

Spool Queue Line #: 28  
IRIS LU/Filename : 18/L.TIFD83.9291

Printed on/at : FEB 7, 1990 14:51:00  
For Group/User: 0, 1  
On Port No: 5

Print control parameters :  
Printer Class code : 0  
Form Code/paper type : ?  
Print Priority (0-7) : 5  
Starting Page Number : 1  
This is copy number : 1  
Keep file (Y/N) : Y  
Notify User when done: N  
Comments, optional : For RELSE CNTRL

```
.EOT ;"TIP00" (DISCSUBS GROUP 5) FOR IRIS R9. xx  
.EOT  
.EOT ;"DSUBDEFS" FOR IRIS  
.END
```

ASM 18/A. TIPDB3. 9291!, @18/L. TIPDB3. 9291!, B050, -B051, B052  
FEB 7, 1990 11:40:43

; Batchfile: R95JCL. TIPDB3

; A = 9291

; -R95DEFSPZ  
; -R95DSUBDEFSD  
; R92TIPDSB83SA

.EOT ; "TIPB3" (DISCSUBS GROUP 5) FOR IRIS R9. xx

<< BT = R92TIPDSB83SA; BD = 18/A.TIPD83.9291! >>  
DSUB83.R?

R9 TYPISY Discsub, written by JPMH, 27-Jly-82

This discsub provides the following functions:-

- 0) It returns its own revision number
- 1) It permits the insertion of one string into another
- 2) It provides the PEEK function
- 3) It provides the POKE function
- 4) It provides the SOFT function

The PEEK and POKE functions require that an extra parameter, the key, be provided. This parameter is checked against the sum of the PLC and the UVS thus ensuring that the program is unmodified.

To determine the value required as the key, word 57720 should be changed from SUBZ 1,1 to JMP .+1, thus causing an error to be set when the key is wrong. In such a case the correct key will be returned in the parameter A.

Calling sequence:-

CALL 83,MODE,<params list>

MODE=0, <params list>=returned revision number (numeric)  
 MODE=1, <params list>=string to insert into, insert string,  
           insertion position (numeric)  
 MODE=2, <params list>=address, value, key  
 MODE=3, <params list>=address, new value, key  
 MODE=4, <params list>=dummy, code (string), key

```

5          REV18 =          5          ;first released revision, 29-Jly-82

          1          .TXTM 1          ;set the text mode
          104000      .LOC LTP03      ;the address in DISCSUBS
104000      162 DSB83:  TIP03          ;it is actual number 162
104001          3          START-DSB83 ;address of start
104002 177422      DSB83-DSBEND      ;size

104003 54577 START: STA          3,RET ;the return address
104004 50577      STA          2,ACC2 ;will be required by SOFT
104005 44577      STA          1,ACC1 ;as above
104006 40577      STA          0,ACC0 ;as above
104007 50540      STA          2,APT  ;the pointer to the arguments
104010 4422       JSR          PICK   ;go and get the first (MODE)
104011 4407       JSR          VECEND ;jump over the vector table, etc.

