

* NAME BASIC-INIT-A

DOC. 70181909000

REV. A

PAGE

1

0001
0002
0003
0004
0005
0006
0007
0008
0009
0010
0011
0012
0013
0014
0015
0016
0017
0018
0019
0020
0021
0022
0023
0024
0025
0026
0027
0028
0029

* NAME BASIC-INIT-A

DOC. 70181909000

REV. A

BSI00010

*

BSI00020

*

BSI00030

* DESCRIPTION INITIALIZATION ROUTINE FOR STAND-ALONE VERSION OF BASIC-16

BSI00040

*

BSI00050

*

BSI00060

* REVISION HISTORY

BSI00070

*

REV.

DATE

ECO NO.

BSI00080

*

A

RELEASED

BSI00090

*

BSI00100

*

BSI00110

*

BSI00120

*

BSI00130

*

BSI00140

* DOCUMENTATION REFERENCES

BSI00150

*

TITLE

DOC. NO.

BSI00160

*

BASIC-16 USER'S MANUAL

70130072543

BSI00170

*

BSI00180

*

BSI00190

*

BSI00200

*

BSI00210

*

BSI00220

* COPYRIGHT 1971 BY HONEYWELL INFORMATION SYSTEMS INC., COMPUTER

BSI00230

* SYSTEMS DIVISION, FRAMINGHAM, MASSACHUSETTS. CONTENTS OF THIS

BSI00240

* PUBLICATION MAY NOT BE REPRODUCED IN ANY FORM, IN WHOLE OR IN

BSI00250

* PART, WITHOUT PERMISSION OF THE COPYRIGHT OWNER. ALL RIGHTS

BSI00260

* RESERVED.

BSI00270

*****BSI00280

EJCT

BSI00290

* NAME BASIC-IN11-A

DOC. 70181909000

REV. A

PAGE 2

0030		EXT	SII	ADDRESS OF HIGHEST AVAILABLE MEMORY	BSI00300
0031	*			LOCATION	BSI00310
0032		EXT	PIB	ADDRESS OF LOWEST WORD IN USER SPACE	BSI00320
0033		EXT	TYPE	OUTPUTS MESSAGE ON ASR	BSI00330
0034		EXT	IPUT	INPUTS A LINE FROM THE ASR	BSI00340
0035		EXT	GNBC	GETS NEXT NON-BLANK CHARACTER FROM THE	BSI00350
0036	*			INPUT BUFFER	BSI00360
0037		EXT	LODF	FLAGS PAPER TAPE INPUT FROM ASR	BSI00370
0038		EXT	LFCR	OUTPUTS CARRIAGE RETURN, LINE FEED ON ASR	BSI00380
0039		EXT	JOB	JOB COMMAND PROCESSOR ENTRY	BSI00390
0040		EXT	FINT	INTEGER TO FLOATING POINT CONVERSION	BSI00400
0041		EXT	DELT	DAC TO ROUTINE TO FLAG ERROR DF	BSI00410
0042		EXT	ATND	DAC TO ARCTANGENT FUNCTION	BSI00420
0043		EXT	TAND	DAC TO TANGENT FUNCTION	BSI00430
0044		EXT	SIND	DAC TO SINE FUNCTION	BSI00440
0045		EXT	COSD	DAC TO COSINE FUNCTION	BSI00450
0046		EXT	SQRD	DAC TO SQUARE ROOT FUNCTION	BSI00460
0047		EXT	SCVL	STORES A NUMBER INTO THE FLOATING POINT	BSI00470
0048	*			ACCUMULATOR	BSI00480
0049		EXT	PCVL	PRINTS THE NUMBER IN THE FLOATING POINT	BSI00490
0050	*			ACCUMULATOR	BSI00500
0051		EXT	C1		BSI00510
0052		EXT	C10		BSI00520
0053		EXT	C215		BSI00530
0054		EXT	C240		BSI00540
0055		EXT	C241		BSI00550
0056		EXT	C260		BSI00560
0057	*				BSI00570
0058	*				BSI00580
0059	*				BSI00590
0060		ENT	INIT		BSI00600
0061		ENT	CSRH	HIGH CROSS SECTOR REFERENCE GENERATOR	BSI00610
0062	*				BSI00620
0063	*				BSI00630
0064	*				BSI00640
0065		REL			BSI00650
0066	*				BSI00660



