```
    * ACTION.S -- Action Procedures for Small Runoff; Dec 6, 1981
    BEGIN ACTION;
    *---------------
      ENT BRK,PUTOUT,SPACE;
      ENT GETVAL,SETVAL,LEADBL,UNDERL;
      ENT INSERT,DELETE,SPREAD;
      ENT GETWRD,GOTWRD;
      ENT NADA,PLUS,MINUS;
      ENT POS,NEG;
    *---------------
      EXT TRUE,FALSE,NULL;
      EXT OUTFILE;
      EXT INBUF,OUTBUF;
      EXT PSEUDO;
      EXT M1VAL,M2VAL,M3VAL,M4VAL;
      EXT FILLVAL,CURPAG,NEWPAG;
      EXT LINENO,TIVAL,LSVAL,INVAL;
      EXT BOTTOM,OUTP,OUTW;
      EXT HEAD,FOOT;
      EXT DIRRT,OUTWRDS,NEXTRA,MLINEW;
      EXT WRDBUF,WRDLEN;
    *---------------
      EXT PROC WRITE,PROC STREQ,PROC CAT2;
      EXT PROC PWRITE,PROC QWRITE,PROC WRITELN;
    *---------------
      DCL IZ1;
      DCL PLUS=('+'),MINUS=('-'),BLANK=(' ');
      MSG BLINE=' ';                          * Blank line
    *---------------
    PROC SKIPLINES(NZ1);                      * Skip Lines
1     IZ1=1;
1     DO WHILE IZ1 LE NZ1;
2       CALL WRITE(OUTFILE,BLINE);
2       IZ1=IZ1+1;
2       ENDDO
1     RETURN
1     ENDPROC
0   *---------------
0     DCL LX1;                                * Line Length
0     DCL CZ1;                                * Temp char buffer
0   *---------------
0   PROC PUTTL(SZ3);                          * Output Title with Page Number
1     LX1=SZ3;                                * Length of arg string
1     IZ1=1;
1     DO WHILE IZ1 LE LX1;
2       CZ1=SZ3(IZ1);
2       IF CZ1 EQ PSEUDO;
3         THEN CALL QWRITE(OUTFILE,CURPAG);
3         ELSE CALL PWRITE(OUTFILE,CZ1);
3         ENDIF
2       IZ1=IZ1+1;
2       ENDDO
1     CALL WRITELN(OUTFILE);
1     RETURN
1     ENDPROC
0   *---------------
0     DCL IZ2;
0   *---------------
0   PROC PHEAD;                       * Write header at top of page
1     CURPAG=NEWPAG;
```

```
1    NEWPAG=NEWPAG+1;
1    IF M1VAL GT 0;
2      THEN
2        IZ2=M1VAL-1;
2        CALL SKIPLINES(IZ2);
2        CALL PUTTL(HEAD);
2      ENDIF
1    CALL SKIPLINES(M2VAL);
1    LINENO=M1VAL+M2VAL;
1    RETURN
1    ENDPROC
0  *----------------
0  PROC PFOOT;                     * Write footer at bottom of page
1    CALL SKIPLINES(M3VAL);
1    IF M4VAL GT 0;
2      THEN
2        CALL PUTTL(FOOT);
2        IZ2=M4VAL-1;
2        CALL SKIPLINES(IZ2);
2      ENDIF
1    LINENO=0;
1    RETURN
1    ENDPROC
0  *---------------
0    DCL IZ3;
0    DCL OUTP1;
0  *---------------
0  PROC PUTOUT;                    * Write out the currant line
1    IF LINENO EQ 0;
2      THEN CALL PHEAD;
2      ELSE
2        IF LINENO GE BOTTOM;
3          THEN CALL PHEAD;
3          ENDIF
2      ENDIF
1    IZ3=1;
1    DO WHILE IZ3 LE TIVAL;
2      CALL PWRITE(OUTFILE,BLANK);
2      IZ3=IZ3+1;
2      ENDDO
1    TIVAL=INVAL;
1    OUTP1=OUTP-1;
1    IZ3=1;
1    DO WHILE IZ3 LE OUTP1;
