

Appendix  
C

**SAMPLE  
PROGRAMS**

*Get on the  
Fast Track!*



TM

**SYS-ED/  
COMPUTER  
EDUCATION  
TECHNIQUES, INC.**

```
/*          */  
/*  ALLOCATE SYSPROC  */  
/*          */
```

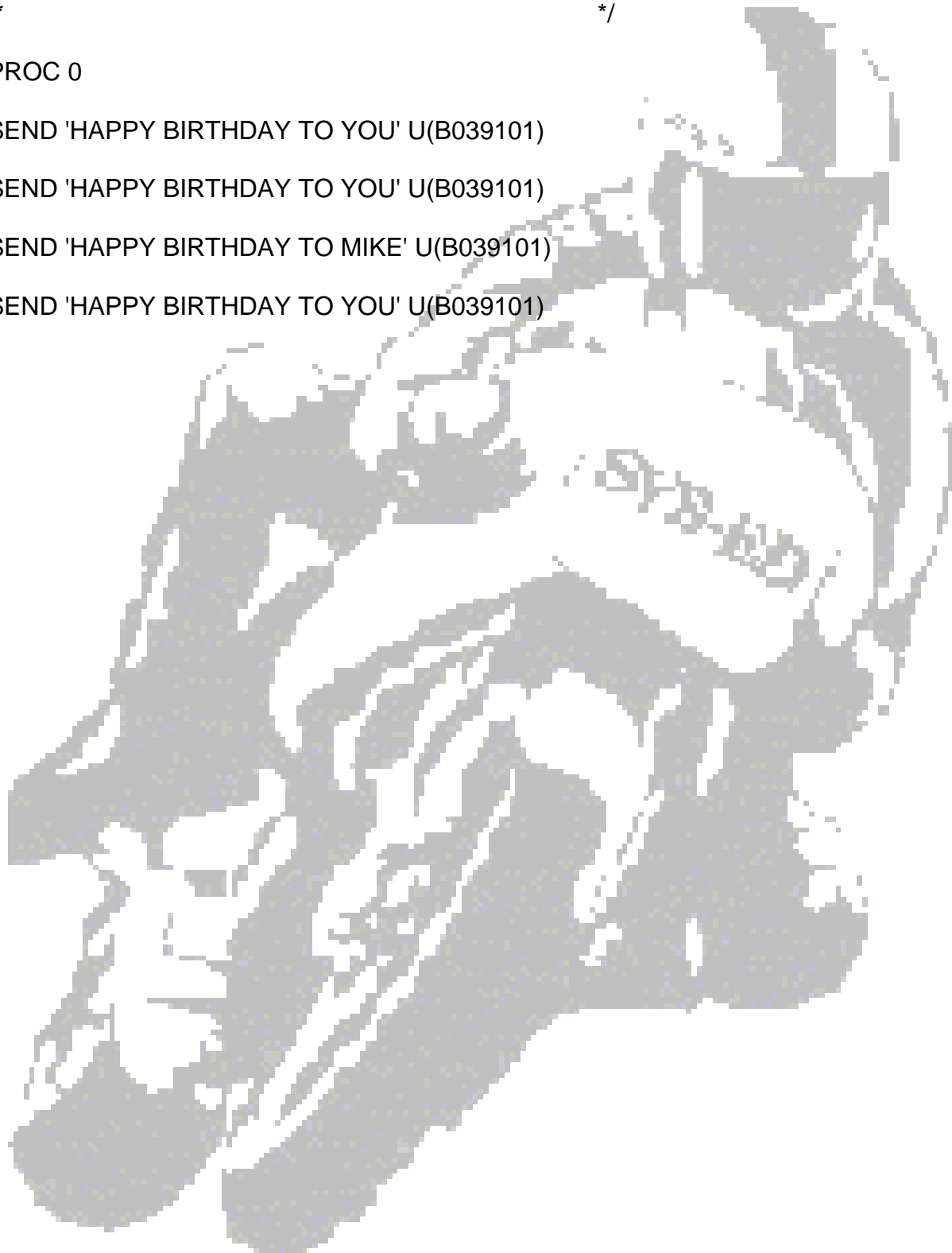
PROC 0

FREE DD(SYSPROC)

```
ALLOC      DD(SYSPROC)      -  
           DA('SYS1.CLISTLIB'  
             MYLIB.CLIST)  SHR
