa Clara, California 95051. (16K and 32K systems.) \$9.95. By the manufacturers of the PET computer.

6502 Assembly Language Programming, Lance Leventhal, Osborne/McGraw-Hill, 630 Bancroft Way, Berkeley, California 94710, 1979. \$16.99.

Appendix E

Conversion Tables

This appendix contains the following reference tables:

- Hexadecimal-Decimal Integer Conversion
- Powers of Two
- Mathematical Constants
- Powers of Sixteen
- Powers of Ten

HEXADECIMAL-DECIMAL INTEGER CONVERSION

The table below provides for direct conversions between hexadecimal integers in the range 0-FFF and decimal integers in the range 0-4095. For conversion of larger integers, the table values may be added to the following figures:

Hexadecimal	Decimal	Hexadecimal	Decimal
01 000	4 096	20 000	131 072
02 000	8 192	30 000	196 608
03 000	12 288	40 000	262 144
04 000	16 384	50 000	327 680
05 000	20 480	60 000	393 216
06 000	24 576	70 000	458 752
07 000	28 672	80 000	524 288
08 000	32 768	90 000	589 824
09 000	36 864	A0 000	655 360
0A 000	40 960	B0 000	720 896
0B 000	45 056	C0 000	786 432
0C 000	49 152	D0 000	851 968
0D 000	53 248	E0 000	917 504
0E 000	57 344	F0 000	983 040
0F 000	61 440	100 000	1 048 576
10 000	65 536	200 000	2 097 152
11 000	69 632	300 000	3 145 728
12 000	73 728	400 000	4 194 304
13 000	77 824	500 000	5 242 880
14 000	81 920	600 000	6 291 456
15 000	86 016	700 000	7 340 032
16 000	90 112	800 000	8 388 608
17 000	94 208	900 000	9 437 184
18 000	98 304	A00 000	10 485 760
19 000	102 400	B00 000	11 534 336
1A 000	106 496	C00 000	12 582 912
1B 000	110 592	D00 000	13 631 488
1C 000	114 688	E00 000	14 680 064
1D 000	118 784	F00 000	15 728 640
1E 000	122 880	1 000 000	16 777 216
1F 000	126 976	2 000 000	33 554 432

Hexadecimal fractions may be converted to decimal fractions as follows:

- Express the hexadecimal fraction as an integer times 16^{-n} , where n is the number of significant hexadecimal places to the right of the hexadecimal point.

$$0. CA9BF3_{16} = CA9BF3_{16} \times 16^{-6}$$

- Find the decimal equivalent of the hexadecimal integer:

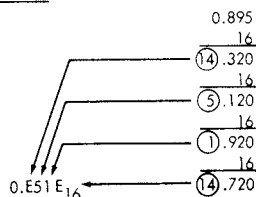
$$CA9BF3_{16} = 13\,278\,195_{10}$$

- Multiply the decimal equivalent by 16^{-n}

$$\begin{array}{r} 13\,278\,195 \\ \times 596\,046\,448 \times 10^{-16} \\ \hline 0.791\,442\,096_{10} \end{array}$$

Decimal fractions may be converted to hexadecimal fractions by successively multiplying the decimal fraction by 16_{10} . After each multiplication, the integer portion is removed to form a hexadecimal fraction by building to the right of the hexadecimal point. However, since decimal arithmetic is used in this conversion, the integer portion of each product must be converted to hexadecimal numbers.