HONEYWELL INFORMATION SYSTEMS LTD

PROGRAM DOCUMENTATION

* NAME BASIC-IN11-A

DOC. 70181909000

REV. A

PAGE

3

0067
0068
0069

*
*

EJCT

BSI00670
BSI00680
BSI00690

0070
0071
0072
0073
0074
0075
0076
0077
0078
0079
0080
0081
0082
0083
0084
0085
0086
0087
0088
0089
0090
0091
0092
0093
0094
0095
0096
0097
0098
0099
0100
0101
0102
0103
0104
0105
0106

INITIALIZATION ROUTINE

CALLING SEQUENCE:

JST INII
RETURN IS MADE THROUGH JOB COMMAND PROCESSOR

THE PROGRAM TEXT TABLE BASE POINTER (PTB) IS INITIALIZED TO THE LOWEST ADDRESS OF THE INITIALIZATION ROUTINE, AND AN IDENTIFICATION MESSAGE IS PRINTED. IF THE IOS WAS LOADED AFTER MTHPAK, OR IF MIHPAK WAS LOADED AFTER THE INITIALIZATION ROUTINE, AN ILLEGAL CONFIGURATION MESSAGE IS PRINTED, AND THE PROGRAM HALTS. IF ANYTHING WAS LOADED BETWEEN MTHPAK AND THE INITIALIZATION ROUTINE, THERE IS A JUMP TO SET THE STATEMENT INDEX HIGH POINTER (SIT). OTHERWISE THE USER IS ASKED IF HE WOULD LIKE TO DELETE ATN. A NO ANSWER CAUSES A JUMP TO SET SIT. A YES ANSWER CAUSES PTB TO BE SET TO THE FIRST WORD OF THE ROUTINE, AND THE POINTER TO THE ROUTINE IS REPLACED BY A POINTER TO A ROUTINE WHICH FLAGS A DELETED LIBRARY FUNCTION (DF) ERROR. THIS SEQUENCE IS THEN REPEATED FOR SIN, COS, AND TAN, AND SQR.

S11 IS INITIALIZED TO THE HIGH CORE MEMORY ADDRESS, AND THE USER IS ASKED IF IT IS ALL RIGHT TO USE ALL OF CORE. A YES ANSWER CAUSES A JUMP TO CALCULATE THE AMOUNT OF USER SPACE. OTHERWISE THE USER INPUT HIGH OCTAL ADDRESS IS FORMED. NON-OCTAL NUMBERS, OVERFLOW, OR AN ADDRESS GREATER THAN THE HIGH CORE ADDRESS CAUSES A MESSAGE TO BE PRINTED REQUESTING A YES OR A HIGH OCTAL ADDRESS. OTHERWISE S11 IS SET TO THE USER INPUT HIGH OCTAL ADDRESS. THE AMOUNT OF USER SPACE IS CALCULATED BY SUBTRACTING PTB FROM S11 AND ADDING ONE. IF THE RESULT IS LESS THAN ELEVEN, A MESSAGE IS PRINTED TO FLAG INSUFFICIENT USER SPACE, AND THE ROUTINE LOOPS TO RESET S11. OTHERWISE THE AMOUNT OF USER SPACE IS PRINTED, THE CALL TO THE INITIALIZATION ROUTINE IS REPLACED WITH A CRA INSTRUCTION, AND THE ROUTINE EXITS THROUGH THE JOB COMMAND PROCESSOR.

BS100700
BS100710
BS100720
BS100730
BS100740
BS100750
BS100760
BS100770
BS100780
BS100790
BS100800
BS100810
BS100820
BS100830
•BS100840
BS100850
BS100860
•BS100870
BS100880
BS100890
BS100900
BS100910
BS100920
BS100930
BS100940
EBS100950
BS100960
BS100970
BS100980
BS100990
BS101000
BS101010
BS101020
BS101030
BS101040
BS101050
BS101060