```

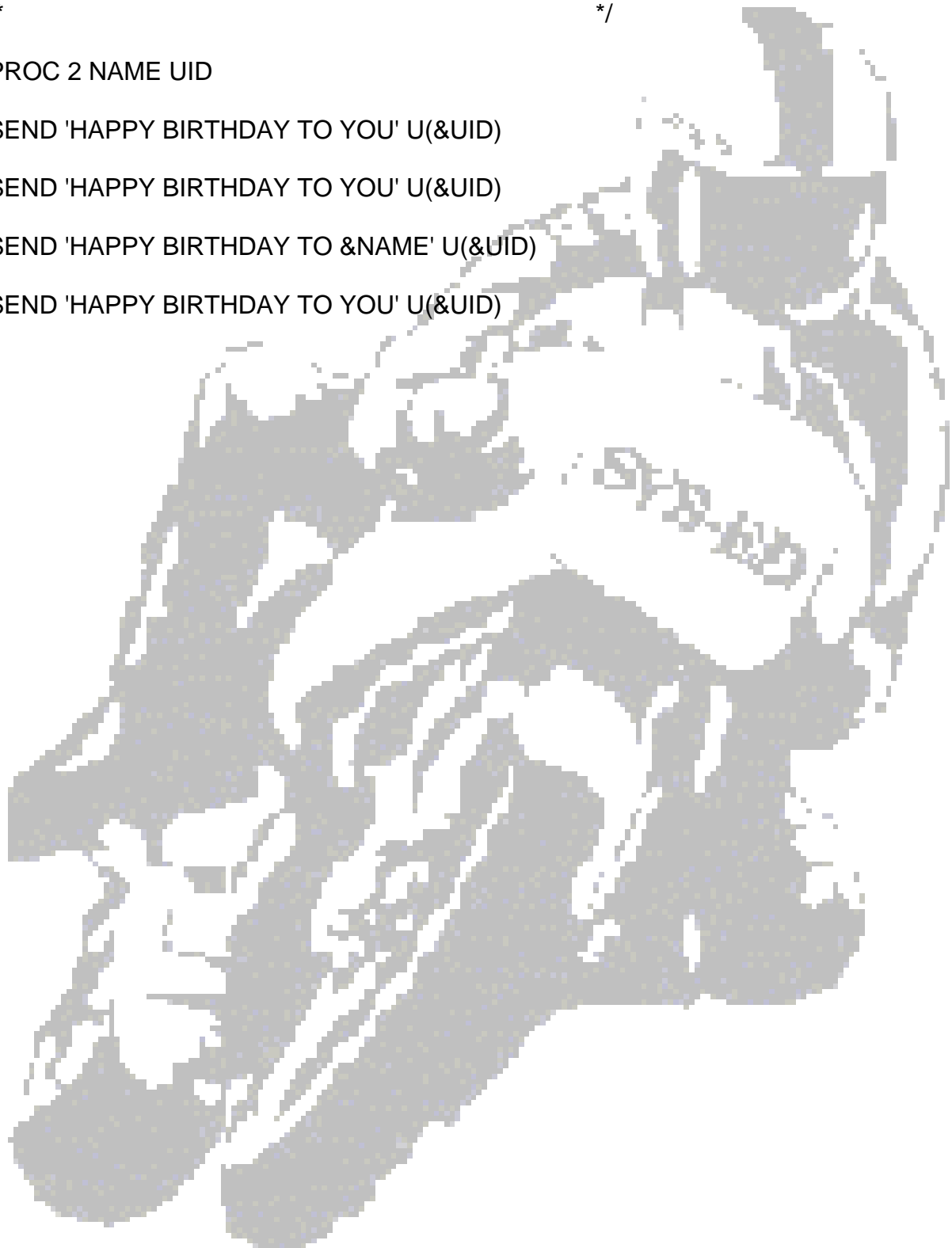
```
/*                               */  
/* SEND A HAPPY BIRTHDAY MESSAGE */  
/*                               */
```

```
PROC 0  
  
SEND 'HAPPY BIRTHDAY TO YOU' U(B039101)  
SEND 'HAPPY BIRTHDAY TO YOU' U(B039101)  
SEND 'HAPPY BIRTHDAY TO MIKE' U(B039101)  
SEND 'HAPPY BIRTHDAY TO YOU' U(B039101)
```



