CNC 8060 8065

Examples manual (·M· model)

(Ref: 1402)

FAGOR

FAGOR AUTOMATION

MACHINE SAFETY

It is up to the machine manufacturer to make sure that the safety of the machine is enabled in order to prevent personal injury and damage to the CNC or to the products connected to it. On start-up and while validating CNC parameters, it checks the status of the following safety elements. If any of them is disabled, the CNC shows a warning message.

- Feedback alarm for analog axes.
- Software limits for analog and sercos linear axes.
- Following error monitoring for analog and sercos axes (except the spindle) both at the CNC and at the drives.
- Tendency test on analog axes.

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC resulting from any of the safety elements being disabled.

HARDWARE EXPANSIONS

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC resulting from any hardware manipulation by personnel unauthorized by Fagor Automation.

If the CNC hardware is modified by personnel unauthorized by Fagor Automation, it will no longer be under warranty.

COMPUTER VIRUSES

FAGOR AUTOMATION guarantees that the software installed contains no computer viruses. It is up to the user to keep the unit virus free in order to guarantee its proper operation.

Computer viruses at the CNC may cause it to malfunction. An antivirus software is highly recommended if the CNC is connected directly to another PC, it is part of a computer network or floppy disks or other computer media is used to transmit data.

FAGOR AUTOMATION shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC due a computer virus in the system.

If a computer virus is found in the system, the unit will no longer be under warranty.

The information described in this manual may be changed due to technical modifications. Fagor Automation reserves the right to make any changes to the contents of this manual without prior notice.

All the trade marks appearing in the manual belong to the corresponding owners. The use of these marks by third parties for their own purpose could violate the rights of the owners.

It is possible that CNC can execute more functions than those described in its associated documentation; however, Fagor Automation does not guarantee the validity of those applications. Therefore, except under the express permission from Fagor Automation, any CNC application that is not described in the documentation must be considered as "impossible". In any case, Fagor Automation shall not be held responsible for any personal injuries or physical damage caused or suffered by the CNC if it is used in any way other than as explained in the related documentation.

The content of this manual and its validity for the product described here has been verified. Even so, involuntary errors are possible, thus no absolute match is guaranteed. Anyway, the contents of the manual is periodically checked making and including the necessary corrections in a future edition. We appreciate your suggestions for improvement.

The examples described in this manual are for learning purposes. Before using them in industrial applications, they must be properly adapted making sure that the safety regulations are fully met.

INDEX

BASIC CONCEPTS.

1

Purpose of the exercises.

The purpose of the following programming examples is to familiarize with editing, simulating and executing programs. Machining starts with raw piece that after running various operations and cycles on it, it becomes the finished part, programming also the relevant machining conditions as well as the tools to be used.

The feedrate and spindle speed values are only illustrative and depend on the material of the part and the tool being used. When using the examples of this manual to make real parts (on a machine), the feedrate and spindle speed values must be adapted properly.

1.1 Basic CNC operating concepts

Some useful keys.

Some operating modes.

Key.	Function.
AUTO	Automatic mode. Execute a part-program in "single block" or "automatic" mode.
EDIT	EDISIMU mode. Edit and simulate the execution of a part-program displaying a graphic representation of the program being simulated.
TABLES	User tables (zero offsets, fixtures and arithmetic parameters).
TOOLS	Tool and magazine table.

Execution keys.

Key.	Function.
|	Cycle start key (START). Execute the selected program in automatic mode, a block in MDI/MDA mode, etc.
◯	Cycle stop key (STOP). Interrupt the execution of the CNC.
RESET	Reset key. It initializes the system setting the initial conditions as defined by machine parameters.
SINGLE	Single-block execution mode.

Editing a program.

Programs are edited in the EDISIMU mode. Once in this mode, the "Open program" softkey may be used to select the program to be edited which may be either a new one or an existing one. When selecting this option, the CNC shows a list of the available programs.

To select a program from the list:

1. Select the folder that contains the program. If it is a new program, it will be saved in this folder.
2. Select the program from the list or write its name in the bottom window. To edit a new program, write the name of the program in the lower window and the CNC will open an empty program or a predefined template depending on how the editor is configured.
3. Press [ENTER] to accept the selection and open the program or [ESC] to cancel it and close the program listing.

Syntax check.

The CNC analyzes each program block while editing. If the CNC detects a syntax error in the block, the error window at the bottom of the screen will display it.

The entire program can also be checked. To do that, press the vertical softkey for syntax check. The errors found will be indicated like with the other method.

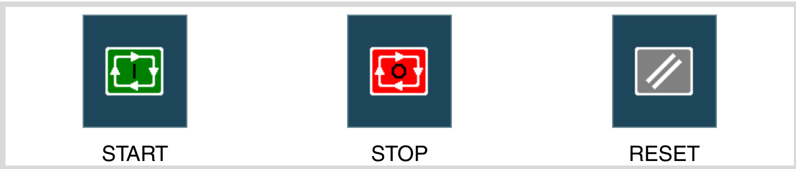
Program simulation.

The program simulation procedure is the following:

1 Choose the type of graphic representation, its dimensions and the point of view. This data may also be modified during the simulation of the program.

2 Activate the desired simulation options using the softkey menu.

3 Pressing the [START] softkey begins the simulation of the program that is being edited. The simulation may be interrupted with the [STOP] softkey or canceled with the [RESET] softkey.



The simulation of the program starts at the first block of the program and ends after executing one of the end-of-program functions "M02" or "M30". As an option, it is possible to define the first and last blocks of the simulation.

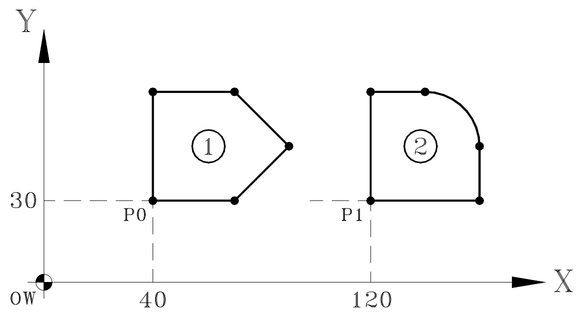
1.2 Set part zero.

With this CNC, it is possible to program movements in the machine reference system or apply offsets in order to use reference systems referred to the fixtures or the part without having to change the coordinates of the different points of the part in the program.

The part zero must be located so as to make programming easier. If no part zero is set, the coordinates will be referred to the machine reference system.

Coordinate preset (G92).

When presetting a coordinate, the CNC interprets that the axis coordinates programmed after the G92 set the current position of the axes. The rest of the axes that have not been defined with G92 are not affected by the preset.




```
N100 G90 G01 X40 Y30      (Positioning at P0)
N110 G92 X0 Y0            (Presetting P0 as part zero)
N200 G90 G01 X80 Y0       (Positioning at P1)
N210 G92 X0 Y0            (Presetting P1 as part zero)
```

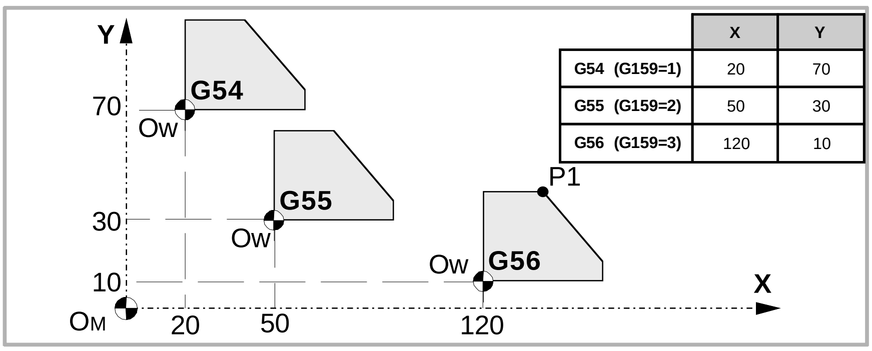
Zero offsets (G54-G59/G159).

Using zero offsets, it is possible to place the part zero in different positions of the machine. When applying a zero offset, the CNC assumes as the new part zero the point defined by the selected zero offset.

In order to apply a zero offset, it must have been previously defined. To do that, the CNC has a table where the operator may define up to 99 different zero offsets. The table data may be defined:

- Manually from the CNC's front panel (as described in the Operating Manual).
- By program, assigning the corresponding value (of the "n" offset and of the "Xn" axis) to the "V.A.ORGT[n].Xn" variable.

Once the zero offsets have been defined in the table, they may be activated via program by programming function G59 followed by the offset number to be activated. The first six zero offsets of the table can also be applied using functions G54 through G59; G54 for the first one (same as G159=1), G55 for the second one (same as G159=2) and so on.



	X	Y
G54 (G159=1)	20	70
G55 (G159=2)	50	30
G56 (G159=3)	120	10

```
N100 V.A.ORGT[1].X=20   V.A.ORGT[1].Y=70
N110 V.A.ORGT[2].X=50   V.A.ORGT[2].Y=30
N100 V.A.ORGT[3].X=120   V.A.ORGT[3].Y=10
...
N100 G54
    (It applies the first zero offset)
N200 G159=2
    (It applies the second zero offset)
N300 G56 X20 Y30
    (It applies the third zero offset)
    (The axes move to point X20 Y30 (point P1) referred to the third origin)
```

Cancellation of the part zero (G53).