* NAME BASIC-INIT-A

DOC. 70181909000

REV. A

PAGE

5

0107			*						BSI01070
0108	00000	0 000000	INTA	DAC	*			ADDRESS OF FIRST WORD OF INITIALIZATION	BSI01080
0109			*					ROUTINE	BSI01090
0110	00001	0 000000	INIT	DAC	**			ENTRY	BSI01100
0111	00002	0 02 00000	LDA	INIA				INITIALIZE LOW POINTER TO FIRST WORD	BSI01110
0112	00003	0 04 00000	STA	PTB				OF INITIALIZATION ROUTINE	BSI01120
0113	00004	0 04 00000	STA	LODF				SET FLAG FOR NO PAPER TAPE INPUT	BSI01130
0114	00005	0 10 00000	JST	LFCR				C/R, LINE FEED	BSI01140
0115	00006	0 10 00000	JST	IYPE				OUTPUT ID MESSAGE	BSI01150
0116	00007	0 000000	XAC	IDMS				X	BSI01160
0117	00010	0 10 00000	JST	LFCR				A COUPLE OF LINE FEEDS	BSI01170
0118	00011	0 10 00000	JST	LFCR				X	BSI01180
0119			*						BSI01190
0120			*					CHECK FOR ILLEGAL CONFIGURATION	BSI01200
0121			*						BSI01210
0122	00012	0 02 00205	LDA	HMPK				LOAD HIGH ADDRESS OF MTHPAK	BSI01220
0123	00013	0 07 00206	SUB	IOSA				SUBTRACT ADDRESS IN IOS	BSI01230
0124	00014	100400	SPL					SKIP IF POSITIVE	BSI01240
0125	00015	0 01 00156	JMP	IN06				JUMP TO FLAG ILLEGAL CONFIGURATION IF	BSI01250
0126			*					NEGATIVE	BSI01260
0127	00016	0 02 00205	LDA	HMPK				LOAD THE LAST ADDRESS OF THE MTHPAK	BSI01270
0128	00017	101100	SLN					SKIP IF THE ADDRESS IS ODD	BSI01280
0129	00020	141206	AOA					IF IT'S EVEN ADD ONE SINCE LOADING STARTS	BSI01290
0130			*					ON AN EVEN LOCATION	BSI01300
0131	00021	141206	AOA					ADD ONE AND	BSI01310
0132	00022	0 11 00000	CAS	INTA				COMPARE WITH THE ADDRESS OF THE FIRST WORD	BSI01320
0133			*					OF THE INITIALIZATION ROUTINE	BSI01330
0134	00023	0 01 00156	JMP	IN06				IF THE MTHPAK WAS LOADED AFTER THE INITIAL-	BSI01340
0135			*					IALIZATION ROUTINE THEN JUMP TO FLAG AN	BSI01350
0136			*					ILLEGAL CONFIGURATION	BSI01360
0137	00024	100000	SKP					IF MTHPAK WAS LOADED JUST BEFORE THE	BSI01370
0138			*					INITIALIZATION ROUTINE THEN JUMP TO	BSI01380
0139			*					ASK USER IF HE WISHES TO DELETE ANY OF	BSI01390
0140			*					LIBRARY FUNCTION ROUTINES	BSI01400
0141	00025	0 01 00047	JMP	IN01				IF ANYTHING WAS LOADED BETWEEN THE MTHPAK	BSI01410
0142			*					AND THE INITIALIZATION ROUTINE THEN	BSI01420
0143			*					JUMP TO SET THE HIGH POINTER (SIT)	BSI01430