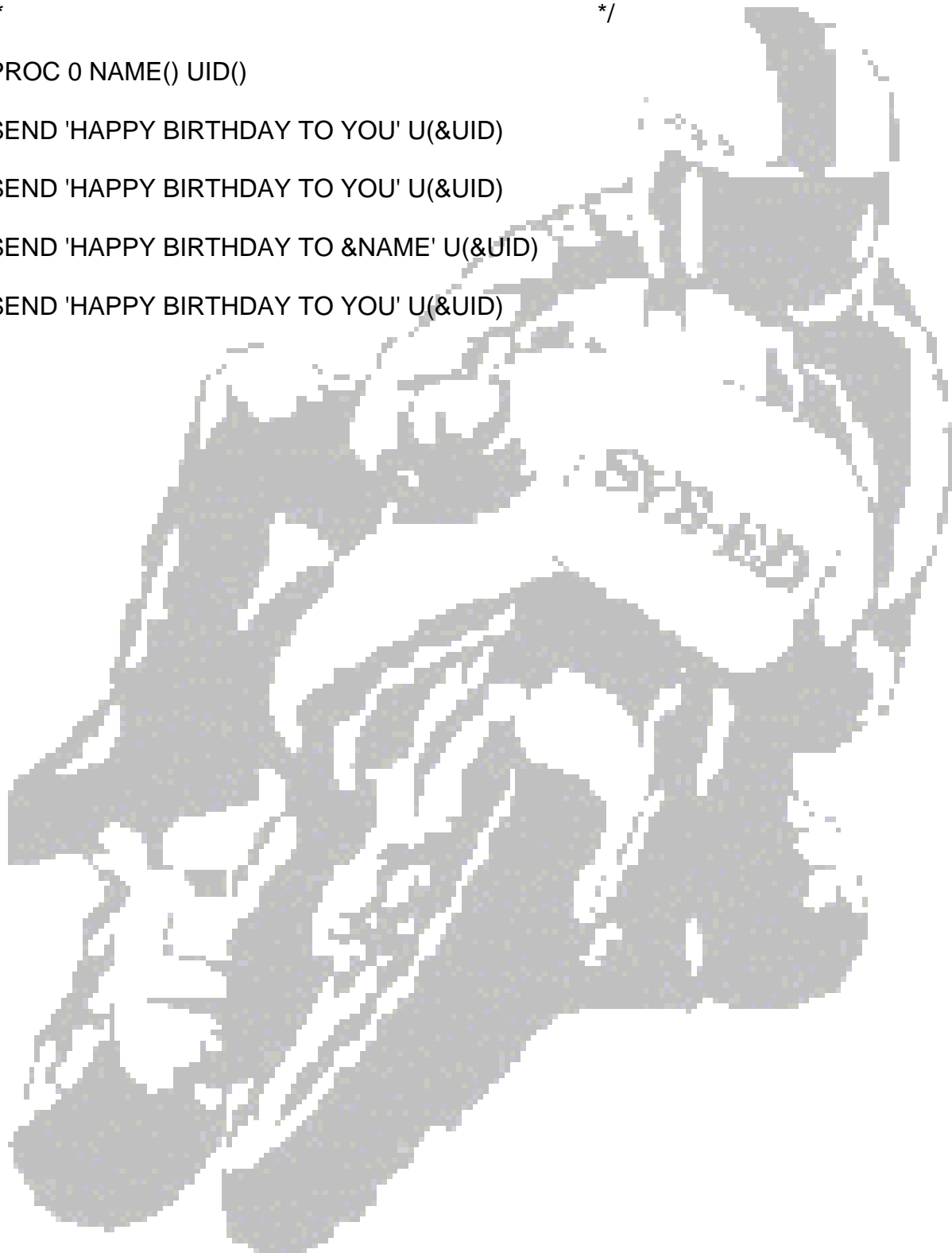
```
/*                               */  
/* SEND A HAPPY BIRTHDAY MESSAGE */  
/*                               */
```

```
PROC 2 NAME UID  
  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)  
SEND 'HAPPY BIRTHDAY TO &NAME' U(&UID)  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)
```