```





104122	106032		SCE	0,1	; ??? is the pointer greater than 0	
104123	2457		JMP	@RET	; NO, error return	
104124	24536		LDA	1, P1DIM	; get the size of the first param	1
104125	106433		SLE	0,1	; ??? is position greater than size	
104126	2454		JMP	@RET	; YES, error return	
104127	20041		LDA	0, DBA	; used in converting relative/absoul	0
104130	24524		LDA	1, P1ADD	; the address of the destination str	1
104131	106540		SUBOL	0,1	; convert to a relative address so	1
					; that it may be used by XGETBYTE	
104132	44522		STA	1, P1ADD	; update the stored version	
104133	24523		LDA	1, P2ADD	; the address of the new string	1
104134	106540		SUBOL	0,1	; convert to a relative byte address	1
104135	44521		STA	1, P2ADD	; store it away	
104136	102420		SUBZ	0,0	; generate a zero	0
104137	40516		STA	0, P1LEN	; clear the length counter of P1\$	
104140	24514		LDA	1, P1ADD	; get the address of P1\$, we are going	1
					; to look for the length	
104141	6144	P1LUP:	XGETBYTE		; get the next character	
104142	151015		SNZ	2,2	; ??? have we found the terminator	2
104143	405		JMP	FNDP1	; YES, we have found end of P1\$	
104144	10511		ISZ	P1LEN	; there is another character	
104145	125400		INC	1,1	; set up for XGETBYTE	1
104146	773		JMP	P1LUP	; go and look for the next	
104147	0	APT:	0		; argument pointer from RUN	
104150	24505	FNDP1:	LDA	1, P1LEN	; the calculated length of P1\$	1
104151	30510		LDA	2, INSPOS	; the insertion position	2
104152	146433		SLE	2,1	; ??? is insertion in real part	
104153	2427		JMP	@RET	; NO, error returned	
104154	102420		SUBZ	0,0	; clear counter for length of P2\$	0
104155	40502		STA	0, P2LEN	; we are going to find len of P2\$	
104156	24500		LDA	1, P2ADD	; address of P2\$	1
104157	6144	P2LUP:	XGETBYTE		; get the next character	2
104160	151015		SNZ	2,2	; ??? are we at the null yet	
104161	404		JMP	FNDP2	; YES, we have found the end of P2\$	
104162	10475		ISZ	P2LEN	; No, count the extra character	
104163	125400		INC	1,1	; step the pointer	1
104164	773		JMP	P2LUP	; deal with the next position	
104165	30472	FNDP2:	LDA	2, P2LEN	; the calculated length of P2\$	2
104166	151015		SNZ	2,2	; ??? is it null string	
104167	677		JMP	EXIT	; YES, we are finished	
104170	24465		LDA	1, P1LEN	; the length of the original	1
104171	147020		ADDZ	2,1	; the new length if performed	1
104172	30470		LDA	2, P1DIM	; the maximum permitted	2
104173	132433		SLE	1,2	; ??? will the new string fit	
104174	2406		JMP	@RET	; NO, error return	
104175	24457		LDA	1, P1ADD	; get the address of the first	1
104176	20463		LDA	0, INSPOS	; where to perform the insertion	0
104177	123020		ADDZ	1,0	; the address in the string of the ins	0
104200	30455		LDA	2, P1LEN	; the length of the inserted section	2
104201	405		JMP	BRIDG	; skip over the central data	
104202	0	RET:	0		; initial values of accumulators	
104203	0	ACC2:	0			
104204	0	ACC1:	0			
104205	0	ACCO:	0			



