S: I've got it! Now, I divide the macro body into two parts as follows:

(P8)

```
O9000 ;
#100 = #18 ;
#101 = #26 ;
#102 = #21 ;
#103 = #9 ;
M99 ;
```

(P9)

```
O9082 ;
G00 X#24 Y#25 ;
G00 Z#100 ;
G01 Z#101 F103 ;
G04 P#102
G01 Z-[ROUND [#100] + ROUND [#101]] ;
M99 ;
```

and I write macro call as follows:

(P10)

```
G91 ;
G65 P9000 R-80. Z-40. U3.0 F250. ;
G65 P9082 X100. Y50. ;
G65 P9082 X··· Y··· ;
  .
  .
  .
```

T: Very good.

****

S: Wait a minute! This canned cycle always returns to the initial point.

T: I also overlooked it. A program would be useless if it is disabled for the designation of the initial point return (G98) and point-R return (G99). To solve this problem, use the system variable called "current value of modal information command" to know which state, G98 or G99, is provided, and change the specification of the tool return destination.

S: G98 and G99 belong to group "10." So, I have to use system variable #4010. Is it right?

T: Yes. Then, using "IF ··· GO TO ···" command, change the specification of the tool return destination.
There is one more point to be improved. Write the program so that the group "01" G codes before execution may be retained after the execution of this macro. Group "01" G codes include G00 through G03.

S: It looks difficult ···
What would you say to the following program?

T: Well done!

(P11)

```
Macro Call Program
G91 G99;
G65 P9000 R-80. Z-40. U3.0 F250;
G65 P9082 X100. Y50.;
G65 P9082 X...Y...;
  .
  .
  .
G98;
G65 P9082 X...Y...;
```

User Macro Body (P12)

```
O9000;
#100 = #18;
#101 = #26;
#102 = #21;
#103 = #9;
#104 = 0
----------------------------------------
O9082;
#104 = #104 + 1
#1 = #4001; ... G0 to G3
#2 = #4010; ... G98/G99
G00 X#24 Y#25;
IF[ #104 NE1] GO TO 1;
IF[ #2 EQ99] GO TO 2;
G00 Z#100;
N1 G01 Z#101 F#103;
G04 P#102;
IF[ #2 EQ 98] GO TO 2;
G00 7-[ #101]
GO TO 3;
N2 G00 Z-[ROUND [ #100] + ROUND [ #101]];
N3 G#1; ... Restore G Code
M99;
```