The part zero stays active until it is canceled with a preset, a zero offset or with a "G53".

1.3 Programming of the machining conditions.

Feedrate programming units (G94/G95).

G94 Feedrate in millimeters/minute (inches/minute).

The feedrate is independent from the spindle speed.

G95 Feedrate in millimeters/revolution (inches/revolution).

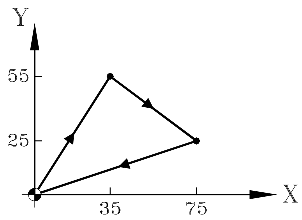
The feedrate changes with the spindle speed (usual operation on lathes).

By default the type of feedrate is set in machine parameter IFEED.

1.4 Programming coordinates.

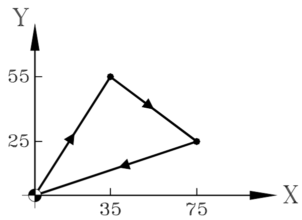
Absolute (G90) or incremental (G91) coordinates.

G90 Programming in absolute coordinates. The coordinates of the point are referred to the current origin of the coordinate system, usually the part zero.




```
N10 G00 G71 G90 X0 Y0
N20 G01 X35 Y55 F450
N30 X75 Y25
N40 X0 Y0
N50 M30
```

G91 Programming in incremental coordinates. The coordinates of the point are referred to the current tool position.




```
N10 G00 G71 G90 X0 Y0
N20 G01 G91 X35 Y55 F450
N30 X40 Y-30
N40 X-75 Y-25
N50 M30
```

By default the type of feedrate is set in machine parameter ISYSTEM.

1.5 Tool path programming

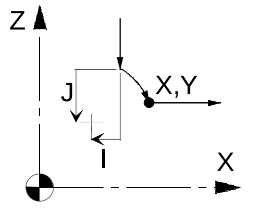
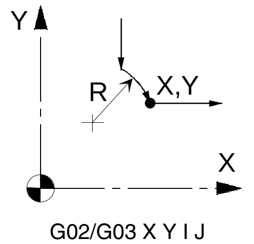
G00 Rapid traverse.

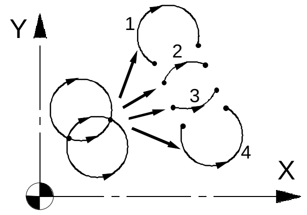
G01 Linear interpolation.

G02 Clockwise circular interpolation.

G03 Counterclockwise circular interpolation.

Functions G02/G03 offer two ways of programming in Cartesian coordinates.

Setting the end point and the radius.	Setting the end point and the center.
	

Sign of the radius.	
	Arc 1: G02 X... Z... R-... Arc 2: G02 X... Z... R+... Arc 3: G03 X... Z... R+... Arc 4: G03 X... Z... R-...

G36 Corner rounding, radius blend.

The programming format is "G36 I–" where "I" is the radius. Parameter I is valid for all four functions G36, G37, G38 and G39 and stays active until a new value is programmed.

G37 Tangential entry.

The programming format is "G37 I–" where "I" is the radius.

G38 Tangential exit.

The programming format is "G38 I–" where "I" is the radius.

G39 Corner chamfering.

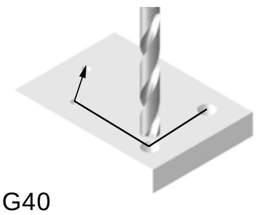
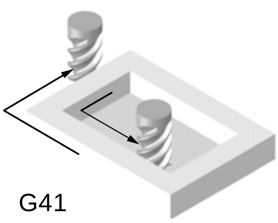
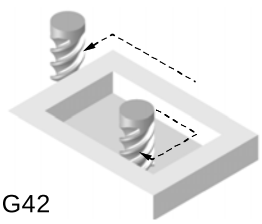
The programming format is "G39 I–" where "I" is the size of the chamfer.

G40 Cancellation of tool radius compensation.

G41 Left-hand tool radius compensation.

G42 Right-hand tool radius compensation.

The tool will position to the left or to the right of the programmed path, according to the machining direction.

Without compensation.	With compensation.	
 G40	 G41	 G42

ISO PROGRAMMING

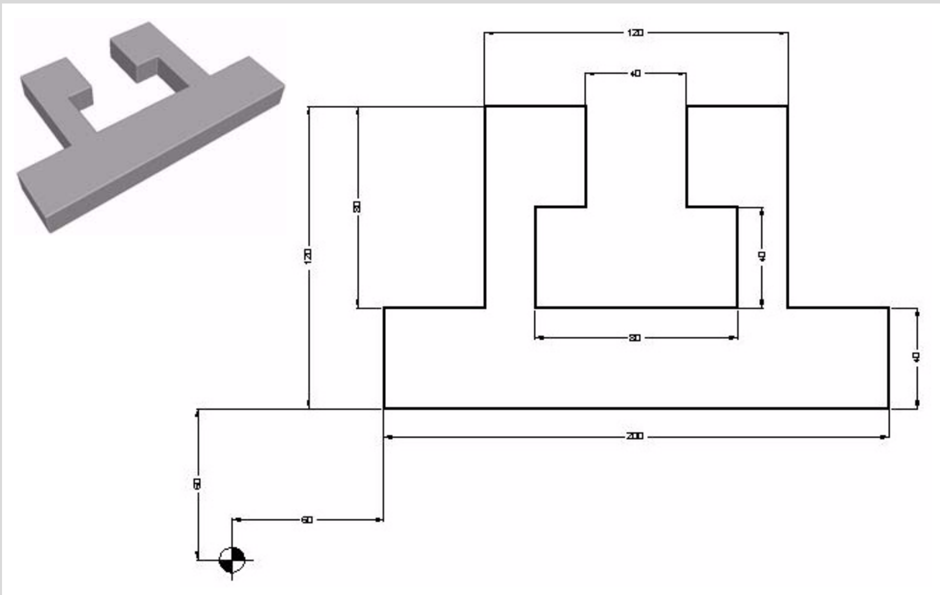
2

In general, ISO programming consists in entering a number of blocks that properly ordered make up a machining program. Basically, the programs are divided into three parts:

1. Header.
2. Geometry.
3. End.

2.1 Example. Programming of paths 1.

External contouring (climb cutting) with tangential input and a total depth of 20 mm with 5mm passes.



Operation.	Tool.
Contouring	Endmill Ø15 T10 D1

Header.

```
G0 Z100
    ; Safety positioning.
T10 D1
    ; Calling a tool and a tool offset.
M6
    ; Tool change
S1000 M3
X30 Y30
Z0
N1:
    ; Positioning of label Nr 1.
G91 G1 Z-5 F100
    ; Initial "Z" pass.
G90 G42 X60 Y60 F1000
    ; Tangential entry with tool radius compensation.
G37 I10
```

Geometry.

```
X260
Y100
X220
Y180
X180
Y140
X200
Y100
X120
Y140
X140
Y180
X100
Y100
X60
Y60
G38 I10
G40 X30 Y30
```

End.

```
N2:
    ; Positioning of label Nr 2.
#RPT[N1,N2,3]
    ; Repetitions.
G0 Z100
M30
    ; Return to the safety position and end of program.
```

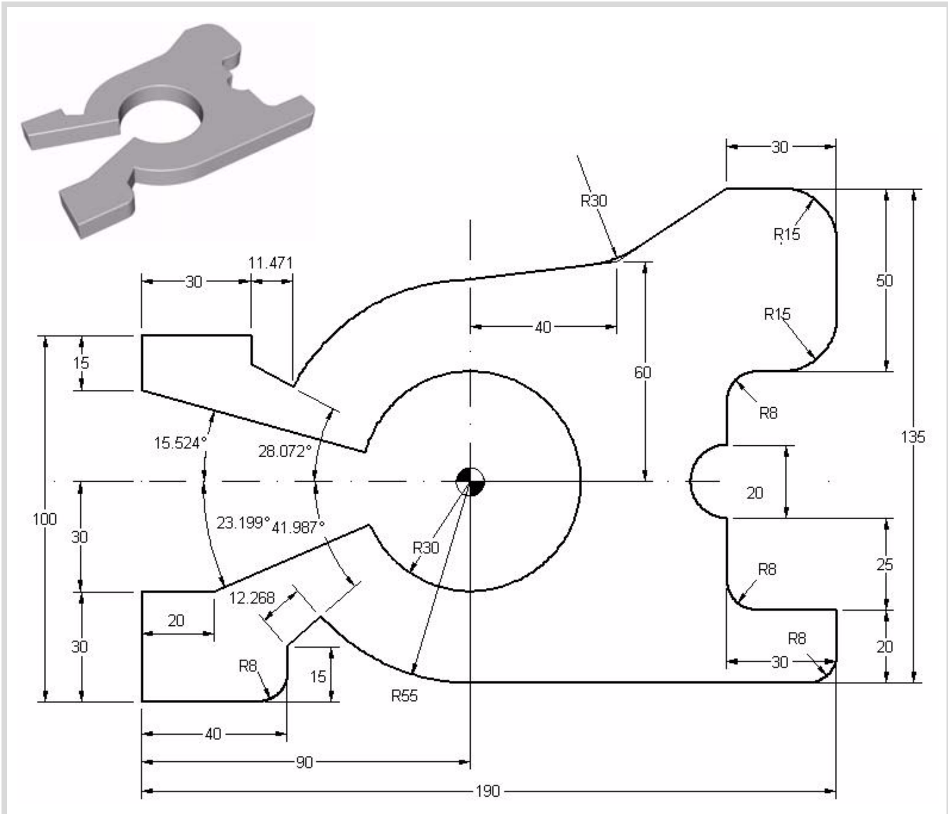
2.2 Example. Programming of paths 2.