2      CZ1=OUTBUF(IZ3);
2      IF CZ1 EQ PSEUDO;
3        THEN CALL PWRITE(OUTFILE,BLANK);
3        ELSE CALL PWRITE(OUTFILE,CZ1);
3        ENDIF
2      IZ3=IZ3+1;
2      ENDDO
1    CALL WRITELN(OUTFILE);
1    LINENO=LINENO+1;
1    OUTP=0;
1    OUTW=0;
1    IF LINENO GT (BOTTOM-LSVAL);
2      THEN IZ3=BOTTOM-LINENO;
2      ELSE IZ3=LSVAL-1;
2      ENDIF
1    CALL SKIPLINES(IZ3);
```

```
1     LINENO=LINENO+LSVAL-1;
1     IF LINENO GE BOTTOM;
2       THEN CALL PFOOT;
2       ENDIF
1     RETURN
1     ENDPROC
0   *---------------
0     DCL IZ4;
0   *---------------
0   PROC BRK;                       * Write out current line using PUTOUT
1     IF OUTP GT 0;
2       THEN CALL PUTOUT;
2       ENDIF
1     OUTP=0;
1     OUTW=0;
1     OUTWRDS=0;
1     RETURN
1     ENDPROC
0   *---------------
0     DCL IZ5,OLDLSVAL;
0   *---------------
0   PROC SPACE(NZ2);                * Write out NZ2 blank lines
1     CALL BRK;
1     IZ4=NZ2;
1     IF IZ4 GT BOTTOM-LINENO;
2       THEN IZ4=BOTTOM-LINENO;
2       ENDIF
1     OLDLSVAL=LSVAL;
1     LSVAL=1;
1     IZ5=1;
1     DO WHILE IZ5 LE IZ4;
2       CALL PUTOUT;
2       IZ5=IZ5+1;
2       ENDDO
1     LSVAL=OLDLSVAL;
1     RETURN
1     ENDPROC
0   *---------------
0   PROC INDIG(CH8);                * True if arg is a digit
1     IF CH8 LT '0';
2       THEN RETURN FALSE;
2       ELSE
2         IF CH8 LE '9';
3           THEN RETURN TRUE;
3           ELSE RETURN FALSE;
3           ENDIF
2       ENDIF
1     ENDPROC
0   *---------------
0   PROC INTVAL(CH7);               * True if arg is +, - or digit
1     IF CH7 EQ PLUS;
2       THEN RETURN TRUE;
2       ELSE
2         IF CH7 EQ MINUS;
3           THEN RETURN TRUE;
3           ELSE RETURN INDIG(CH7);
3           ENDIF
2       ENDIF
1     ENDPROC
0   *---------------
```

```
0    DCL CH;
0    DCL NADA,POS,NEG;
0    DCL IZ6,LX6,GETV;
0  *---------------
0  PROC GETVAL(GETVA);              * Get value of paramater
1    NADA=FALSE;
1    POS=FALSE;
1    NEG=FALSE;
1    LX6=INBUF;                     * Length of INBUF
1    IZ6=1;
1    DO WHILE IZ6 LT LX6;           * Skip over non-blanks
2      IF INBUF(IZ6) EQ BLANK;
3        THEN EXIT;
3        ENDIF
2      IZ6=IZ6+1;
2      ENDDO
1    DO WHILE IZ6 LT LX6;           * Now skip over blanks
2      IF INBUF(IZ6) NE BLANK;
3        THEN EXIT;
3        ENDIF
2      IZ6=IZ6+1;
2      ENDDO
1    CH=INBUF(IZ6);
1    IF NOT INTVAL(CH);
2      THEN NADA=TRUE;
2      ELSE
2        IF CH EQ PLUS;
3          THEN POS=TRUE;
3          ELSE
3            IF CH EQ MINUS;
4              THEN NEG=TRUE;
4              ENDIF
3          ENDIF
2      ENDIF
1    IF POS OR NEG;
2      THEN IZ6=IZ6+1;
2      ENDIF
1    GETV=0;
1    DO WHILE IZ6 LE LX6;
2      CH=INBUF(IZ6);
2      IF INDIG(CH);