Example: Convert 0.895_{10} to its hexadecimal equivalent



	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00	0000	0001	0002	0003	0004	0005	0006	0007	0008	0009	0010	0011	0012	0013	0014	0015
01	0016	0017	0018	0019	0020	0021	0022	0023	0024	0025	0026	0027	0028	0029	0030	0031
02	0032	0033	0034	0035	0036	0037	0038	0039	0040	0041	0042	0043	0044	0045	0046	0047
03	0048	0049	0050	0051	0052	0053	0054	0055	0056	0057	0058	0059	0060	0061	0062	0063
04	0064	0065	0066	0067	0068	0069	0070	0071	0072	0073	0074	0075	0076	0077	0078	0079
05	0080	0081	0082	0083	0084	0085	0086	0087	0088	0089	0090	0091	0092	0093	0094	0095
06	0096	0097	0098	0099	0100	0101	0102	0103	0104	0105	0106	0107	0108	0109	0110	0111
07	0112	0113	0114	0115	0116	0117	0118	0119	0120	0121	0122	0123	0124	0125	0126	0127
08	0128	0129	0130	0131	0132	0133	0134	0135	0136	0137	0138	0139	0140	0141	0142	0143
09	0144	0145	0146	0147	0148	0149	0150	0151	0152	0153	0154	0155	0156	0157	0158	0159
0A	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175
0B	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191
0C	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207
0D	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223
0E	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239
0F	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
10	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271
11	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287
12	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303
13	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319
14	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335
15	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351
16	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367
17	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383
18	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399
19	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415
1A	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431
1B	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447
1C	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463
1D	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479
1E	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495
1F	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511
20	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527
21	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543
22	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559
23	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575
24	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591
25	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607
26	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623
27	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639
28	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655
29	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671
2A	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687
2B	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703
2C	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719
2D	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735
2E	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751
2F	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767
30	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783
31	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799
32	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815
33	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831
34	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847
35	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863
36	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879
37	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895
38	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911
39	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927
3A	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943
3B	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959
3C	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975
3D	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991
3E	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007
3F	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
4C	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039
41	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055
42	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071
43	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087
44	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103
45	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119
46	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135
47	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151
48	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167
49	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183
4A	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199
4B	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215
4C	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231
4D	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247
4E	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263
4F	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279
50	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295
51	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311
52	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327
53	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343
54	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359
55	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375
56	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391
57	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407
58	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423
59	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439
5A	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455
5B	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471
5C	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487
5D	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503
5E	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518	1519
5F	1520	1521	1522	1523	1524	1525	1526	1527	1528	1529	1530	1531	1532	1533	1534	1535
60	1536	1537	1538	1539	1540	1541	1542	1543	1544	1545	1546	1547	1548	1549	1550	1551
61	1552	1553	1554	1555	1556	1557	1558	1559	1560	1561	1562	1563	1564	1565	1566	1567
62	1568	1569	1570	1571	1572	1573	1574	1575	1576	1577	1578	1579	1580	1581	1582	1583
63	1584	1585	1586	1587	1588	1589	1590	1591	1592	1593	1594	1595	1596	1597	1598	1599
64	1600	1601	1602	1603	1604	1605	1606	1607	1608	1609	1610	1611	1612	1613	1614	1615
65	1616	1617	1618	1619	1620	1621	1622	1623	1624	1625	1626	1627	1628	1629	1630	1631
66	1632	1633	1634	1635	1636	1637	1638	1639	1640	1641	1642	1643	1644	1645	1646	1647
67	1648	1649	1650	1651	1652	1653	1654	1655	1656	1657	1658	1659	1660	1661	1662	1663
68	1664	1665	1666	1667	1668	1669	1670	1671	1672	1673	1674	1675	1676	1677	1678	1679
69	1680	1681	1682	1683	1684	1685	1686	1687	1688	1689	1690	1691	1692	1693	1694	1695
6A	1696	1697	1698	1699	1700	1701	1702	1703	1704	1705	1706	1707	1708	1709	1710	1711
6B	1712	1713	1714	1715	1716	1717	1718	1719	1720	1721	1722	1723	1724	1725	1726	1727
6C	1728	1729	1730	1731	1732	1733	1734	1735	1736	1737	1738	1739	1740	1741	1742	1743
6D	1744	1745	1746	1747	1748	1749	1750	1751	1752	1753	1754	1755	1756	1757	1758	1759
6E	1760	1761	1762	1763	1764	1765	1766	1767	1768	1769	1770	1771	1772	1773	1774	1775
6F	1776	1777	1778	1779	1780	1781	1782	1783	1784	1785	1786	1787	1788	1789	1790	1791