In this exercise, you'll do a contour by entering polar coordinates because the data specifying the necessary points in X and Y is missing.

Programming a polar coordinate requires defining a center, a radius and angle (to do a straight line) or by just having an angle (arcs). This center is called Polar Center and is defined using function G30.

In this exercise, you will do an external contour of the geometry obtaining a total depth of 12 mm. Bear in mind that this geometry contains inside rounding with a radius of 8 mm and a tool with a larger diameter cannot be used.

Making this part requires the following steps:



Operation.	Tool.
Contouring	Endmill Ø8 T4 D1

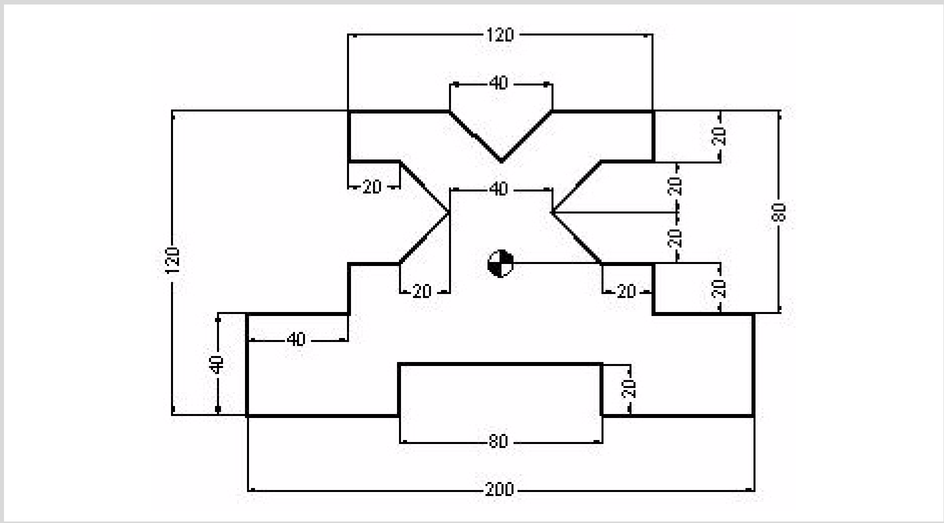
Doing the exercise

```
G0 Z100
    ; Safety positioning.
T4 D1
S1000 M3
X-30 Y-30
Z0
N1:
    ; Positioning of label Nr 1.
G91 G1 Z-2 F100
G90 G42 X0 Y0 F1000
G37 I10
X40
G36 I8
G1 Y15
G30 I90 J60
G1 R55 Q221.987 F1000
G3 Q270
G1 X190
G36 I8
G91 Y20
X -30
G36 I8
Y25
G2 X0 Y20 R10
G90 G1 Y90
G36 I8
X190
G36 I15
Y140
G36 I15
G91 X-30
G90 G1 X130 Y120
G36 I30
X90 Y115
G3 Q151.958
G1 R67.268 Q151.928
Y100
X0
Y85
G1 R30 Q164.476
G2 Q203.199
G1 X20 Y30
X0
Y0
G38 I10
X-30 Y-30
N2:
    ; Positioning of label Nr 2.
#RPT [N1,N2,5]
    ; Repetitions.
M30
```

Remarks

Exercise done in ISO code, using Polar coordinates for linear moves (G30 I J, G1 R Q) as well as for arcs (G30 I J, G2/3 Q).

2.3 Example. Programming of paths 3.



Header.

```
G0 Z100
T4 D1
M6
S1000 M3
X-130 Y-90
Z0
N1:
G1 G91 Z-5 F120
G90 G42 X-100 Y-60 F1000
```

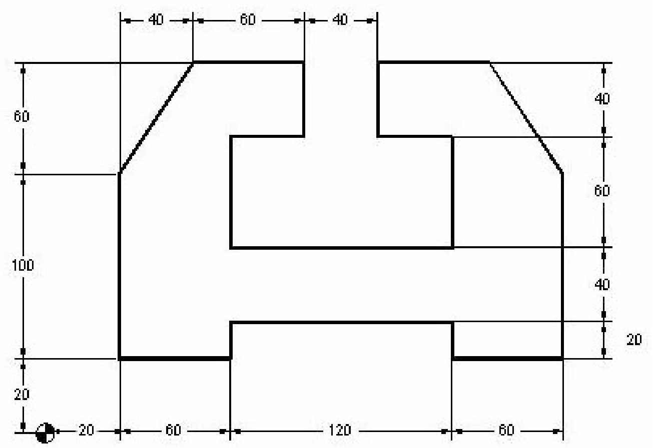
Geometry.

```
G37 I10
X -40
Y -40
X40
Y -60
X100
Y -20
X60
Y0
X40
X20 Y20
X40 Y40
X60
Y60
X20
X0 Y40
X-20 Y60
X -60
Y40
X -40
X-20 Y20
X-40 Y0
X -60
Y -20
X -100
Y -60
```

End.

```
G38 I10
G40 X-130 Y-90
N2:
#RPT [N1,N2,4]
G0 Z100
M30
```

2.4 Example. Programming of paths 4.



Header.

```
G0 Z100
T4 D1
M6
S1000 M3
X-10 Y-10
Z0
N1:
G1 G91 Z-5 F150
G90 G42 X20 Y20 F1000
G37 I10
```

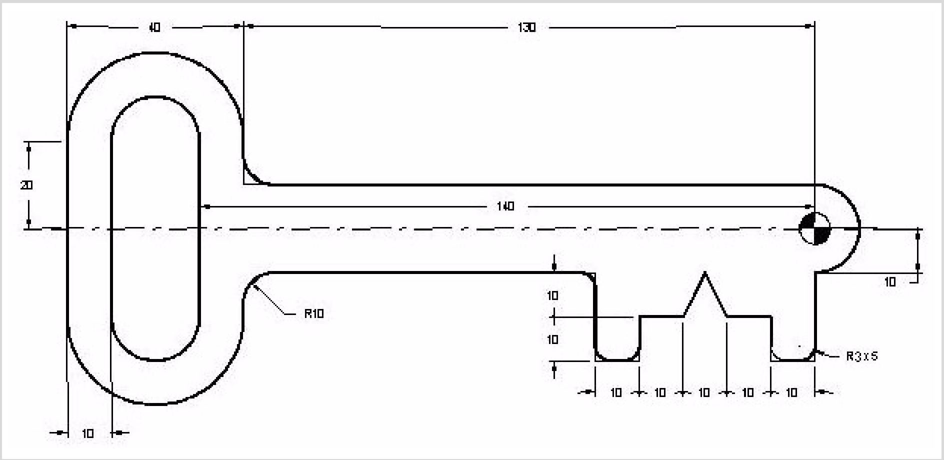
Geometry.

```
X80
Y40
X200
Y20
X260
Y120
X220 Y180
X160
Y140
X200
Y80
X80
Y140
X120
Y180
X60
X20 Y120
Y20
G38 I10
G40 X-10 Y-10
```

End.

```
N2:
#RPT [N1,N2,4]
G0 Z100
M30
```

2.5 Example. Programming of paths 5.



Header.

```
G0 Z100
T4 D1
M6
S1000 M3
X-90 Y-40
Z0
N1:
G1 G91 Z-5 F160
G90 G42 Y-10
G37 I10
```

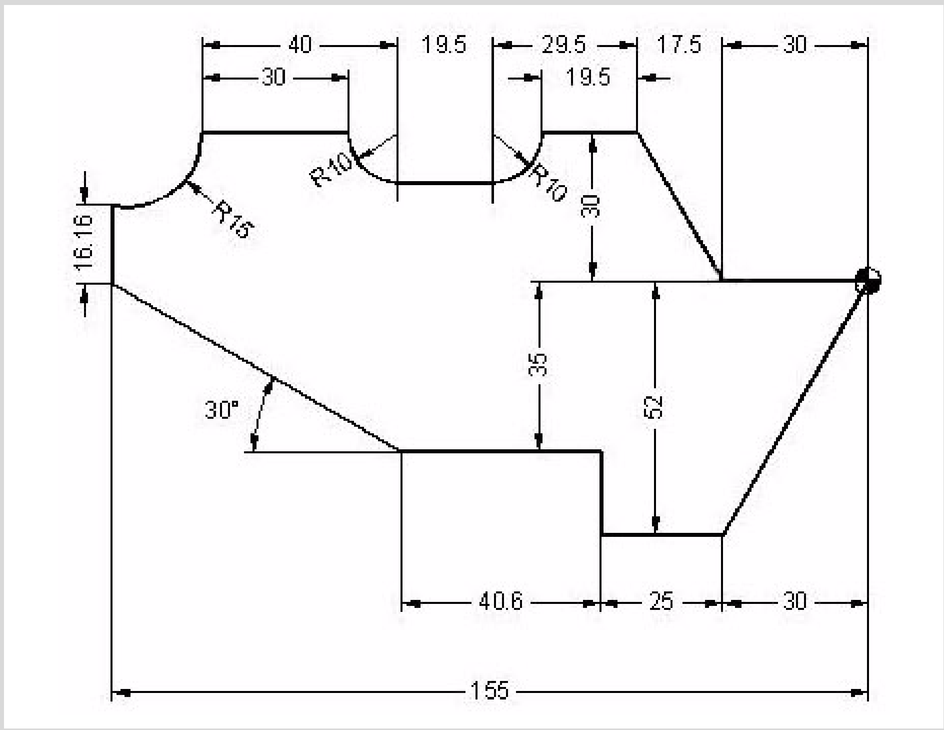
Geometry.

```
X -50
G36 I3
Y -30
G36 I3
X -40
G36 I3
Y -20
X -30
X-25 Y-10
X-20 Y-20
X -10
Y -30
G36 I3
X0
G36 I3
Y -10
G3 X0 Y10 R10
G1 X-130
G36 I10
Y20
G3 X-170 Y20 R20
G1 Y-20
G3 X-130 Y-20 R20
G1 Y-10
G36 I10
X -90
G38 R10
G40 Y-40
```