104206	147020	BRIDG:	ADDZ	2,1	;end of old P1\$	1
104207	30450		LDA	2,P2LEN	;length of the new section	2
104210	113020		ADDZ	0,2	;A2 gets new position of the second	2
					;segment of the first string	
104211	4414		JSR	MOVE	;perform the move to make space for	????
					;P2\$ in P1\$	
104212	20444		LDA	0,P2ADD	;where is P2\$	0
104213	152000		ADC	2,2	;generate -1	2
104214	24443		LDA	1,P2LEN	;the length of the second	1
104215	147020		ADDZ	2,1	;make length -1	1
104216	107020		ADDZ	0,1	;position of end of P2\$	1
104217	30435		LDA	2,P1ADD	;address of the original string	2
104220	34441		LDA	3,INSPOS	;where in P1\$ is the insertion	3
104221	173020		ADDZ	3,2	;set A2 as the actual position	2
104222	4403		JSR	MOVE	;move P2\$ into place	????
104223	643		JMP	EXIT	;we are finished	
104224	606	JPICK:	JMP	PICK	;stepping stone	
	104225	MOVE:	;	;	A0 = where to copy to	0
			;	;	A1 = old end position (last char)	1
			;	;	A2 = new end position (first char)	2
104225	40425		STA	0,BEGS	;note the source beginning	
104226	44425		STA	1,ENDS	;note the end of the source	
104227	54617		STA	3,RETS	;note the return address	
104230	106400		SUB	0,1	;the number to move	1
104231	133020		ADDZ	1,2	;where to start in destination	2
104232	50426		STA	2,DESTC	;note where we will go to	
104233	24420	MOVLUP:	LDA	1,ENDS	;the end of the source	1
104234	6144		XGETBYTE		;get the character from the source	2
104235	141000		MOV	2,0	;move the character	0
104236	24422		LDA	1,DESTC	;current destination character	1
104237	6145		XPUTBYTE		;put the character in its new position	
104240	20413		LDA	0,ENDS	;the end of the source	0
104241	24411		LDA	1,BEGS	;beginning of destination	1
104242	106415		SNE	0,1	;??? are they equal, have we copied all	
104243	2603		JMP	@RETS	;YES, return	
104244	14414		DSZ	DESTC	;ready for next character	
104245	14406		DSZ	ENDS	;we have moved one more	
104246	765		JMP	MOVLUP	;delay with the next	
104247	0	ERRFLG: 0			;0-no error, 1-error in KEY	
104250	602	JRETNUM:	JMP	RETNUM	;stepping stone to RETNUM	
104251	615	JEXIT:	JMP	EXIT	;stepping stone to exit	
104252	0	BEGS:	0		;beginning of source for move	
104253	0	ENDS:	0		;end of source for move	
104254	0	P1ADD:	0		;address of P1\$	
104255	0	P1LEN:	0		;length of P1\$	
104256	0	P2ADD:	0		;address of P2\$	
104257	0	P2LEN:	0		;len of P2\$	
104260	0	DESTC:	0		;pointer to next destination char	





```

104325 50430 STA 2,PADDR ;the address to poke
104326 30621 LDA 2,APT ;get the address of 2nd parameter
104327 151400 INC 2,2 ;step to the third, the value
104330 151402 INC 2,2
104331 4673 JSR JPICK ;get the value
104332 30423 LDA 2,PADDR ;get the address to poke
104333 41000 STA 0,0,2 ;put the value into core
104334 715 JMP JEXIT ;we are finished

104335 30612 SOFT1: LDA 2,APT ;pointer to the string to execute
104336 145000 MOV 2,1 ;ready for passing externally
104337 20002 LDA 0,C2 ;constant for resetting APT
104340 106420 SUBZ 0,1 ;A1 now points at original APTs
104341 21003 LDA 0,3,2 ;the number type of the string
104342 34640 LDA 3,RET ;return address to RUN
104343 101133 MOVZL# 0,0,SNC ;??? is it a string
104344 1400 JMP 0,3 ;NO, error non-skip return
104345 31002 LDA 2,2,2 ;address of the string
104346 50407 STA 2,PADDR ;store it to jump through
104347 30634 LDA 2,ACC2 ;restore the original values of accs
104350 24634 LDA 1,ACC1 ;
104351 20634 LDA 0,ACC0 ;
104352 2403 JMP @PADDR ;go and execute the routine with all
; ACCS set as thogh we came from RUN

104353 0 CHKRET: 0 ;return address for the routine CHKKEY
104354 0 SWCH: 0 ;0=PEEK: 1=POKE: 2=SOFT
104355 0 PADDR: 0 ;address to poke

104356 DSBEND= ;this is the end
0 .ERR DSB83+400<. ;OVERFLOW CHECK

.END

```

```

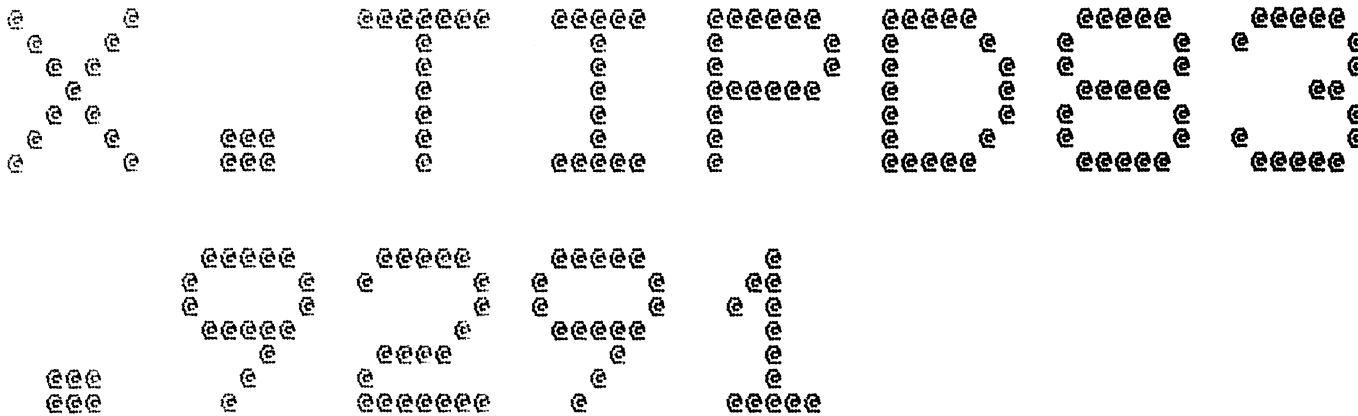
RUN
0???
2
2
1
0
1
0
3
2
2
1
0