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
70	1792	1793	1794	1795	1796	1797	1798	1799	1800	1801	1802	1803	1804	1805	1806	1807
71	1808	1809	1810	1811	1812	1813	1814	1815	1816	1817	1818	1819	1820	1821	1822	1823
72	1824	1825	1826	1827	1828	1829	1830	1831	1832	1833	1834	1835	1836	1837	1838	1839
73	1840	1841	1842	1843	1844	1845	1846	1847	1848	1849	1850	1851	1852	1853	1854	1855
74	1856	1857	1858	1859	1860	1861	1862	1863	1864	1865	1866	1867	1868	1869	1870	1871
75	1872	1873	1874	1875	1876	1877	1878	1879	1880	1881	1882	1883	1884	1885	1886	1887
76	1888	1889	1890	1891	1892	1893	1894	1895	1896	1897	1898	1899	1900	1901	1902	1903
77	1904	1905	1906	1907	1908	1909	1910	1911	1912	1913	1914	1915	1916	1917	1918	1919
78	1920	1921	1922	1923	1924	1925	1926	1927	1928	1929	1930	1931	1932	1933	1934	1935
79	1936	1937	1938	1939	1940	1941	1942	1943	1944	1945	1946	1947	1948	1949	1950	1951
7A	1952	1953	1954	1955	1956	1957	1958	1959	1960	1961	1962	1963	1964	1965	1966	1967
7B	1968	1969	1970	1971	1972	1973	1974	1975	1976	1977	1978	1979	1980	1981	1982	1983
7C	1984	1985	1986	1987	1988	1989	1990	1991	1992	1993	1994	1995	1996	1997	1998	1999
7D	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015
7E	2016	2017	2018	2019	2020	2021	2022	2023	2024	2025	2026	2027	2028	2029	2030	2031
7F	2032	2033	2034	2035	2036	2037	2038	2039	2040	2041	2042	2043	2044	2045	2046	2047
80	2048	2049	2050	2051	2052	2053	2054	2055	2056	2057	2058	2059	2060	2061	2062	2063
81	2064	2065	2066	2067	2068	2069	2070	2071	2072	2073	2074	2075	2076	2077	2078	2079
82	2080	2081	2082	2083	2084	2085	2086	2087	2088	2089	2090	2091	2092	2093	2094	2095
83	2096	2097	2098	2099	2100	2101	2102	2103	2104	2105	2106	2107	2108	2109	2110	2111
84	2112	2113	2114	2115	2116	2117	2118	2119	2120	2121	2122	2123	2124	2125	2126	2127
85	2128	2129	2130	2131	2132	2133	2134	2135	2136	2137	2138	2139	2140	2141	2142	2143
86	2144	2145	2146	2147	2148	2149	2150	2151	2152	2153	2154	2155	2156	2157	2158	2159
87	2160	2161	2162	2163	2164	2165	2166	2167	2168	2169	2170	2171	2172	2173	2174	2175
88	2176	2177	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191
89	2192	2193	2194	2195	2196	2197	2198	2199	2200	2201	2202	2203	2204	2205	2206	2207
8A	2208	2209	2210	2211	2212	2213	2214	2215	2216	2217	2218	2219	2220	2221	2222	2223
8B	2224	2225	2226	2227	2228	2229	2230	2231	2232	2233	2234	2235	2236	2237	2238	2239
8C	2240	2241	2242	2243	2244	2245	2246	2247	2248	2249	2250	2251	2252	2253	2254	2255
8D	2256	2257	2258	2259	2260	2261	2262	2263	2264	2265	2266	2267	2268	2269	2270	2271
8E	2272	2273	2274	2275	2276	2277	2278	2279	2280	2281	2282	2283	2284	2285	2286	2287
8F	2288	2289	2290	2291	2292	2293	2294	2295	2296	2297	2298	2299	2300	2301	2302	2303
90	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319
91	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335
92	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351
93	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367
94	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383
95	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
96	2400	2401	2402	2403	2404	2405	2406	2407	2408	2409	2410	2411	2412	2413	2414	2415
97	2416	2417	2418	2419	2420	2421	2422	2423	2424	2425	2426	2427	2428	2429	2430	2431
98	2432	2433	2434	2435	2436	2437	2438	2439	2440	2441	2442	2443	2444	2445	2446	2447
99	2448	2449	2450	2451	2452	2453	2454	2455	2456	2457	2458	2459	2460	2461	2462	2463
9A	2464	2465	2466	2467	2468	2469	2470	2471	2472	2473	2474	2475	2476	2477	2478	2479
9B	2480	2481	2482	2483	2484	2485	2486	2487	2488	2489	2490	2491	2492	2493	2494	2495
9C	2496	2497	2498	2499	2500	2501	2502	2503	2504	2505	2506	2507	2508	2509	2510	2511
9D	2512	2513	2514	2515	2516	2517	2518	2519	2520	2521	2522	2523	2524	2525	2526	2527
9E	2528	2529	2530	2531	2532	2533	2534	2535	2536	2537	2538	2539	2540	2541	2542	2543
9F	2544	2545	2546	2547	2548	2549	2550	2551	2552	2553	2554	2555	2556	2557	2558	2559