End.

```
N2:
#RPT[N1,N2,3]
G0 Z100
T2 D1
    ; Replace the current tool with another one whose Ø20 to do the slot.
M6
X-150 Y20
Z2
G1 Z0 F100
G91 Z-5
Y -40
Z-5
Y40
G0 G90 Z100
M30
```

2.6 Example. Programming of paths 6.



Header.

```
G0 Z100
T4 D1
M6
S1000 M3
X25 Y25
Z0
N1:
G1 G91 Z-5 F100
G90 G41 X0 Y0 F1000
G37 I10
```

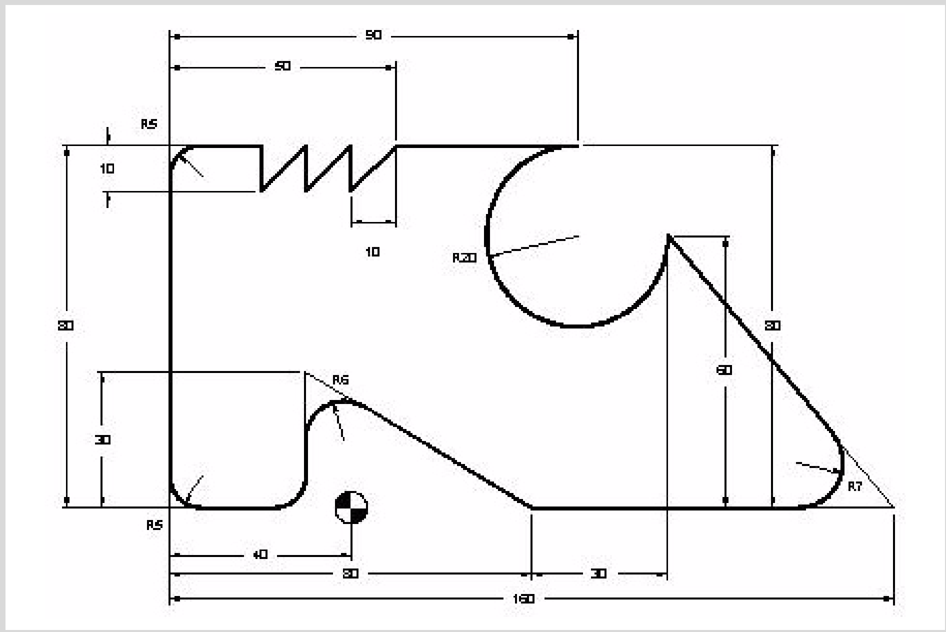
Geometry.

```
X-30 Y-52
X -55
Y -35
X -95.6
X-155 Y0
G91 Y16.16
G90 G3 X-136.5 Y30 R15
G91 G1 X30
G3 X10 Y-10 R10
G1 X19.5
G3 X10 Y10 R10
G1 X19.5
G90 X-30 Y0
X0 Y0
G38 I10
G40 X25 Y25
```

End.

```
N2:
#RPT [N1,N2,4]
G0 Z100
M30
```

2.7 Example. Programming of paths 7.



Header.

```
G0 Z100
T5 D1
M6
S1000 M3
X20 Y-30
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X40 Y0 F1000
G37 I10
```

Geometry.

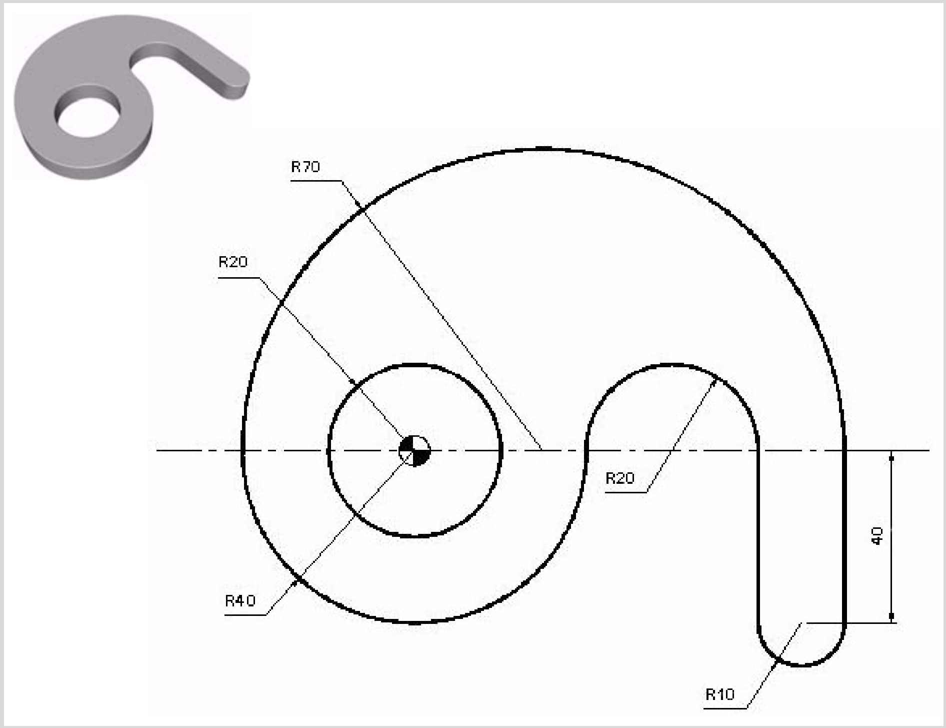
```
X120
G36 I7
X70 Y60
G2 X50 Y80 R-20
    ; The arc radius has a negative sign because it exceeds 180º.
G1 X10
X0 Y70
Y80
X-10 Y70
Y80
X-20 Y70
Y80
X -40
G36 I5
Y0
G36 I5
X -10
G36 I5
Y30
G36 I6
X40 Y0
G38 I10
G40 X20 Y-30
```

End

```
N2:
#RPT [N1,N2,4]
G0 Z100
M30
```

2.8 Example. Circular interpolation.

All circular interpolation exercises are based on the following figure.



2.8.1 Circular interpolation. G2/3 XY R

Exercise done using the format:

G2/3 X_ Y_ R_

XY End point.

R Radius of the arc.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G3 X40 Y0 R40
G2 X80 Y0 R20
G1 Y-40
G3 X100 Y-40 R10
G1 Y0
G3 X-40 Y0 R70
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.2 Circular interpolation. G2/3 XY IJ

Exercise done using the format:

G2/3 X_ Y_ I_ J_

XY End point.

IJ They define the arc center in incremental coordinates referred to the arc's starting point.

The arc center has been defined by incremental auxiliary coordinates.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G3 X40 Y0 I40 J0
G2 X80 Y0 I20 J0
G1 Y-40
G3 X100 Y-40 I10 J0
G1 Y0
G3 X-40 Y0 I-70 J0
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.3 Circular interpolation. G6 G2/3 XY IJ

Exercise done using the format:

G6 G2/3 X_ Y_ I_ J_

XY End point.

IJ Arc center referred to part zero, only if G6 is at the beginning of the block.

The arc center has been defined by absolute auxiliary coordinates.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G6 G3 X40 Y0 I0 J0
G6 G2 X80 Y0 I60 J0
G1 Y-40
G6 G3 X100 Y-40 I90 J-40
G1 Y0
G6 G3 X-40 Y0 I30 J0
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.4 Circular interpolation. G2/3 Q IJ

Exercise done using the format:

G2/3 Q_ I_ J_

Q Angle.

IJ Incremental distance from the arc's starting point to the arc center.

Using the Polar format with the center in incremental coordinates.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G31 G3 Q0 I40 J0
G31 G2 Q0 I20 J0
G1 Y-40
G31 G3 Q0 I10 J0
G1 Y0
G31 G3 Q180 I-70 J0
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.5 Circular interpolation. G6 G2/3 Q IJ

Exercise done using the format:

G6 G2/3 Q_ I_ J_

Q Angle.

IJ Arc center referred to part zero, only if G6 is at the beginning of the block.

Using Polar format and center definition in absolute coordinates.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G31 G6 G3 Q0 I0 J0
G31 G6 G2 Q0 I60 J0
G1 Y-40
G31 G6 G3 Q0 I90 J-40
G1 Y0
G31 G6 G3 Q180 I30 J0
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.6 Circular interpolation. G2/3 Q

Exercise defining the Polar center (G30) and then the movement to carry out.