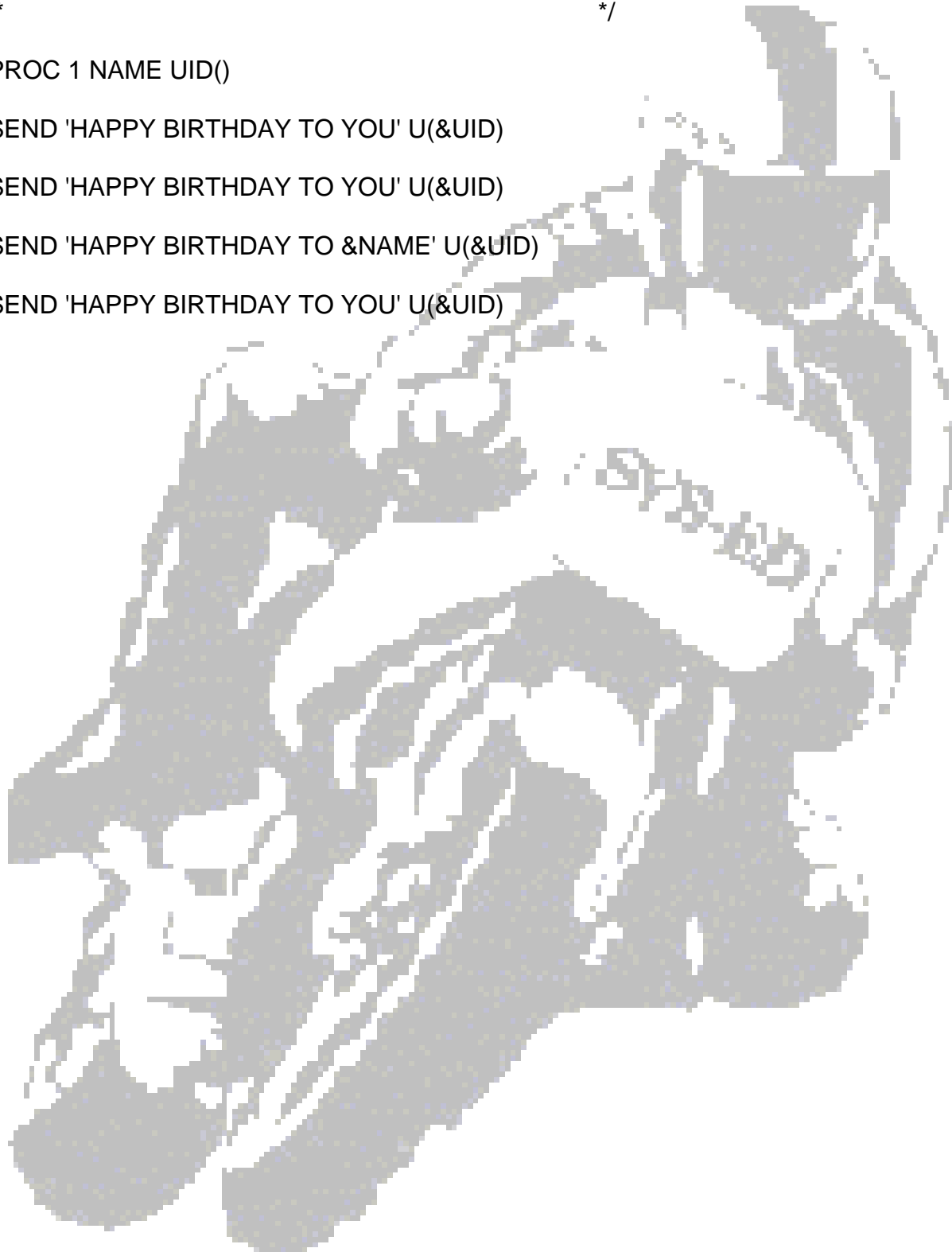
```
/*                               */  
/* SEND A HAPPY BIRTHDAY MESSAGE */  
/*                               */
```

```
PROC 0 NAME() UID()  
  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)  
SEND 'HAPPY BIRTHDAY TO &NAME' U(&UID)  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)
```



```
/*                               */  
/* SEND A HAPPY BIRTHDAY MESSAGE */  
/*                               */
```

```
PROC 1 NAME UID()  
  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)  
  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)  
  
SEND 'HAPPY BIRTHDAY TO &NAME' U(&UID)  
  
SEND 'HAPPY BIRTHDAY TO YOU' U(&UID)
```



```
/*                                     */
/* CALCULATE TRACK INFORMATION        */
/*                                     */

PROC 2 RECL FACTOR

/* CALCULATE THE BLOCK SIZE          */
SET &BLKSIZE = &RECL * &FACTOR

/* CALCULATE THE NUMBER OF BLOCKS IN A TRACK */
/* ASSUME THAT A TRACK SIZE IS 47,000 BYTES */
SET &BLK_IN_TRACK = 47000 / &BLKSIZE
SET &WASTE = 47000 - (&BLKSIZE * &BLK_IN_TRACK)

WRITE RECORD LENGTH = &RECL
WRITE BLOCKING FACTOR = &FACTOR
WRITE BLOCK SIZE = &BLKSIZE
WRITE BLOCKS IN A TRACK = &BLK_IN_TRACK
WRITE WASTED SPACE = &WASTE

EXIT
```

```
/*                                     */
/* CALCULATE TRACK INFORMATION        */
/*                                     */

PROC 0

/* CALCULATE THE BLOCK SIZE          */

WRITENR ENTER THE RECORD LENGTH ==>
READ RECL

WRITENR ENTER THE NUMBER OF RECORDS IN A BLOCK ==>
READ FACTOR

SET &BLKSIZE = &RECL * &FACTOR

/* CALCULATE THE NUMBER OF BLOCKS IN A TRACK */
/* ASSUME THAT A TRACK SIZE IS 47,000 BYTES */
SET &BLK_IN_TRACK = 47000 / &BLKSIZE
SET &WASTE = 47000 - (&BLKSIZE * &BLK_IN_TRACK)

WRITE RECORD LENGTH = &RECL
WRITE BLOCKING FACTOR = &FACTOR
WRITE BLOCK SIZE = &BLKSIZE
WRITE BLOCKS IN A TRACK = &BLK_IN_TRACK
WRITE WASTED SPACE = &WASTE

EXIT
```

```
/*                                     */
/* EXECUTE A LISTDS COMMAND          */
/*                                     */
PROC 1 DSN OPTION

IF &OPTION = OPTION THEN +