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
A0	2560	2561	2562	2563	2564	2565	2566	2567	2568	2569	2570	2571	2572	2573	2574	2575
A1	2576	2577	2578	2579	2580	2581	2582	2583	2584	2585	2586	2587	2588	2589	2590	2591
A2	2592	2593	2594	2595	2596	2597	2598	2599	2600	2601	2602	2603	2604	2605	2606	2607
A3	2608	2609	2610	2611	2612	2613	2614	2615	2616	2617	2618	2619	2620	2621	2622	2623
A4	2624	2625	2626	2627	2628	2629	2630	2631	2632	2633	2634	2635	2636	2637	2638	2639
A5	2640	2641	2642	2643	2644	2645	2646	2647	2648	2649	2650	2651	2652	2653	2654	2655
A6	2656	2657	2658	2659	2660	2661	2662	2663	2664	2665	2666	2667	2668	2669	2670	2671
A7	2672	2673	2674	2675	2676	2677	2678	2679	2680	2681	2682	2683	2684	2685	2686	2687
A8	2688	2689	2690	2691	2692	2693	2694	2695	2696	2697	2698	2699	2700	2701	2702	2703
A9	2704	2705	2706	2707	2708	2709	2710	2711	2712	2713	2714	2715	2716	2717	2718	2719
AA	2720	2721	2722	2723	2724	2725	2726	2727	2728	2729	2730	2731	2732	2733	2734	2735
AB	2736	2737	2738	2739	2740	2741	2742	2743	2744	2745	2746	2747	2748	2749	2750	2751
AC	2752	2753	2754	2755	2756	2757	2758	2759	2760	2761	2762	2763	2764	2765	2766	2767
AD	2768	2769	2770	2771	2772	2773	2774	2775	2776	2777	2778	2779	2780	2781	2782	2783
AE	2784	2785	2786	2787	2788	2789	2790	2791	2792	2793	2794	2795	2796	2797	2798	2799
AF	2800	2801	2802	2803	2804	2805	2806	2807	2808	2809	2810	2811	2812	2813	2814	2815
B0	2816	2817	2818	2819	2820	2821	2822	2823	2824	2825	2826	2827	2828	2829	2830	2831
B1	2832	2833	2834	2835	2836	2837	2838	2839	2840	2841	2842	2843	2844	2845	2846	2847
B2	2848	2849	2850	2851	2852	2853	2854	2855	2856	2857	2858	2859	2860	2861	2862	2863
B3	2864	2865	2866	2867	2868	2869	2870	2871	2872	2873	2874	2875	2876	2877	2878	2879
B4	2880	2881	2882	2883	2884	2885	2886	2887	2888	2889	2890	2891	2892	2893	2894	2895
B5	2896	2897	2898	2899	2900	2901	2902	2903	2904	2905	2906	2907	2908	2909	2910	2911
B6	2912	2913	2914	2915	2916	2917	2918	2919	2920	2921	2922	2923	2924	2925	2926	2927
B7	2928	2929	2930	2931	2932	2933	2934	2935	2936	2937	2938	2939	2940	2941	2942	2943
B8	2944	2945	2946	2947	2948	2949	2950	2951	2952	2953	2954	2955	2956	2957	2958	2959
B9	2960	2961	2962	2963	2964	2965	2966	2967	2968	2969	2970	2971	2972	2973	2974	2975
BA	2976	2977	2978	2979	2980	2981	2982	2983	2984	2985	2986	2987	2988	2989	2990	2991
BB	2992	2993	2994	2995	2996	2997	2998	2999	3000	3001	3002	3003	3004	3005	3006	3007
BC	3008	3009	3010	3011	3012	3013	3014	3015	3016	3017	3018	3019	3020	3021	3022	3023
BD	3024	3025	3026	3027	3028	3029	3030	3031	3032	3033	3034	3035	3036	3037	3038	3039
BE	3040	3041	3042	3043	3044	3045	3046	3047	3048	3049	3050	3051	3052	3053	3054	3055
BF	3056	3057	3058	3059	3060	3061	3062	3063	3064	3065	3066	3067	3068	3069	3070	3071
C0	3072	3073	3074	3075	3076	3077	3078	3079	3080	3081	3082	3083	3084	3085	3086	3087
C1	3088	3089	3090	3091	3092	3093	3094	3095	3096	3097	3098	3099	3100	3101	3102	3103
C2	3104	3105	3106	3107	3108	3109	3110	3111	3112	3113	3114	3115	3116	3117	3118	3119
C3	3120	3121	3122	3123	3124	3125	3126	3127	3128	3129	3130	3131	3132	3133	3134	3135
C4	3136	3137	3138	3139	3140	3141	3142	3143	3144	3145	3146	3147	3148	3149	3150	3151
C5	3152	3153	3154	3155	3156	3157	3158	3159	3160	3161	3162	3163	3164	3165	3166	3167
C6	3168	3169	3170	3171	3172	3173	3174	3175	3176	3177	3178	3179	3180	3181	3182	3183
C7	3184	3185	3186	3187	3188	3189	3190	3191	3192	3193	3194	3195	3196	3197	3198	3199
C8	3200	3201	3202	3203	3204	3205	3206	3207	3208	3209	3210	3211	3212	3213	3214	3215
C9	3216	3217	3218	3219	3220	3221	3222	3223	3224	3225	3226	3227	3228	3229	3230	3231
CA	3232	3233	3234	3235	3236	3237	3238	3239	3240	3241	3242	3243	3244	3245	3246	3247
CB	3248	3249	3250	3251	3252	3253	3254	3255	3256	3257	3258	3259	3260	3261	3262	3263
CC	3264	3265	3266	3267	3268	3269	3270	3271	3272	3273	3274	3275	3276	3277	3278	3279
CD	3280	3281	3282	3283	3284	3285	3286	3287	3288	3289	3290	3291	3292	3293	3294	3295
CE	3296	3297	3298	3299	3300	3301	3302	3303	3304	3305	3306	3307	3308	3309	3310	3311
CF	3312	3313	3314	3315	3316	3317	3318	3319	3320	3321	3322	3323	3324	3325	3326	3327