0144			*				BSI01440
0145			*	SET LOW POINTER, PTB			BSI01450
0146			*				BSI01460
0147	00026	0 02 00214	IN02	LDA	ATQD	LOAD ADDRESS OF ATN MESSAGE	BSI01470
0148	00027	0 10 00161		JST	DFUA	DELETE ARCTANGENT FUNCTION	BSI01480
0149	00030	0 02 00000		LDA	DELT	YES-REPLACE POINTER TO AIN WITH A	BSI01490
0150	00031	0 13 00000		IMA	ATND	POINTER TO ROUTINE TO FLAG ERROR DF	BSI01500
0151	00032	0 04 00000		STA	PTB	AND SET LOW POINTER TO FIRST WORD OF ATN	BSI01510
0152	00033	0 02 00216		LDA	SCQD	LOAD ADDRESS OF SIN, COS, TAN MESSAGE	BSI01520
0153	00034	0 10 00161		JST	DFUA	DELETE SIN, COS, TAN FUNCTIONS	BSI01530
0154	00035	0 02 00000		LDA	DELT	YES- REPLACE POINTER TO THOSE FUNCTIONS	BSI01540
0155	00036	0 04 00000		STA	IAND	WITH POINTER TO FLAG ERROR DF	BSI01550
0156	00037	0 04 00000		STA	SIND	X	BSI01560
0157	00040	0 13 00000		IMA	COSD	X	BSI01570
0158	00041	0 04 00000		STA	PTB	SET LOW POINTER TO FIRST WORD OF COS	BSI01580
0159	00042	0 02 00215		LDA	SQQD	LOAD ADDRESS OF SQUARE ROOT MESSAGE	BSI01590
0160	00043	0 10 00161		JST	DFUA	DELETE SQUARE ROOT FUNCTION	BSI01600
0161	00044	0 02 00000		LDA	DELT	YES-REPLACE POINTER TO SQUARE ROOT FUNCTION	BSI01610
0162	00045	0 13 00000		IMA	SQRD	WITH POINTER TO ROUTINE TO FLAG ERROR DF	BSI01620
0163	00046	0 04 00000		STA	PTB	AND SET LOW POINTER TO FIRST WORD OF SQR	BSI01630
0164			*				BSI01640
0165			*	NOW SET HIGH POINTER, SIT			BSI01650
0166			*				BSI01660
0167	00047	0 02 00211	IN01	LDA	C16K	LOAD HIGHEST POSSIBLE ADDRESS (16K)	BSI01670
0168	00050	0 04 00000	IN10	STA	SIT	STORE ADDRESS IN HIGH POINTER	BSI01680
0169	00051	-0 13 00000		IMA*	SIT	CHECK IF ADDRESS EXISTS BY STORING,	BSI01690
0170	00052	-0 13 00000		IMA*	SIT	THEN LOADING (BUT DON'T DESTROY CONTENTS OF	BSI01700
0171	00053	0 05 00000		ERA	SIT	ADDRESS), AND COMPARING	BSI01710
0172	00054	101040		SNZ		TEST IF THE SAME	BSI01720
0173	00055	0 01 00060		JMP	*+3	YES-HIGH MEMORY ADDRESS IS IN SIT	BSI01730
0174	00056	0 07 00210		SUB	C4K	NO-SUBTRACT 4K	BSI01740
0175	00057	0 01 00050		JMP	IN10	AND TRY AGAIN	BSI01750
0176	00060	0 10 00000		JST	IYPE	ASK USER IF HE WOULD LIKE TO USE ALL OF	BSI01760
0177	00061	0 000000		XAC	HMAM	CORE	BSI01770
0178	00062	0 10 00000	IN03	JST	IYPE	REQUEST A YES OR HIGH OCTAL ADDRESS	BSI01780
0179	00063	0 000000		XAC	AYOH	X	BSI01790
0180	00064	0 10 00000		JST	LFCH	C/R, LINE FEED	BSI01800