G30 I J	Definition of the Polar center.
G2/3 Q	Interpolation with an angle.

IJ Absolute arc center coordinates referred to part zero. The Polar center is not affected by the incremental coordinates because the format itself is already absolute.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G30 I0 J0
G6 G3 Q0
G30 I60 J0
G6 G2 Q0
G1 Y-40
G30 I90 J-40
G6 G3 Q0
G1 Y0
G30 I30 J0
G6 G3 Q180
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.7 Circular interpolation. G8 XY

Exercise done using the format:

G8 X_ Y_

XY End point.

Function for an arc tangent to previous arc.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G3 X40 Y0 R40
G8 X80 Y0
G1 Y-40
G8 X100 Y-40
G1 Y0
G8 X-40 Y0
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.8 Circular interpolation. G9 XY IJ

Exercise done using the format:

G8 X_ Y_ I_ J_

XY End point.

IJ It defines any point of the arc.

Using the function for an arc defined by three points.

Part-program.

```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G37 I10
G9 X40 Y0 I0 J-40
G9 X80 Y0 I60 J20
G1 Y-40
G9 X100 Y-40 I90 J-50
G1 Y0
G9 X-40 Y0 I30 J70
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.8.9 Circular interpolation. G9 RQ IJ

Using the function for an arc defined by three points, in Polar.

G30 I J Definition of the Polar center. Using absolute auxiliary coordinates.

G9 R_ Q_ I_ J_

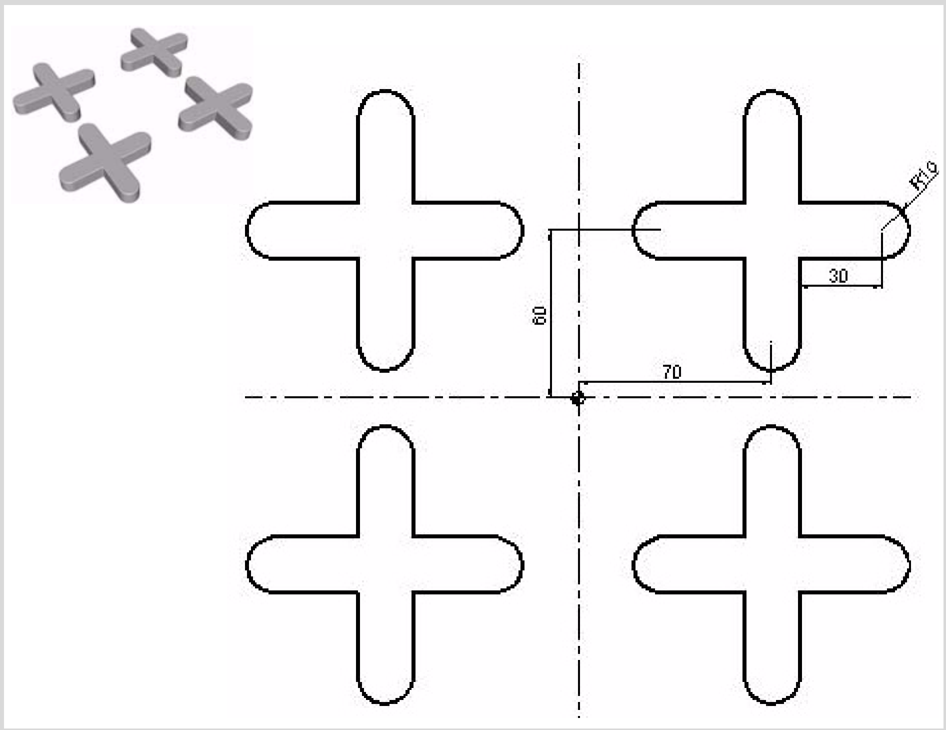
RQ Arc radius and angle referred to the Polar center.

IJ It defines any point of the arc.

Part-program.

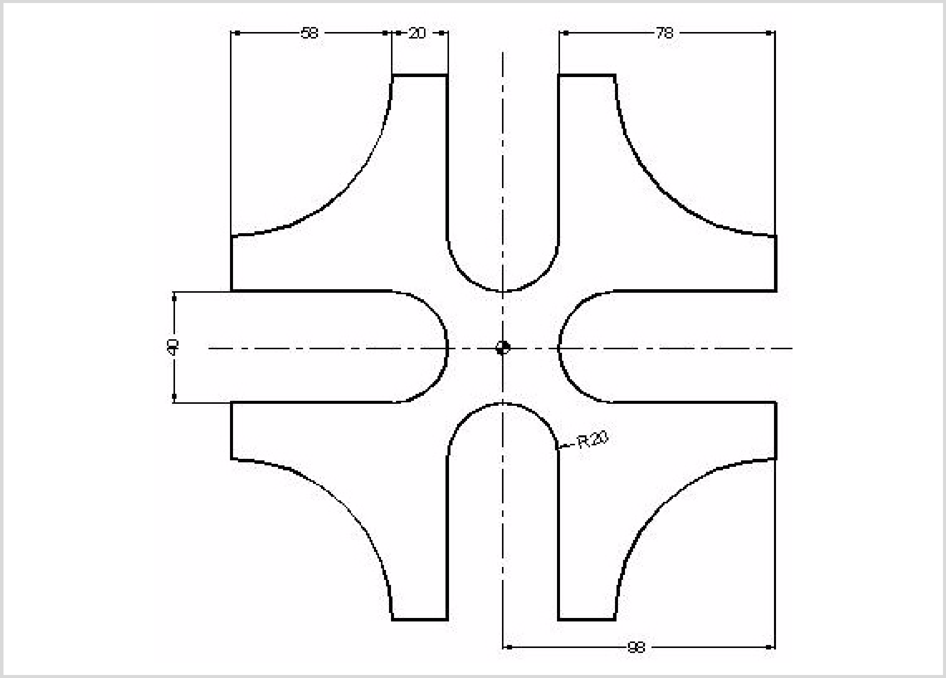
```
G0 Z100
T4 D1
M6
S1000 M3
X-70 Y0
Z0
N1:
G1 G91 Z-5 F100
G90 G42 X-40 Y0 F1000
G30 I0 J0
G37 I10
G9 R40 Q0 I0 J-40
G30 I60 J0
G9 R20 Q0 I60 J20
G1 Y-40
G30 I90 J-40
G9 R10 Q0 I90 J-50
G1 Y0
G30 I30 J0
G9 R70 Q180 I30 J70
G1 Z20
G1 X-20 Y0
G1 Z-20
G3 X-20 Y0 I20 J0
G1 Z20
G38 I10
G1 G40 X-70 Y0
G1 Z-20
N2:
#RPT[N1,N2,3]
G0 Z100
M30
```

2.9 Example. Mirror function.



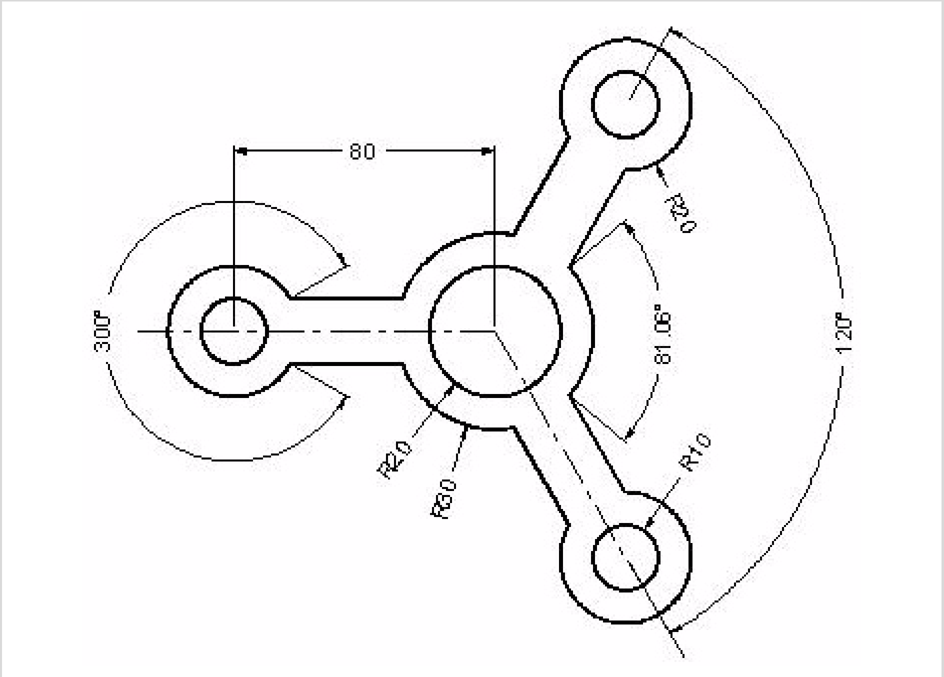

```
N1:
G0 Z100
T4 D1
M6
S1000 M3
X100 Y20
Z0
G1 Z-5 F100
G42 X100 Y50 F1000
 X110
G3 X110 Y70 R10
G1 X80
Y100
G3 X60 Y100 R10
G1 Y70
X30
G3 X30 Y50 R10
G1 X60
Y20
G3 X80 Y20 R10
G1 Y50
X100
G40 Y20
G0 Z100
N2:
G11
     ; Mirror function in X.
#RPT[N1,N2]
G10
G12
     ; Mirror function in Y.
#RPT[N1,N2]
G10
     ; Mirror function cancellation.
G11 G12
#RPT[N1,N2]
G10
M30
```

2.10 Example. Coordinate rotation.