HEXADECIMAL-DECIMAL INTEGER CONVERSION (Continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
D0	3328	3329	3330	3331	3332	3333	3334	3335	3336	3337	3338	3339	3340	3341	3342	3343
D1	3344	3345	3346	3347	3348	3349	3350	3351	3352	3353	3354	3355	3356	3357	3358	3359
D2	3360	3361	3362	3363	3364	3365	3366	3367	3368	3369	3370	3371	3372	3373	3374	3375
D3	3376	3377	3378	3379	3380	3381	3382	3383	3384	3385	3386	3387	3388	3389	3390	3391
D4	3392	3393	3394	3395	3396	3397	3398	3399	3400	3401	3402	3403	3404	3405	3406	3407
D5	3408	3409	3410	3411	3412	3413	3414	3415	3416	3417	3418	3419	3420	3421	3422	3423
D6	3424	3425	3426	3427	3428	3429	3430	3431	3432	3433	3434	3435	3436	3437	3438	3439
D7	3440	3441	3442	3443	3444	3445	3446	3447	3448	3449	3450	3451	3452	3453	3454	3455
D8	3456	3457	3458	3459	3460	3461	3462	3463	3464	3465	3466	3467	3468	3469	3470	3471
D9	3472	3473	3474	3475	3476	3477	3478	3479	3480	3481	3482	3483	3484	3485	3486	3487
DA	3488	3489	3490	3491	3492	3493	3494	3495	3496	3497	3498	3499	3500	3501	3502	3503
DB	3504	3505	3506	3507	3508	3509	3510	3511	3512	3513	3514	3515	3516	3517	3518	3519
DC	3520	3521	3522	3523	3524	3525	3526	3527	3528	3529	3530	3531	3532	3533	3534	3535
DD	3536	3537	3538	3539	3540	3541	3542	3543	3544	3545	3546	3547	3548	3549	3550	3551
DE	3552	3553	3554	3555	3556	3557	3558	3559	3560	3561	3562	3563	3564	3565	3566	3567
DF	3568	3569	3570	3571	3572	3573	3574	3575	3576	3577	3578	3579	3580	3581	3582	3583
E0	3584	3585	3586	3587	3588	3589	3590	3591	3592	3593	3594	3595	3596	3597	3598	3599
E1	3600	3601	3602	3603	3604	3605	3606	3607	3608	3609	3610	3611	3612	3613	3614	3615
E2	3616	3617	3618	3619	3620	3621	3622	3623	3624	3625	3626	3627	3628	3629	3630	3631
E3	3632	3633	3634	3635	3636	3637	3638	3639	3640	3641	3642	3643	3644	3645	3646	3647
E4	3648	3649	3650	3651	3652	3653	3654	3655	3656	3657	3658	3659	3660	3661	3662	3663
E5	3664	3665	3666	3667	3668	3669	3670	3671	3672	3673	3674	3675	3676	3677	3678	3679
E6	3680	3681	3682	3683	3684	3685	3686	3687	3688	3689	3690	3691	3692	3693	3694	3695