* NAME BASIC-INIT-A

DOC. 70181909000

REV. A

PAGE

7

0181	00065	0 02 00000	LDA	C241	OUTPUT EXCLAMATION POINT	BSI01810
0182	00066	0 10 00000	JSI	IPUT	THEN INPUT LINE FROM ASR	BSI01820
0183	00067	0 000000	XAC	SBUF	INPUT BUFFER	BSI01830
0184	00070	0 10 00000	JSI	GNBC	GET THE FIRST NON-BLANK CHARACTER	BSI01840
0185	00071	0 11 00213	CAS	C331	IS IT A 'Y'	BSI01850
0186	00072	0 01 00062	JMP	IN03	NOT Y OR NUMBER-REQUEST YES OR HIGH	BSI01860
0187			*		OCTAL ADDRESS	BSI01870
0188	00073	0 01 00127	JMP	IN04	YES-ASSUME YES ANSWER-JUMP TO CALCULATE	BSI01880
0189			*		USER SPACE	BSI01890
0190	00074	000201	IAB		SAVE THE CHARACTER	BSI01900
0191	00075	140040	CRA	CRA		BSI01910
0192	00076	0 04 00207	STA	H0A	INITIALIZE HIGH OCTAL ADDRESS TO ZERO	BSI01920
0193	00077	000201	IAB		AND CLEAR B REGISTER	BSI01930
0194			*			BSI01940
0195			*		HERE TO INPUT USER HIGH OCTAL ADDRESS	BSI01950
0196			*			BSI01960
0197	00100	0 07 00000	IN05 SUB	C260	SUBTRACT '260 TO GET PURE NUMBER	BSI01970
0198	00101	100400	SPL		MAKE SURE IT'S NOT LESS THAN ZERO	BSI01980
0199	00102	0 01 00062	JMP	IN03	IF LESS, REQUEST YES OR HIGH ADDRESS	BSI01990
0200	00103	0 11 00000	CAS	C10	OR GREATER THAN 7	BSI02000
0201	00104	101000	NOP		X	BSI02010
0202	00105	0 01 00062	JMP	IN03	IF GREATER, THEN REQUEST YES OR HIGH OCTAL	BSI02020
0203			*		ADDRESS	BSI02030
0204	00106	0 13 00207	IMA	H0A	MAKE ROOM FOR THE DIGIT	BSI02040
0205	00107	0412 75	LLR	3	BY SHIFTING ADDRESS THREE PLACES LEFT	BSI02050
0206	00110	0 05 00207	ERA	H0A	ADD THE DIGIT	BSI02060
0207	00111	000201	IAB		MAKE SURE THERE IS NO OVERFLOW BY TESTING	BSI02070
0208	00112	100040	SZE		THE B REGISTER	BSI02080
0209	00113	0 01 00062	JMP	IN03	OVERFLOW-REQUEST YES OR HIGH OCTAL ADDRESS	BSI02090
0210	00114	000201	IAB		REPOSITION	BSI02100
0211	00115	0 04 00207	STA	H0A	SAVE ADDRESS	BSI02110
0212	00116	0 10 00000	JSI	GNBC	GET NEXT NON-BLANK CHARACTER	BSI02120
0213	00117	0 11 00000	CAS	C215	CHECK FOR C/R	BSI02130
0214	00120	0 01 00100	JMP	IN05	NO-LOOP TO ADD DIGIT TO HIGH ADDRESS	BSI02140
0215	00121	0 02 00207	LDA	H0A	YES-LOAD USER HIGH INPUT ADDRESS	BSI02150
0216	00122	101400	SMI		SKIP IF USER INPUT HIGH ADDRESS IS NEGATIVE	BSI02160
0217	00123	0 11 00000	CAS	SIT	COMPARE WITH HIGH CORE ADDRESS	BSI02170

0218	00124	0 01 00062	JMP	IN03	USER INPUT HIGH ADDRESS IS TOO LARGE OR	BS102180
0219		*			NEGATIVE-JUMP TO INPUT AGAIN	BS102190
0220	00125	101000	NOP		X	BS102200
0221	00126	0 04 00000	STA	SIT	OK-SAVE AS HIGH POINTER	BS102210
0222		*				BS102220
0223		*				BS102230
0224		*			CALCULATE AND PRINT TOTAL USER SPACE	BS102240
0225	00127	0 02 00000	IN04 LDA	SIT	LOAD HIGH POINTER	BS102250
0226	00130	0 07 00000	SUB	PIB	SUBTRACT LOW POINTER	BS102260
0227	00131	0 06 00000	ADD	C1	ADD ONE	BS102270
0228	00132	0 11 00000	CAS	C10	MAKE SURE IT ISN'T TOO SMALL	BS102280
0229	00133	0 01 00141	JMP	**6	OK-JUMP TO PRINT USER SPACE MESSAGE	BS102290
0230	00134	101000	NOP		X	BS102300
0231	00135	0 10 00000	JSI	TYPE	TOO SMALL-FLAG INSUFFICIENT USER SPACE	BS102310
0232	00136	0 000000	XAC	ISSM	X	BS102320
0233	00137	0 10 00000	JSI	LFCR	C/R, LINE FEED	BS102330
0234	00140	0 01 00047	JMP	IN01	LOOP TO SET HIGH POINTER	BS102340
0235	00141	0 10 00000	JSI	FINT	FLOAT THE NUMBER	BS102350
0236	00142	0 10 00000	JSI	SCVL	STORE IT INTO FLOATING POINT ACCUMULATOR	BS102360
0237	00143	0 02 00000	LDA	C240	SURPRESS BLANKS	BS102370
0238	00144	0 10 00000	JSI	PCVL	PRINT THE NUMBER	BS102380
0239	00145	0 10 00000	JSI	TYPE	PRINT USER SPACE MESSAGE	BS102390
0240	00146	0 000000	XAC	USPM	X	BS102400
0241	00147	0 10 00000	JSI	LFCR	C/R, LINE FEED	BS102410
0242		*				BS102420
0243		*				BS102430
0244		*			CHANGE CALL TO THE INITIALIZATION ROUTINE TO A CRA	BS102440
0245	00150	0 02 00001	LDA	INIT	LOAD ADDRESS PLUS ONE OF CALL	BS102450
0246	00151	0 07 00000	SUB	C1	SUBTRACT ONE	BS102460
0247	00152	0 04 00001	STA	INIT	AND SAVE	BS102470
0248	00153	0 02 00075	LDA	CRA	LOAD CRA	BS102480
0249	00154	-0 04 00001	STA*	INIT	STORE IN LOCATION OF CALL	BS102490
0250	00155	0 01 00000	JMP	JOB	EXIT THROUGH JOB COMMAND PROCESSOR	BS102500
0251		*				BS102510
0252		*			HERE TO FLAG AN ILLEGAL CONFIGURATION	BS102520
0253		*				BS102530
0254	00156	0 10 00000	IN06 JSI	TYPE	PRINT MESSAGE TO RELOAD	BS102540