```
G0 Z100
T4 D1
M6
S1000 M3
X120 Y0
Z0
N3:
G1 G91 Z-5 F100
G90 G42 X98 Y20 F1000
G37 I10
N1:
Y40
G2 X40 Y98 R58
G1 X20
Y40
G2 X-20 Y40 R20
G1 Y98
G73 Q90
    ; Coordinate rotation
N2:
#RPT[N1,N2,3]
G73
    ; Cancellation of coordinate rotation
G38 I10
 G40 X120 Y0
N4:
#RPT [N3,N4,5]
G0 Z100
M30
```

2.11 Example. Polar Coordinate rotation.

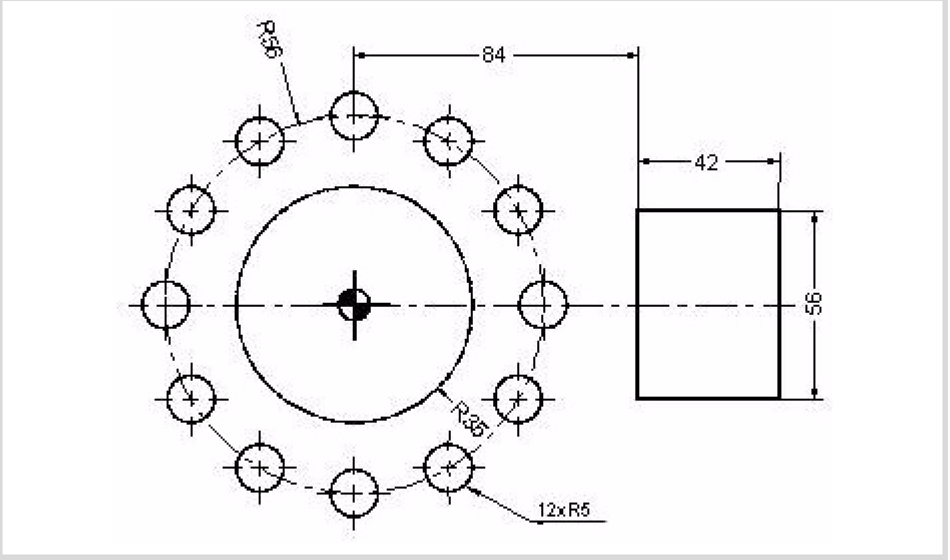



```
G0 Z100
T4 D1
M6
S1000 M3
R60 Q120
Z0
N3:
G1 G91 Z-5 F100
G90 G42 R30 Q120 F1000
G37 I10
N1:
G3 Q160.53
G30 I-80 J0
G1 R20 Q30
G3 Q-30
G30 I0 J0
G1 R30 Q-160.53
G3 Q-120
G73 Q120
N2:
#RPT[N1,N2,2]
G73
G38 I10
G30 I0 J0
 G40 G1 R60 Q120
N4:
#RPT [N3,N4,5]
G0 Z100
G99 X0 Y0
G88 Z2 I-30 D2 J20 B3
G0 G80 Z100
G99 R80 Q180
G88 Z2 I-30 D2 J10 B3
G91 Q120
G91 Q120
G90 G0 G80 Z100
M30
```

2.12 Example. Canned cycles.

Programming cycles always has the following sequence:

1 Prior positioning (starting plane).
2 Type of withdrawal (G98/G99) and XY position.
3 Cycle definition.
4 Cycle cancellation (G90) and withdrawal.




```
G0 Z100
T4 D1
M6
S1000 M3
G99 X0 Y0 F1000
G88 Z2 I-10 D2 J35 B3 L0.5 H500 V50
    ; Circular pocket canned cycle.
G0 G80 Z100
X105 Y0
G87 Z2 I-10 D2 J21 K28 B3 L1 H480 V30
    ; Rectangular pocket canned cycle.
G0 G80 Z100
T11 D1
M6
X0 Y56 G81 Z2 I-10
    ; Direct drilling.
N1:
G91 Q30
    ; Angular increment.
N2:
#RPT[N1,N2,10]
    ; Angular repetition.
G90 G0 G80 Z100
M30
```

2.13 Example. Canned cycle (G81) and multiple positioning (G162).

Any cycle, once defined, may be repeated in several ways using multiple machining.

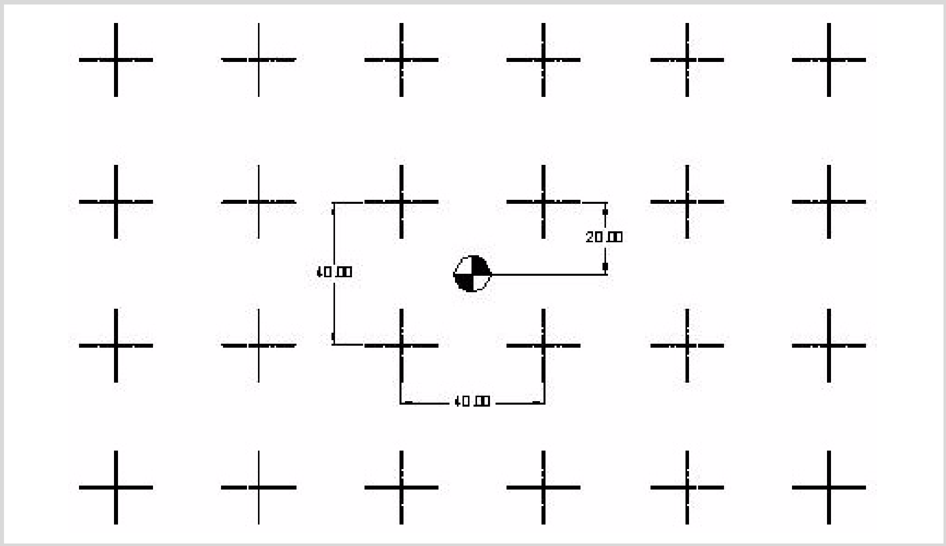
G160 - Multiple positioning in a straight line.

G161 - Multiple positioning in a parallelogram pattern.

G162 - Multiple positioning in a grid pattern.

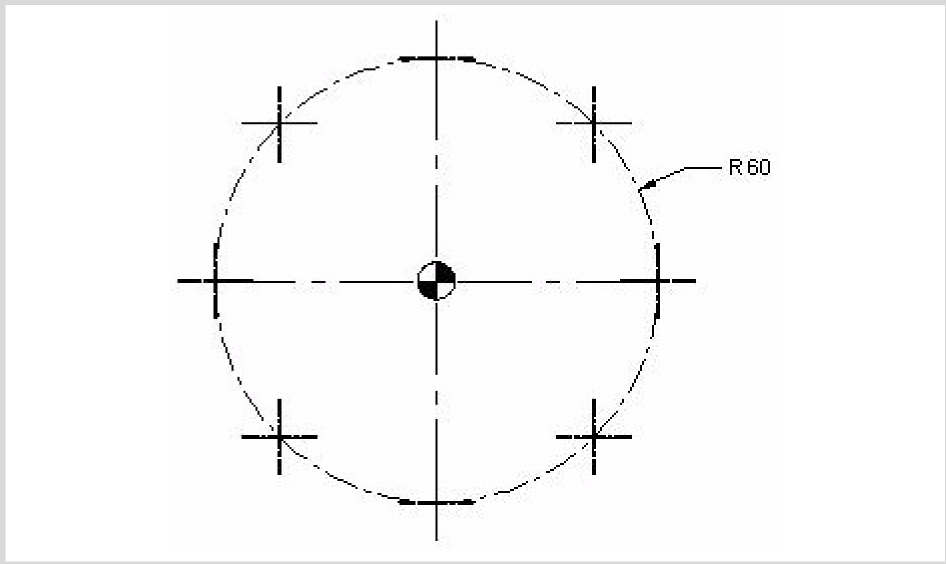
G163 - Multiple positioning in a circular pattern.

G165 - Multiple positioning using an arc chord.



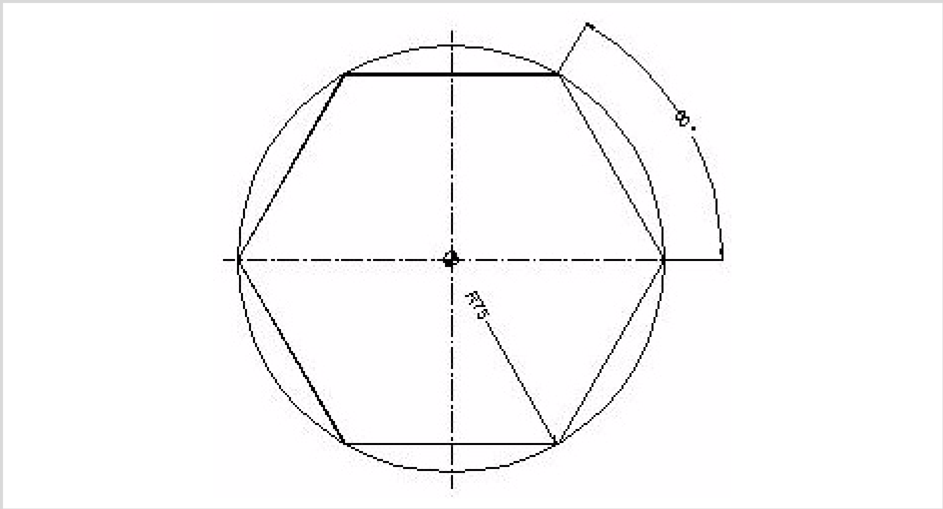

```
G0 Z100
T6 D1
M6
S1000 M3
G99 X-100 Y60 F1000
     ; Coordinate of the first drilling point (hole).
G81 Z2 I-10
G162 I40 K6 J-40 D4
     ; Multiple machining in a grid pattern.
G0 G80 Z100
M30
```

2.14 Example. Canned cycle (G81) and multiple positioning (G163).