E7	3696	3697	3698	3699	3700	3701	3702	3703	3704	3705	3706	3707	3708	3709	3710	3711
E8	3712	3713	3714	3715	3716	3717	3718	3719	3720	3721	3722	3723	3724	3725	3726	3727
E9	3728	3729	3730	3731	3732	3733	3734	3735	3736	3737	3738	3739	3740	3741	3742	3743
EA	3744	3745	3746	3747	3748	3749	3750	3751	3752	3753	3754	3755	3756	3757	3758	3759
EB	3760	3761	3762	3763	3764	3765	3766	3767	3768	3769	3770	3771	3772	3773	3774	3775
EC	3776	3777	3778	3779	3780	3781	3782	3783	3784	3785	3786	3787	3788	3789	3790	3791
ED	3792	3793	3794	3795	3796	3797	3798	3799	3800	3801	3802	3803	3804	3805	3806	3807
EE	3808	3809	3810	3811	3812	3813	3814	3815	3816	3817	3818	3819	3820	3821	3822	3823
EF	3824	3825	3826	3827	3828	3829	3830	3831	3832	3833	3834	3835	3836	3837	3838	3839
F0	3840	3841	3842	3843	3844	3845	3846	3847	3848	3849	3850	3851	3852	3853	3854	3855
F1	3856	3857	3858	3859	3860	3861	3862	3863	3864	3865	3866	3867	3868	3869	3870	3871
F2	3872	3873	3874	3875	3876	3877	3878	3879	3880	3881	3882	3883	3884	3885	3886	3887
F3	3888	3889	3890	3891	3892	3893	3894	3895	3896	3897	3898	3899	3900	3901	3902	3903
F4	3904	3905	3906	3907	3908	3909	3910	3911	3912	3913	3914	3915	3916	3917	3918	3919
F5	3920	3921	3922	3923	3924	3925	3926	3927	3928	3929	3930	3931	3932	3933	3934	3935
F6	3936	3937	3938	3939	3940	3941	3942	3943	3944	3945	3946	3947	3948	3949	3950	3951
F7	3952	3953	3954	3955	3956	3957	3958	3959	3960	3961	3962	3963	3964	3965	3966	3967
F8	3968	3969	3970	3971	3972	3973	3974	3975	3976	3977	3978	3979	3980	3981	3982	3983
F9	3984	3985	3986	3987	3988	3989	3990	3991	3992	3993	3994	3995	3996	3997	3998	3999
FA	4000	4001	4002	4003	4004	4005	4006	4007	4008	4009	4010	4011	4012	4013	4014	4015
FB	4016	4017	4018	4019	4020	4021	4022	4023	4024	4025	4026	4027	4028	4029	4030	4031
FC	4032	4033	4034	4035	4036	4037	4038	4039	4040	4041	4042	4043	4044	4045	4046	4047
FD	4048	4049	4050	4051	4052	4053	4054	4055	4056	4057	4058	4059	4060	4061	4062	4063
FE	4064	4065	4066	4067	4068	4069	4070	4071	4072	4073	4074	4075	4076	4077	4078	4079
FF	4080	4081	4082	4083	4084	4085	4086	4087	4088	4089	4090	4091	4092	4093	4094	4095