* NAME BASIC-IN11-A

DOC. 70181909000

REV. A

PAGE 9

0255 00157 0 000217
0256 00160 000000
0257
0258

DAC ILCM
HLI
*
EJCI

X
AND HALT

BSI02550
BSI02560
BSI02570
BSI02580

0259	*				BSI02590
0260	*				BSI02600
0261	*				BSI02610
0262	*				BSI02620
0263	*				BSI02630
0264	*				BSI02640
0265	*				BSI02650
0266	*				BSI02660
0267	*				BSI02670
0268	*				BSI02680
0269	*				BSI02690
0270	*				BSI02700
0271	*				BSI02710
0272	*				BSI02720
0273	*				BSI02730
0274	*				BSI02740
0275	*				BSI02750
0276	*				BSI02760
0277	*				BSI02770
0278	*				BSI02780
0279	00161	0 000000	DFUA DAC **	ENTRY	BSI02790
0280	00162	0 04 00166	STA DFID	STORE FUNCTION NAME MESSAGE	BSI02800
0281	00163	0 10 00000	JSI TYPE	OUTPUT QUESTION PREFIX	BSI02810
0282	00164	0 000000	XAC DFQ	X	BSI02820
0283	00165	0 10 00000	JSI TYPE	PRINT FUNCTION NAME	BSI02830
0284	00166	0 000000	DFID DAC **	X	BSI02840
0285	00167	0 10 00000	JSI LFCR	OUTPUT C/R, LINE FEED	BSI02850
0286	00170	0 02 00000	DF02 LDA C241	OUTPUT AN EXCAMATION POINT	BSI02860
0287	00171	0 10 00000	JSI IPUT	INPUT FROM THE ASR UNTIL A C/R	BSI02870
0288	00172	0 000000	XAC SBUF	INPUT BUFFER	BSI02880
0289	00173	0 10 00000	JSI GNBC	GET THE FIRST NON-BLANK CHARACTER	BSI02890
0290	00174	0 11 00213	CAS C331	IS IT A 'Y'	BSI02900
0291	00175	0 01 00202	JMP DF01	NO-REQUEST A YES OR A NO	BSI02910
0292	00176	-0 01 00161	JMP* DFUA	Y-ASSUME YES ANSWER-EXIT TO RESET LOW	BSI02920
0293			*	POINTER	BSI02930
0294	00177	0 11 00212	CAS C316	IS IT A 'N'	BSI02940
0295	00200	100000	SKP	NO-REQUEST A YES OR NO ANSWER	BSI02950

DELETE LIBRARY FUNCTION ROUTINE

CALLING SEQUENCE:

JSI DFUA ADDRESS OF MESSAGE COANTAINING FUNCTION
 NAME(S) IN THE A REGISTER
RETURN IF YES ANSWER

A MESSAGE IS PRINTED ASKING THE USER IF HE WOULD LIKE TO
 DELETE THE LIBRARY FUNCTION(S) WHOSE NAME(S) IS CONTAINED IN THE
 MESSAGE. A LINE IS INPUTED FROM THE ASR, AND THE FIRST NON-BLANK
 CHARACTER IS TESTED. IF IT IS A Y, RETURN IS MADE. IF IT IS A N,
 THE ROUTINE JUMPS INTO INIT TO SET IT. IF THE FIRST NON-BLANK
 CHARACTER IS OTHER THAN A Y OR A N, A MESSAGE IS PRINTED REQUESTING
 A YES OR A NO ANSWER.

* NAME BASIC-INIT-A

DOC. 70181909000

REV. A

PAGE 11

0296	00201	0 01 00047	JMP	IN01
0297	00202	0 10 00000 DF01	JSI	TYPE
0298	00203	0 000000	XAC	AYON
0299	00204	0 01 00170	JMP	DF02
0300			EJCT	

N-ASSUME NO ANSWER-JUMP TO SET HIGH POINTER	BSI02960
NO-REQUEST A YES OR NO ANSWER	BSI02970
X	BSI02980
LOOP TO INPUT ANSWER	BSI02990
	BSI03000

* NAME BASIC-IN11-A

DOC. 70181909000

REV. A

PAGE 12

0301			*						BSI03010
0302			*						BSI03020
0303			*						BSI03030
0304	00205	0 000000	HMPK XAC	PP12		ADDRESS OF LAST WORD OF MTKPAK			BSI03040
0305	00206	0 000000	IUSA XAC	IPUT		ADDRESS IN IOS PACKAGE			BSI03050
0306	00207	000000	HUA BSZ	1					BSI03060
0307	00210	010000	C4K OCI	10000					BSI03070
0308	00211	037777	C16K OCI	37777					BSI03080
0309	00212	000316	C316 OCI	316		ASCII N			BSI03090
0310	00213	000331	C331 OCI	331		ASCII Y			BSI03100
0311	00214	0 000000	AIQD XAC	ATNQ		ADDRESS OF ATN NAME MESSAGE			BSI03110
0312	00215	0 000000	SQQD XAC	SQRQ		ADDRESS OF SIN, COS, AND TAN NAME MESSAGE			BSI03120
0313	00216	0 000000	SCQD XAC	SCIQ		ADDRESS OF SQR NAME MESSAGE			BSI03130
0314	00217	144714	ILCM BCI	14,ILLEGAL CONFIGURATION-RELOAD					BSI03140
	00220	146305							
	00221	143701							
	00222	146240							
	00223	141717							
	00224	147306							
	00225	144707							
	00226	152722							
	00227	140724							
	00230	144717							
	00231	147255							
	00232	151305							
	00233	146317							
	00234	140704							
0315	00235	000000		OCI	0				BSI03150
0316		000236	CSRH EQU	*		USING LDR-APM, THIS WILL GENERATE THE LAST			BSI03160
0317			*			CROSS SECTOR REFERENCE IN SECTOR ZERO			BSI03170
0318			*						BSI03180
0319			*						BSI03190
0320			*						BSI03200
0321				END					BSI03210

ATND	000000E	AIQD	000214	C1	000000E	C10	000000E
C16K	000211	C215	000000E	C240	000000E	C241	000000E



C260	000000E	C316	000212	C331	000213	C4K	000210
COSD	000000E	CRA	000075	CSRH	000236	DELI	000000E
DF01	000202	DF02	000170	DFID	000166	DFUA	000161
FINT	000000E	GNBC	000000E	HMPK	000205	HUA	000207
ILCM	000217	IN01	000047	IN02	000026	IN03	000062
IN04	000127	IN05	000100	IN06	000156	IN10	000050
INIT	000001	INTA	000000	IOSA	000206	IPUT	000000E
JOB	000000E	LFCR	000000E	LODF	000000E	PCVL	000000E
PTB	000000E	SCQD	000216	SCVL	000000E	SIND	000000E
SIT	000000E	SQQD	000215	SQRD	000000E	TAND	000000E
TYPE	000000E						

0000 WARNING OR ERROR FLAGS

DAP-16 MOD 2 REV. D 06-28-71