3        THEN GETV=10*GETV+CH-'0';
3        ELSE EXIT;
3        ENDIF
2      IZ6=IZ6+1;
2      ENDDO
1    GETVA=GETV;
1    RETURN
1    ENDPROC
0  *---------------
0  * Set the value of the command parameter within limits
0  PROC SETVAL(PARAM,VAL,MINVAL,MAXVAL);
1    IF POS;
2      THEN PARAM=PARAM+VAL;
2      ELSE
2        IF NEG;
3          THEN PARAM=PARAM-VAL;
3          ELSE PARAM=VAL;
3          ENDIF
2      ENDIF
```

```
1      IF PARAM GT MAXVAL;
2        THEN PARAM=MAXVAL;
2        ELSE
2          IF PARAM LT MINVAL;
3            THEN PARAM=MINVAL;
3            ENDIF
2        ENDIF
1      RETURN
1      ENDPROC
0    *---------------
0      DCL IY1,LY1;
0      DCL STILLBLANK;
0      DCL ONE=1;
0    *---------------
0    * Delete leading blanks from INBUF, set TIVAL
0    PROC LEADBL;
1      LY1=INBUF;                      * Length of INBUF
1      IF LY1 GT 0;
2        THEN
2          IY1=1;
2          STILLBLANK=TRUE;
2          DO WHILE IY1 LE LY1;
3            IF NOT STILLBLANK;
4              THEN EXIT;
4              ENDIF
3            IF INBUF(IY1) NE BLANK;
4              THEN STILLBLANK=FALSE;
4              ELSE IY1=IY1+1;
4              ENDIF
3            ENDDO
2          IY1=IY1-1;
2          IF IY1 NE LY1;
3            THEN
3              IF NOT FILLVAL;
4                THEN TIVAL=IY1+INVAL;
4                ENDIF
3            ENDIF
2          IF IY1 GT 0;
3            THEN CALL DELETE(INBUF,ONE,IY1);
3            ENDIF
2        ENDIF
1        RETURN
1        ENDPROC
0    *---------------
0      DCL LOWERA=97,LOWERZ=122;
0    *---------------
0    PROC INALPH(CH9);                 * Return true if CH9 is a letter
1      IF CH9 LT 'A';
2        THEN RETURN FALSE;
2        ELSE
2          IF CH9 LE 'Z';
3            THEN RETURN TRUE;
3            ELSE
3              IF CH9 LT LOWERA;
4                THEN RETURN FALSE;
4                ELSE
4                  IF CH9 LE LOWERZ;
5                    THEN RETURN TRUE;
5                    ENDIF
4                ENDIF
```

```
3           ENDIF
2       ENDIF
1     RETURN FALSE;
1     ENDPROC
0   *---------------
0     DCL IY2,LY2;
0     DCL UNDERS=(2,8,95);          * Backspace, Underscore
0   *---------------
0   PROC UNDERL;
1     IY2=1;
1     DO WHILE IY2 LE INBUF;
2       CH=INBUF(IY2);
2       IF INALPH(CH);
3         THEN
3           IF IY2 EQ INBUF;
4             THEN CALL CAT2(INBUF,UNDERS);
4             ELSE
4               LY2=IY2+1;
4               CALL INSERT(UNDERS,INBUF,LY2);
4             ENDIF
3           IY2=IY2+3;
3         ELSE IY2=IY2+1;
3         ENDIF
2       ENDDO
1     RETURN
1     ENDPROC
0   *---------------
0     DCL IS2,IY3,LY3;
0   *---------------
0   PROC INSERT(NEWS,OLDS,ATS);
1     IS2=OLDS;
1     IY3=NEWS+OLDS;
1     OLDS=IY3;
1     DO WHILE IS2 GE ATS;