POWERS OF TWO

2^n	n	2^{-n}
1	0	1.0
2	1	0.5
4	2	0.25
8	3	0.125
16	4	0.0625
32	5	0.03125
64	6	0.015625
128	7	0.0078125
256	8	0.00390625
512	9	0.001953125
1024	10	0.0009765625
2048	11	0.00048828125
4096	12	0.000244140625
8192	13	0.0001220703125
16384	14	0.00006103515625
32768	15	0.000030517578125
65536	16	0.0000152587890625
131072	17	0.00000762939453125
262144	18	0.000003814697265625
524288	19	0.0000019073486328125
1048576	20	0.00000095367431640625
2097152	21	0.000000476837158203125
4194304	22	0.0000002384185791015625
8388608	23	0.00000011920928955078125
16777216	24	0.000000059604644775390625
33554432	25	0.0000000298023273876953125
67108864	26	0.00000001490116119384765625
134217728	27	0.000000007450580396923828125
268435456	28	0.0000000037252902984619140625
536870912	29	0.00000000186264514923095703125
1073741824	30	0.000000000931322574615478515625
2147483648	31	0.0000000004656612873077392578125
4294967296	32	0.00000000023283064365386962890625
8589934592	33	0.00000000011641532182693481453125
17179869184	34	0.0000000000582076619134674072265625
34359738368	35	0.00000000002910383045673370361328125
68719476736	36	0.000000000014551915228366851806640625
137438953472	37	0.0000000000072759576141834259033203125
274877906944	38	0.00000000000363797880709171295166015625
549755813888	39	0.00000000000181898940354585647583078125
1099511627776	40	0.0000000000009094947017729282379150390625
2199023255552	41	0.00000000000045474735088646411895751953125
4398046511104	42	0.000000000000227373675443232059478759765625
8796093022208	43	0.000000000000113686837721616029739379882125
17592186044416	44	0.000000000000056843188860801486968994140625
35184372088832	45	0.000000000000028421709430404007434844970703125
70368744177664	46	0.000000000000014210854715207063717422853515625
140737488355328	47	0.00000000000000710542735760100185871124267578125
281474976710656	48	0.000000000000003552713678800500929355621337890625
562949953421312	49	0.0000000000000017763568394002504646778106689453125
1125899906842624	50	0.00000000000000088817841970012523233890533447265625
2251799813685248	51	0.00000000000000044408920985062616169452667236328125
4503599627370496	52	0.000000000000000222044604925031308084726336181640625
9007199254740992	53	0.00000000000000011102230246251565404236316680908203125
18014398509481984	54	0.00000000000000005551115123125782702181583404541015625
36028797018963968	55	0.0000000000000000277557556136289135105907917022705078125
72057594032727936	56	0.000000000000000013877878078144567552953958513525390625
144115188075855872	57	0.00000000000000000693899390390722837647697925367626953125
288230376151711744	58	0.0000000000000000034694469519536141888238489627838134765625
576460752303423488	59	0.00000000000000000173472347597680709441192448139190673828125
1152921504606846976	60	0.00000000000000000086736173798840354720596224069595369140625
2305843009213693952	61	0.0000000000000000004336808689942017736029811203679766845703125
4611686018427387904	62	0.00000000000000000021684043449710088680149056017398834228515625
922337236854775808	63	0.000000000000000000108420217248550443400745280086994171142578125