```
G0 Z100
T6 D1
M6
S1000 M3
G99 X-42.4264 Y-42.4264 F1000
     ; Coordinate of the first drilling point (hole).
G81 Z2 I-10
G163 X42.4264 Y42.4264 I45
G0 G80 Z100
M30
```

2.15 Example. Angular repetition.



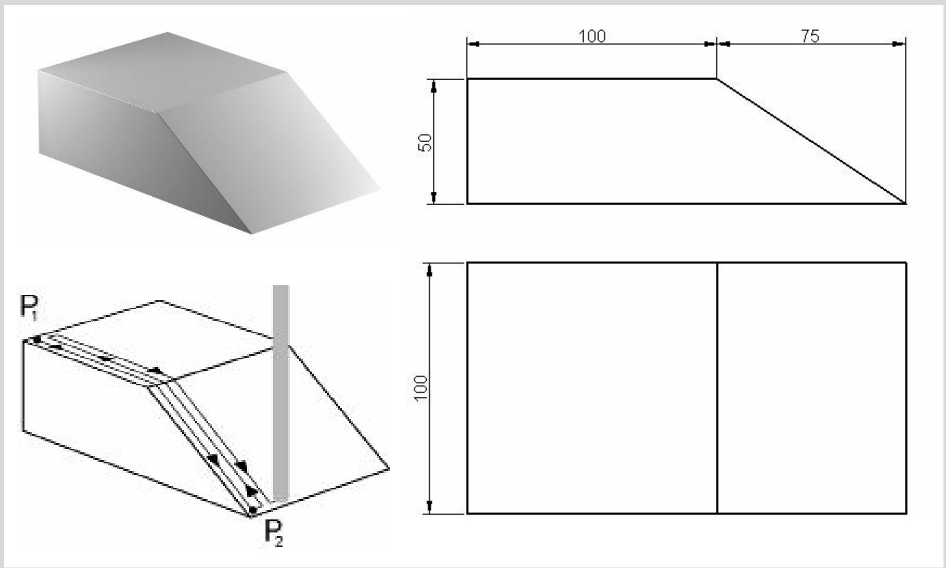

```
G0 Z100
T4 D1
M6
S1000 M3
X100 Y0
Z2
G1 Z0 F175
N1: G91 Z-5
G90 G42 X75 Y0
    ; Repetition of down movements.
N3: G91 Q60
    ; Polar programming of the first side.
N4:
#RPT [N3,N4,5]
    ; Angular repetition of the sides.
G90 G40 X100 Y0
N2:
#RPT [N1,N2,4]
    ; Repetition of down movements.
G0 Z100
M30
```

PARAMETRIC PROGRAMMING.

3

3.1 Exercise. Wedge.

Program a wedge by assigning parameters. Then, by using positioning and increments, do a comparison between the initial point and the final point to be reached. Parametric programming is handy when trying to change the assignment of parameters to obtain the desired dimensions using the same program.



Doing the exercise.

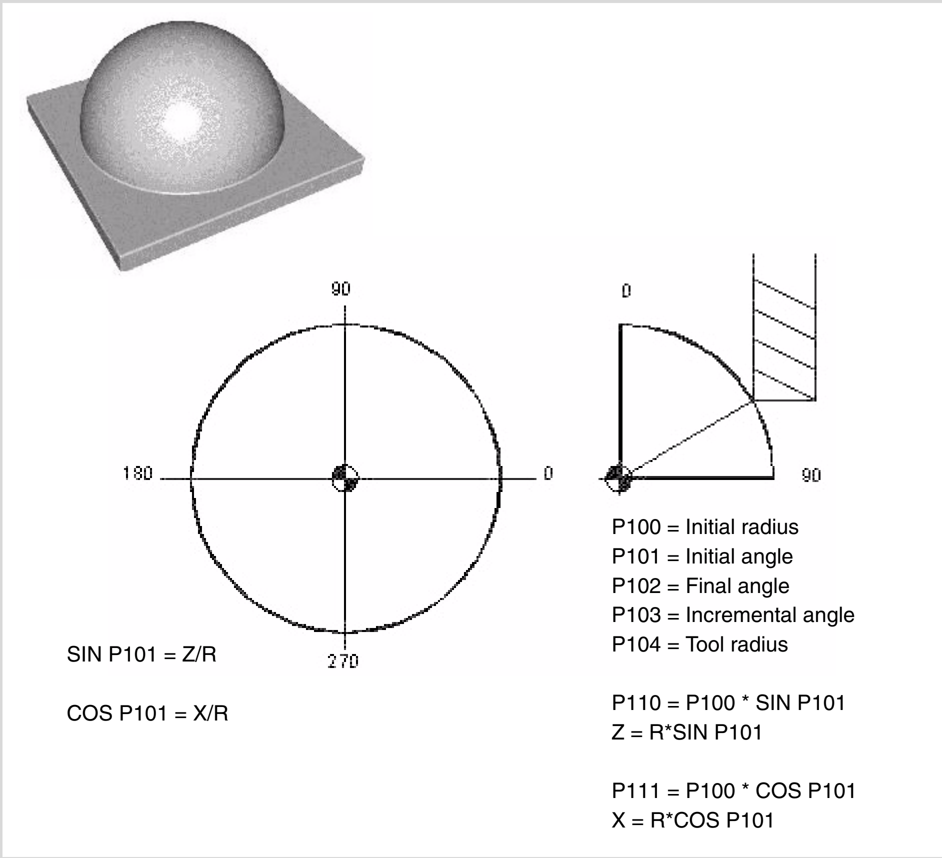
```
P100 = 100   ; Length in X.
P100 = 100   ; Length in Y.
P100 = 75   ; Movement in X.
P103 = 50   ; Depth.
P106 = 2
G0 Z100   ; Z position
T4 D1
M6   ; Calling a tool
S1000 M3
Y0
N1:   ; Label number 1
X0
Z0
G1 XP100 F1000   ; Initial X position
G1 G91 XP102 Z-P103
G90 YP106
G1 Z10
N2:
P106 = P106+2
$IF P106<P101 $GOTO N1   ; Comparison.
          ; If P106 is smaller than P101, the tool returns to label 1.
#RPT[N1,N2]   ; Repetition. Last pass
G0 Z100
M30
```

FAGOR

CNC 8060
CNC 8065

(REF: 1402)

3.2 Exercise. Semi-sphere.



Parameter assignment.

```
P100 = 60  ; Radius of the semi-sphere.
P101 = 90  ; Initial angle.
P102 = 0  ; Final angle.
P103 = 0.5  ; Incremental angle.
P104 = 8  ; Tool radius.
```

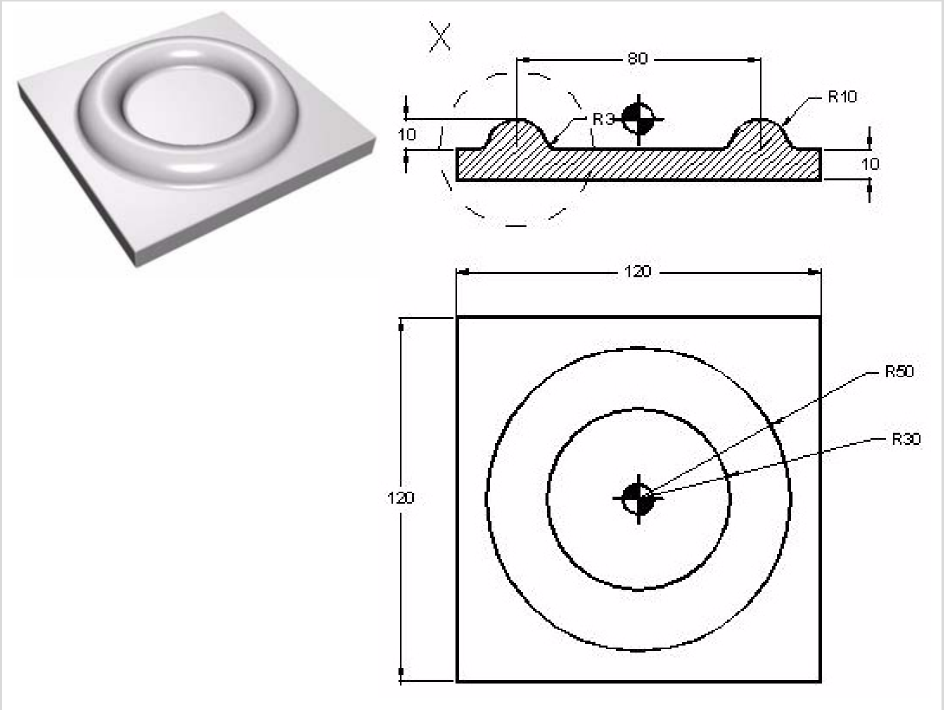
Program.