  DO
    WRITE THE VALID OPTIONS ARE MEMBER, HISTORY OR STATUS
    READ OPTION
  END

LISTDS &DSN &OPTION

EXIT
```

```
/* */
/* CONTROL VARIABLES */
/* */

WRITE TODAY IS &SYSDATE /* YY/MM/DD */
WRITE THE TIME IS &SYSTIME /* HH:MM:SS */
WRITE YOUR USERID IS &SYSUID
WRITE YOUR PREFIX IS &SYSPREF
WRITE ISPF IS &SYSISPF /* ACTIVE, NOT ACTIVE */
WRITE YOU ARE RUNNING &SYSENV /* FORE, BACK */
```

```
/* */
/* USE OF &SYSOUTTRAP */
/* CAPTURE MEMBERS NAMES FROM A LISTDS */
/* */

PROC 0 DSN(DEFAULT)

IF &DSN = DEFAULT THEN +
  DO
    WRITE ENTER DATA SET NAME
    READ DSN
  END

SET SYSOUTTRAP = 500 /* MAXIMUM LINES TO TRAP */
LISTDS &DSN MEMBERS
SET MAXLINES = &SYSOUTLINE /* NUMBER OF LINES TRAPPED */
SET SYSOUTTRAP = 0 /* RESET TO 0 */
SET I = 5 /* START AT LINE 5 */

DO WHILE &I < &MAXLINES
  SET MEMBER = &&SYSOUTLINE&I
  WRITE MEMBER IS &MEMBER
  SET &I = &I + 1
END

EXIT
```

```
/*                                                    */
/* CALCULATE TRACK INFORMATION                        */
/*                                                    */

PROC 0

/* CALCULATE THE BLOCK SIZE                            */

SET FLAG = 'Y'
DO WHILE FLAG = 'Y'

    WRITENR ENTER THE RECORD LENGTH ==>
    READ RECL

    IF &LENGTH(RECL) = 0 THEN EXIT

    IF &DATATYPE(&RECL) = NUM THEN SET FLAG = 'N'
    ELSE WRITE INVALID RECORD LENGTH - TRY UNTIL YOU GET IT RIGHT

END

SET FLAG = 'Y'
DO WHILE FLAG = 'Y'

    WRITENR ENTER THE NUMBER OF RECORDS IN A BLOCK ==>
    READ FACTOR

    IF &LENGTH(&FACTOR) = 0 THEN EXIT

    IF &DATATYPE(&FACTOR) = NUM THEN SET FLAG = 'N'
    ELSE WRITE INVALID FACTOR - PLEASE TRY AGAIN

END
```

```
SET &BLKSIZE = &RECL * &FACTOR
```

```
/* CALCULATE THE NUMBER OF BLOCKS IN A TRACK */
```

```
/* ASSUME THAT A TRACK SIZE IS 47,000 BYTES */
```

```
SET &BLK_IN_TRACK = 47000 / &BLKSIZE
```

```
SET &WASTE = 47000 - (&BLKSIZE * &BLK_IN_TRACK)
```

```
WRITE RECORD LENGTH = &RECL
```

```
WRITE BLOCKING FACTOR = &FACTOR
```

```
WRITE BLOCK SIZE = &BLKSIZE
```

```
WRITE BLOCKS IN A TRACK = &BLK_IN_TRACK
```

```
WRITE WASTED SPACE = &WASTE
```

```
EXIT
```