MATHEMATICAL CONSTANTS

Constant	Decimal Value	Hexadecimal Value
e	3.14159 26535 89793	3.243F 6A89
e^{-1}	0.31830 98861 83790	0.517C C187
\sqrt{e}	1.77245 38509 05516	1.C58F 891C
$\ln e$	1.44272 98858 49400	1.250D 048F
e	2.71828 18284 59045	2.87E1 5163
e^{-1}	0.36787 94411 71442	0.5E2D 58D9
\sqrt{e}	1.64872 12707 00128	1.A612 98E2
$\log_{10} e$	0.43429 44819 03252	0.6F2D EC55
$\log_2 e$	1.44269 50408 88963	1.7154 7653
γ	0.57721 56649 01533	0.93C4 67E4
$\ln \gamma$	-0.54953 93129 81645	-0.8CAE 98C1
$\sqrt{2}$	1.41421 35623 73095	1.6A09 E668
$\ln 2$	0.69314 71805 59945	0.8172 17F8
$\log_{10} 2$	0.30102 99956 63981	0.4D10 4D42
$\sqrt{10}$	3.16227 76601 68379	3.298B 075C
$\ln 10$	2.30258 40929 94046	2.4D75 3777

POWERS OF SIXTEEN

16^n	n	16^{-n}
1	0	0.10000 00000 00000 00000 x 10^{-1}
16	1	0.62500 00000 00000 00000 x 10^{-2}
256	2	0.39062 50000 00000 00000 x 10^{-3}
4 096	3	0.24414 06250 00000 00000 x 10^{-4}
65 536	4	0.15258 78906 25000 00000 x 10^{-5}
1 048 576	5	0.95367 43164 06250 00000 x 10^{-6}
16 777 216	6	0.59604 64477 53906 25000 x 10^{-7}
268 435 456	7	0.37252 90298 45191 40525 x 10^{-8}
4 294 967 296	8	0.23283 06436 53869 62891 x 10^{-9}
68 719 476 736	9	0.14551 91522 83668 51807 x 10^{-10}
1 099 511 627 776	10	0.90949 47017 72928 23792 x 10^{-11}
17 592 186 044 416	11	0.56843 41886 08080 14670 x 10^{-12}
281 474 976 710 656	12	0.35527 13678 90950 09294 x 10^{-13}
4 503 599 627 370 496	13	0.22204 46049 25031 30808 x 10^{-14}
72 057 594 037 927 936	14	0.13877 78780 78144 56755 x 10^{-15}
1 152 921 504 606 846 976	15	0.86736 17379 88403 54721 x 10^{-16}

POWERS OF TEN (Converted to Hexadecimal Values)

10^n	n	10^{-n}
1	0	1.0000 0000 0000 0000
A	1	0.1999 9999 9999 999A
64	2	0.28F5 C28F 5C28 F5C3 x 16^{-1}
3E8	3	0.4189 374B C6A7 EF9E x 16^{-2}
2710	4	0.68DB 8BAC 710C B296 x 16^{-3}
1 86A0	5	0.A7C5 AC47 1B47 8423 x 16^{-4}
F 4240	6	0.10C6 F7A0 B5ED 8D37 x 16^{-5}
98 9680	7	0.1AD7 F29A 8CAF 4858 x 16^{-6}
5F5 E100	8	0.2AF3 1DC4 6118 73BF x 16^{-7}
3B9A CA00	9	0.4488 2FA0 9B5A 52CC x 16^{-8}
2 5408 E400	10	0.6DF3 7F67 5EF6 EADF x 16^{-9}
17 4876 E800	11	0.AFEB FF0B CB24 AAFF x 16^{-10}
E8 D4A5 1000	12	0.1197 9981 2DEA 1119 x 16^{-11}
918 4E72 A000	13	0.1C25 C268 4976 81C2 x 16^{-12}
5AF3 107A 4000	14	0.2D09 370D 4257 3604 x 16^{-13}
3 8D7E A4C6 8000	15	0.480E 8E7B 9D58 566D x 16^{-14}
23 8652 6FC1 0000	16	0.734A CA5F 6226 FOAE x 16^{-15}
163 4578 5D8A 0000	17	0.8877 AA32 36A4 B449 x 16^{-16}
DE0 86B3 A764 0000	18	0.1272 5DD1 D243 ABA1 x 16^{-17}
8AC7 2304 89E8 0000	19	0.1D83 C94F 86D2 AC35 x 16^{-18}