```
G0 Z100
T12 D1
M6
S1000 M3
X0 Y0
N1: P120 = P100*COS [P101] P121 = P100*SIN [P101]  ; XZ position.
P120 = P120+P104  ; Tool compensation.
P121 = P121-P100  ; Zero up.
G1 XP120 ZP121 F1000
G2 Q360
N2:
P101 = P101-P103  ; Angular decrement.
```

Comparison.

```
$IF P101 > P102 $GOTO N1
P101 = P102
#RPT[N1,N2]
G0 Z100
M30
```

3.3 Exercise. Toroid (donut).



Parameter assignment.

```
P100 = -90
P101 = 90
P102 = 1
P103 = 10
P104 = 3
P105 = -P103
P106 = 40
P120 = P103+P104
```

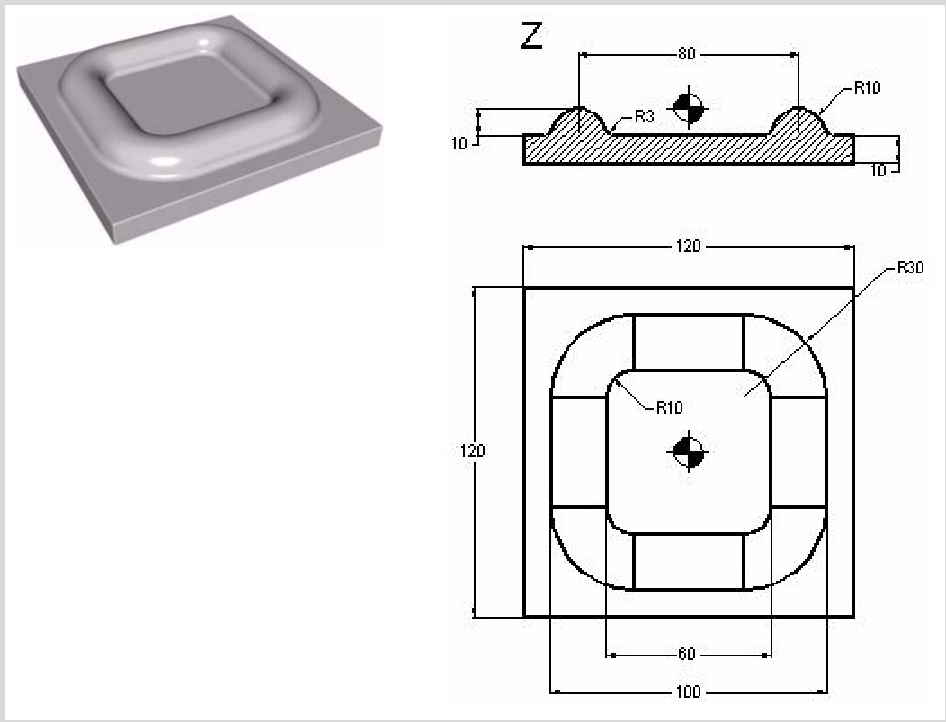
Program.

```
G0 Z100
T12 D1
M6
S1000 M3
X0 Y0
N1: G18
G30 IP105 JP106
G1 RP120 QP100 F1000
G17
G30 I0 J0
G3 Q360
N2:
P100 = P100+P102
```

Comparison.

```
$IF P100<P101 $GOTO N1
P100 = P101
#RPT[N1,N2]
G0 Z100
M30
```

3.4 Exercise. Ashtray.



Parameter assignment.

```
P100 = -90
P101 = 90
P102 = 1
P103 = 10
P104 = 3
P105 = -P103
P106 = 40
P120 = P103+P104
```

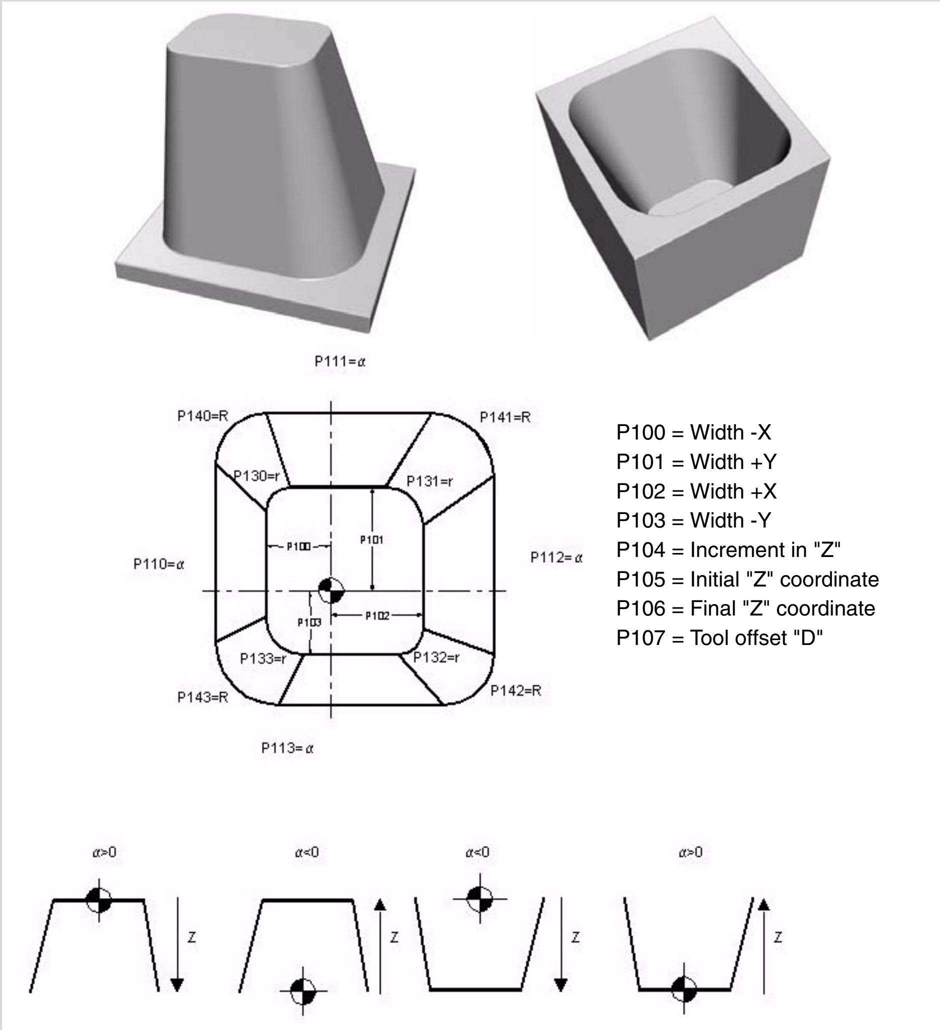
Program.

```
G0 Z100
T12 D1
M6
S1000 M3
X0 Y0
N1: G18
G30 IP105 JP106
G1 RP120 QP100 F1000
G17
G1 Y20
G31 G6 G3 Q90 I20 J20
G1 X-20
G31 G6 G3 Q180 I-20 J20
G1 Y-20
G31 G6 G3 Q-90 I-20 J-20
G1 X20
G31 G6 G3 Q0 I20 J-20
G1 Y0
N2:
P100 = P100+P102
```

Comparison.

```
$IF P100<P101 $GOTO N1
P100 = P101
#RPT[N1,N2]
G0 Z100
M30
```

3.5 Exercise. Pockets with 4 sides and 4 different radii.



Parameter assignment.

```
P102 = 50 P103 = 40   ; External sides.
P107 = 5  ; Tool radius.
P125 = 80 P126 = 60 P127 = 50 P128 = 70  ; Angles.
P130 = 5 P131 = 7 P132 = 4 P133 = 8  ; smaller radii.
P140 = 10 P141 = 12 P142 = 15 P143 = 17  ; Larger radii.
P120 = 0 P121 = 1 P122 = 30
P150 = P122-P120 P151 = P150/P121 P152 = FUP[P151]
P160 = P140-P130 P161 = P141-P131 P162 = P142-P132 P163 = P143-P133
P140 = P140+P107 P141 = P141+P107 P142 = P142+P107 P143 = P143+P107
P164 = P160/P152 P165 = P161/P152 P166 = P162/P152 P167 = P163/P152
G0 Z100
T4 D1
M6
N1:
P170 = P120/TAN[P125]
P171 = P120/TAN[P126]
P172 = P120/TAN[P127]
P173 = P120/TAN[P128]
P180 = P100-P170
P181 = P101-P171
P182 = P102-P172
P183 = P103-P173
```

Program.

```
G01 X-P180 Y0 Z-P120 F2000
YP181
G36 IP140
XP182
G36 IP141
Y-P183
G36 IP142
X-P180
G36 IP143
Y0
N2:
P120 = P120+P121
P140 = P140-P164 P141 = P141-P165 P142 = P142-P166 P143 = P143-P167
```

Comparison.

```
$IF P120<P122 $GOTO N1
P120 = P122
P140 = P130+P107 P141 = P131+P107 P142 = P132+P107 P143 = P133+P107
#RPT[N1,N2]
G00 Z50
M30
```