```

ACCO	104205	ACC1	104204	ACC2	104203	APT	104147	BEGS	104252
BINDI	6115	BINMU	6116	BPI	16	BRIDG	104206	BSACF	75
BUMFU	6117	C10	30	C100	51	C1000	67	C11	31
C12	32	C13	33	C14	34	C15	35	C16	36
C160	174	C163	175	C166	176	C17	37	C170K	21
C171	177	C177	52	C1777	70	C2	2	C20	42
C200	53	C2000	71	C205	54	C215	55	C240	56
C244	57	C260	60	C271	61	C3	3	C300	62
C334	63	C37	43	C377	64	C4	24	C40	44
C400	65	C4000	72	C5	25	C6	26	C600	100
C7	27	C77	50	C774C	22	C777	66	CALL	6101
CHANN	6106	CHKKE	104303	CHKRE	104353	CM400	23	DA	160
DAC	164	DAS	165	DATAP	6110	DB	166	DBA	41
DBC	172	DBS	173	DECIM	6120	DESTC	104260	DFTCA	34106
DMCAL	34110	DQJEU	6105	DSB83	104000	DSBEN	104356	ENDS	104253
ERRF	76	ERRFL	104247	ESCF	73	ETSF	74	EXIT	104066
FINDL	6123	FIX	6121	FLAGC	6102	FLOAT	6122	FNDP1	104150
FNDP2	104165	FREEN	6107	GETBY	6124	HALTS	6153	INBYT	6125
INGFO	104261	INSTB	6126	IOCAL	34103	IOP	6	ISA2D	6127
ISA2L	6130	JEXIT	104251	JFLTO	151	JPICK	104224	JRETN	104250
LACNT	4000	LAFSE	13000	LALCO	47400	LALLO	1400	LATOE	36000
LBAKU	106000	LBILD	5000	LBUIL	4400	LCALL	75000	LCHAN	41000
LCHFLL	30000	LCHSU	61000	LCLEA	7400	LCLOS	7000	LCLPY	76000
LCNVA	11400	LCNVD	12000	LCOMM	33400	LDALC	2000	LDALL	1000
LDB7A	114000	LDB7B	114400	LDB7C	115000	LDB7D	115400	LDB7E	116000
LDB7F	116400	LDB7G	117000	LDB7H	117400	LDB7I	120000	LDEKE	52400
LDELE	3400	LDIRE	50400	LDLTP	20400	LDREN	37400	LDSB1	400
LDSB2	22400	LDSB3	47000	LDSB4	65000	LDSB5	77000	LDSB6	106400
LDSB7	113400	LECHO	37000	LEO87	105400	LERRO	23000	LFAUL	400
LFFIL	2400	LFIXD	57400	LFNDC	112000	LFNDL	20000	LFOFI	17000
LGETR	10000	LQHOP	107400	LGHOS	107000	LGMUX	16000	LHCON	17400
LIBCA	44400	LIBEN	45000	LIBTR	45400	LIDAT	103000	LLINK	35400
LLOAD	34400	LLOGI	32000	LLUIN	112400	LMAPB	73000	LMDEO	65000
LMDEF1	66000	LMDE5	71400	LMRC3	56400	LMRFH	57000	LMRFI	54000
LMTAP	55400	LMTAS	54400	LMTFP	56000	LMTFY	60400	LMTNX	55000
LMTPL	60000	LOADD	6131	LOPEN	6000	LOPNM	13400	LPATQ	110000
LPEXP	23400	LPFAB	72000	LPFLN	73400	LPFNA	3000	LPFRL	72400
LPFSE	67000	LPFSH	70000	LPFSX	70400	LPLDG	24400	LPPWR	33000
LPRAN	36400	LPRCO	71000	LPSIN	25400	LPSQR	22400	LPTAN	25000
LQIBF	63400	LQICL	63000	LQIQP	62400	LRDFH	26400	LRDIS	31400
LRDSE	110400	LREDC	50000	LREDI	11000	LREDM	14000	LREDP	74000
LRENA	15000	LREQP	53000	LRESO	42000	LRWIT	113000	LRWMB	14400
LRWSX	111400	LS105	77000	LS152	102000	LS153	101000	LS154	100400
LS156	101400	LS157	100000	LSAVE	43000	LSAVP	43400	LSEAB	64000
LSEAR	51000	LSETF	40000	LSHUF	52000	LSIGP	12400	LSING	40400
LSMCS	106400	LSPEC	27000	LSTRI	32400	LSYSC	30400	LTP01	102400
LTP03	104000	LTP04	104400	LTP05	105000	LVMUX	42400	LWRIT	47000
LXMIN	62000	MOVE	104225	MOVLU	104233	OUTBY	6132	OUTTE	6133
PIADD	104254	PIDIM	104262	PILFN	104255	PILUP	104141	P2ADD	104256
PELEN	104257	P2LUP	104157	PADDR	104355	PEEK	104263	PIB	4
PICK	104032	POKE	104263	POKE1	104323	PUTBY	6134	QCHAR	6103
QUEUE	6104	READB	6135	RELJM	6136	RET	104202	RETNU	104052
RETS	104046	REVIS	5	REVNO	104047	RJSR	6136	RNUM	104045
RTP	7	RUP	5	SBA	40	SCDCA	34147	SOFT	104263
SOFT1	104335	SPINP	6146	START	104003	STINP	6140	STINS	104070
STINT	6147	STORD	6137	STOUT	6141	SWTCH	104354	TASKQ	15
TRAPP	6142	VECEN	104020	VECTO	104012	VSIZE	104017	WRITB	6143
XGETB	6144	XPUTB	6145	ABA	14	BPS	77	BSA	10