## THE DELETEDS CLIST

```
/******  
/* THIS CLIST PROMPTS THE USER FOR THE NAMES OF THE DATA */  
/* SETS TO BE DELETED, ONE AT A TIME. */  
/******
```

```
SET DONE=NO  
DO WHILE &DONE=NO  
    WRITE Enter the name of the data set you wan deleted.  
    WRITE Omit the identification qualifier (userid).  
    WRITE Do not put the name in quotes.  
    WRITE When you are finished, type an 'F'.  
    READ DSN  
    IF &DSN = F THEN SET DONE=YES  
    ELSE DELETE &DSN
```

```
END
```

THE CALC CLIST  
PROC 3 FVALUE OPER LVALUE

```
/******  
**/  
/* DISPLAY THE ENTIRE EQUATION AT THE TERMINAL, INCLUDING  
*/ /* THE RESULT OF THE EXPRESSION.  
*/  
/******  
*/
```

WRITE &FVALUE&OPER&VALUE = &EVAL(&FVALUE&OPER&VALUE)

THE OUTPUT CLIST  
PROC 1 DSN NEWDSN

CONTROL NOMSG

free f(sysprint,sysut1,sysut2,sysin)

CONTROL MSG

alloc f(sysprint) dummy reuse

alloc f(sysut1) da('&SYSPREF..&DSN..LIST') shr reuse

alloc f(sysut2) da('&SYSPREF..&NEWDSN..BKP')+  
new space (5,1) tracks+

LRECL(80) BLKSIZE(3120) RECFM(F,B)

alloc f(sysin) dummy reuse

call 'sys1.linklib(iebgener)'

free f(sysut1,sysut2,sysprint,sysin)

```

THE SUBMITDS CLIST
PROC 2 DSN ACCT CLASS(C)
/*****
**/
/*IF &ACCT IS INVALID, CONTINUE PROMPTING UNTIL THE USER ENTERS
*/
/*AN ACCEPTABLE VALUE. */
/*****
*/

DO WHILE &LENGTH(&STR(&ACCT)) <= 4 OR &DATATYPE(&ACCT) <= NUM
WRITE Your account number is invalid.
WRITE Reenter a four-digit number.
READ ACCT
END
/*****
*/
/* ONCE ACCOUNT NUMBER HAS BEEN VERIFIED, SUBMIT THE JOB.
*/
/*****
*/

SET SLSHASK=&STR(/*)/ * Set the /* required for jcl comment
statement*/
SUBMIT * END($$)
//&SYSUID.1 JOB &ACCT,&SYSUID,CLASS=&CLASS,NOTIFY=&SYSUID
/&SLSHASK THIS STEP COPIES THE INPUT DATASET TO SYSOUT=A
//COPY EXEC PGM=COPYDS
//SYSUT1 DD DSN=&SYSUID..&DSN,DISP=SHR
//SYSUT2 DD SYSOUT=A
$$

```

```
THE SUBMITFQ CLIST
PROC 2 DSN ACCT CLASS(C)
/*****
**/
/* IF &ACCT IS INVALID, CONTINUE PROMPTING UNTIL THE USER ENTERS
*/
/* AN ACCEPTABLE VALUE. */
/*****
**/

DO WHILE &LENGTH(&STR(&ACCT)) ≠4 OR &DATATYPE(&ACCT) ≠ NUM
WRITE Your account number is invalid.
WRITE Reenter a four-digit number.
READ ACCT
END
/*****
**/
/* IF THE DATA SET IS FULLY QUALIFIED, REMOVE THE QUOTES.
*/
/* OTHERWISE, PREFIX THE CURRENT USERID.
*/
/*****
**/

SET SLSHASK=&STR(/*) /* Set the /* required for the jcl comment statement
*/
SUBMIT * END($$)
//&SYSUID.1 JOB &ACCT,&SYSUID,CLASS=O&CLASS
/&SLSHASK THIS STEP COPIES THE INPUT DATASET TO SYSOUT=A
//COPY EXEC PGM=COPYDS
//SYSUT1 DD DSN=&DSN,DISP=SHR
//SYSUT2 DD SYSOUT=A
$$
```