2       OLDS(IY3)=OLDS(IS2);
2       IY3=IY3-1;
2       IS2=IS2-1;
2       ENDDO
1     IS2=1;
1     IY3=ATS;
1     DO WHILE IS2 LE NEWS;
2       OLDS(IY3)=NEWS(IS2);
2       IY3=IY3+1;
2       IS2=IS2+1;
2       ENDDO
1     RETURN
1     ENDPROC
0   *---------------
0     DCL LD8,ID8,JD8;
0   *---------------
0   * Delete L charactes from DSTR starting at I
0   PROC DELETE(DSTR,ID9,LD9);
1     ID8=ID9+LD9;
1     JD8=ID9;
1     LD8=DSTR;
1     DO WHILE JD8 LE LD8;
2       DSTR(JD8)=DSTR(ID8);
2       JD8=JD8+1;
2       ID8=ID8+1;
2       ENDDO
```

```
1     DSTR=DSTR-LD9;
1     RETURN
1     ENDPROC
0   *---------------
0     DCL IU1,JU1,LU1;
0     DCL FINWRD,GOTWRD;
0   *---------------
0   PROC GETWRD(IPOS);
1     LU1=INBUF;
1     GOTWRD=FALSE;
1     DO WHILE IPOS LE LU1;
2       IF GOTWRD;
3         THEN EXIT;
3         ELSE
3           IF INBUF(IPOS) EQ BLANK;
4             THEN IPOS=IPOS+1;
4             ELSE GOTWRD=TRUE;
4             ENDIF
3         ENDIF
2       ENDDO
1     JU1=IPOS;
1     IF GOTWRD;
2       THEN
2         FINWRD=FALSE;
2         DO WHILE NOT FINWRD;
3           IF JU1 GT LU1;
4             THEN FINWRD=TRUE;
4             ELSE
4               IF INBUF(JU1) EQ BLANK;
5                 THEN FINWRD=TRUE;
5                 ENDIF
4             ENDIF
3           JU1=JU1+1;
3           ENDDO
2         IU1=1;
2         LU1=JU1-IPOS-1;
2         DO WHILE IU1 LE LU1;
3           WRDBUF(IU1)=INBUF(IPOS+IU1-1);
3           IU1=IU1+1;
3           ENDDO
2         WRDLEN=LU1;
2         IPOS=JU1;
2       ENDIF
1     RETURN
1     ENDPROC
0   *---------------
0     DCL IV1,JV1,LV1;
0     DCL NE,NB,NHOLES;
0   *---------------
0   PROC SPREAD;                    * Justify a line
1     IF NEXTRA GT 0;
2       THEN
2         IF OUTWRDS GT 1;
3           THEN
3             DIRRT=NOT DIRRT;
3             NHOLES=OUTWRDS-1;
3             NE=NEXTRA;
3             IV1=OUTP-1;
3             JV1=IV1+NE;
3             IF JV1 GT MLINEW;
```

```
4               THEN JV1=MLINEW;
4               ENDIF
3             DO WHILE IV1 LT JV1;
4               OUTBUF(JV1)=OUTBUF(IV1);
4               IF OUTBUF(IV1) EQ BLANK;
5                 THEN
5                   IF DIRRT;
6                     THEN NB=NE/NHOLES;
6                     ELSE NB=(NE-1)/NHOLES+1;
6                     ENDIF
5                   NHOLES=NHOLES-1;
5                   NE=NE-NB;
5                   DO WHILE NB GT 0;
6                     NB=NB-1;
6                     JV1=JV1-1;
6                     OUTBUF(JV1)=BLANK;
6                     ENDDO
5                 ENDIF
4               IV1=IV1-1;
4               JV1=JV1-1;
4               ENDDO
3           ENDIF
2       ENDIF
1     RETURN
1     ENDPROC
0   *---------------
0   END
 NO ERRORS DETECTED
```