----

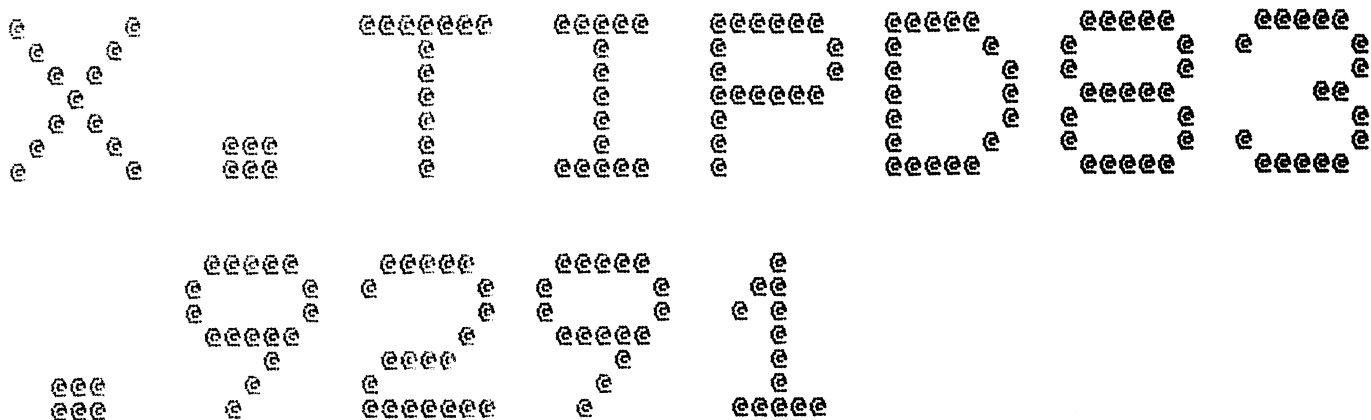
. DA	174	. DAG	175	. DB	176	. DB3	177	. FLTO	152
. HBA	11	. HXA	12	. INFO	100	. INTR	111	. LCM	114
. NRET	112	. SRET	113	. SSA	13				



Spool Queue Line #: 29  
IRIS LU/Filename : 18/X.TIPD83.9291

Printed on/at : FEB 7, 1990 14:52:51  
For Group/User: 0, 1  
On Port No: 5

Print control parameters :  
Printer Class code : 0  
Form Code/paper type : ?  
Print Priority (0-9) : 5  
Starting Page Number : 1  
This is copy number : 1  
Keep file (Y/N) : Y  
Notify User when done: N  
Comments, optional : For RELSE CNTRL



Spool Queue Line #: 29  
IRIS LU/Filename : 18/X.TIPDB3.9291

Printed on/at : FEB 7, 1990 14:52:57  
For Group/User: 0, 1  
On Port No: 5

Print control parameters :  
Printer Class code : 0  
Form Code/paper type : ?  
Print Priority (0-9) : 5  
Starting Page Number : 1  
This is copy number : 1  
Keep file (Y/N) : Y  
Notify User when done: N  
Comments, optional : For RELSE CNTRL





\*\*\*\*\* J O B S T A T I S T I C S \*\*\*\*\*

2	TOTAL # DUPLICATE KEYS
0	TOTAL # DIR. RE-ORGS
198	TOTAL # KEYS INSERTED
0	TOTAL # ASSEMBLY ERRS

.ERR	3.098						
ACCO	2.057	2.239:	3.085				
ACC1	2.056	2.238:	3.084				
ACC2	2.055	2.237:	3.083				
APT	2.058 3.026	2.131 3.027	2.155 3.049	2.158 3.065	2.168 3.073	2.206:	3.025
BEGS	2.267	2.279	2.293:				
BRIDG	2.234	2.243:					
C2	2.156	3.020	3.075				
C6	3.050						
CHKKE	3.023	3.040:					
CHKRE	3.044	3.056	3.090:				
DBA	2.187						
DECIM	2.106	2.137					
DESTC	2.272	2.276	2.282	2.299:	3.052	3.054	
DSB83	2.050:	2.051	2.052	3.098			
DSREN	2.052	3.096=					
ENDS	2.268	2.273	2.278	2.283	2.294:		
ERRFL	2.127	2.138	2.286:	3.031	3.058		
EXIT	2.141:	2.224	2.259	2.289			
FIX	2.107						
FLOAT	2.130						
FNDP1	2.201	2.208:					
FNDP2	2.217	2.222:					
INSP0	2.180	2.209	2.231	2.256	2.300:		
JEXIT	2.289:	3.071					

JPICK	2.261:	3.028	3.053	3.068			
JRETN	2.288:	3.036	3.059				
LTP03	2.048						
MOVE	2.248	2.258	2.263:				
MOVLU	2.273:	2.284					
P1ADD	2.160	2.188	2.191	2.197	2.230	2.255	2.295:
P1DIM	2.165	2.184	2.227	2.301:			
P1LEN	2.196	2.202	2.208	2.225	2.233	2.296:	
P1LUP	2.199:	2.204					
P2ADD	2.170	2.192	2.194	2.214	2.250	2.297:	
P2LEN	2.213	2.218	2.222	2.244	2.252	2.298:	
P2LUP	2.215:	2.220					
PADDR	3.064	3.069	3.082	3.086	3.092:		
PEEK	2.072	2.306:					
PIB	3.045						
PICK	2.059	2.093:	2.176	2.261			
POKE	2.073	3.006:					
POKE1	3.034	3.062:					
RET	2.054 2.141 2.236:	2.083 2.163 3.078	2.086 2.173	2.102 2.183	2.108 2.186	2.134 2.211	2.140 2.229
RETNU	2.129:	2.288					
RETS	2.103	2.110	2.117:	2.269	2.281		
REVIS	2.043=	2.116					
REVND	2.070	2.120:					
RNUM	2.116:	2.128					
SOFT	2.074	3.013:					

SOFT1	3.063	3.073:				
START	2.051	2.054:				
STINS	2.071	2.147:				
SWTCH	3.022	3.032	3.062	3.091:		
TIP03	2.050					
VEGEN	2.060	2.082:				
VECTO	2.070:	2.071	2.072	2.073	2.074	2.079:
VSIZE	2.079:	2.084				
XGETB	2.199	2.215	2.274			
XPUTB	2.277					