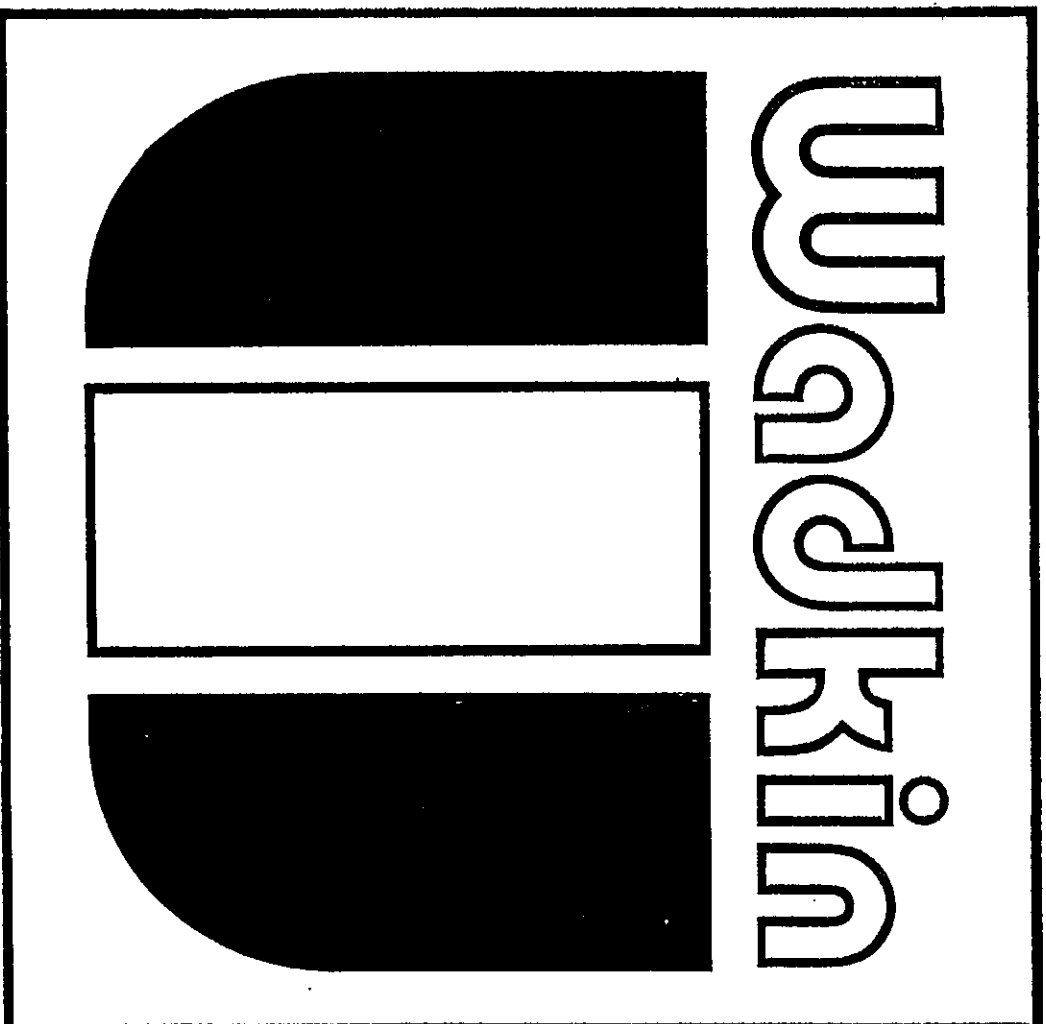


PROGRAMMING MANUAL

FOR BOSCH ALPHA 3 CONTROL

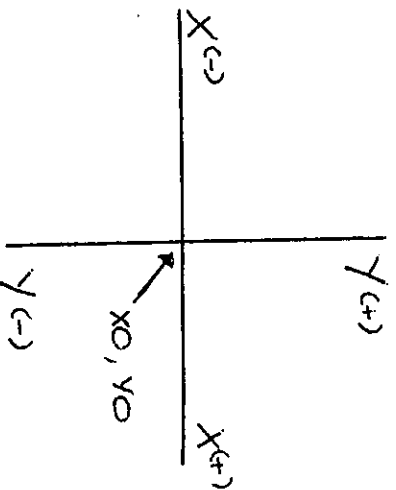


WADKIN COLNE, DIVISION OF WADKIN PLC  
LODGE HOLME, TRAWDEN, COLNE, LANCS BB8 8RB

A MEMBER OF THE THOMAS ROBINSON GROUP PLC.  
TEL:0282 866717 TELEX:635032 FAX:0282 862541

## CNC PROGRAMMING FOR THE BOSCH ALPHA CONTROL

**X and Y axis**



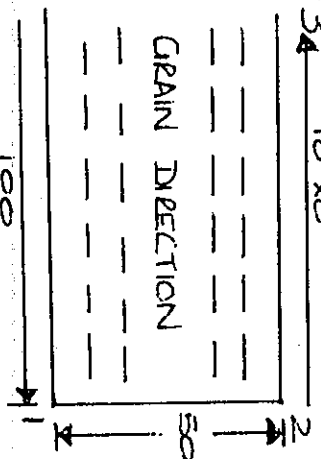
Notes:-

- (1) G90 - Absolute programming, the co-ordinates of a data point are defined in relation to an absolute zero.
- (2) G91 - Incremental programming, the co-ordinates of a data point are defined in relation to the preceding point.

Which ever type of programming is used needs to be stated at the beginning of the program. G90 and G91 cancel each other out.

- (2) The control generates its own block numbers, ie. when you input a line of program it will be issued with the respective block number, ie. N1, N2 ... N13 etc.
- (3) The first line of a program using the Alpha control must be G78 (start of main program), whatever block this is on needs to be noted, for later on in the program.  
All programs are entered consecutively, so it is one continuous run, it is therefore important to note what line each program starts on, for running the correct one.
- (3) The last line of a program will read M30 (end of main program, with jump address), ADDR=A (address of first line of program, G78's block number), this will jump the program back to the start each time it is run.
- (4) When actually cutting you need to tell the control how fast to travel. This is done by entering a feedspeed on the first G1, G2 or G3 line, once it is in it will remain active for the rest of the program, unless otherwise altered. Feedspeeds can be altered on any line of the program, if you want to tell the control to slow down, say if you were cutting across the grain, or cutting a curve the new feedspeed would be programmed on the same line as the information to cut.

eg. 34 → to x0



(1)	G1	X100	Y0	F4000
(2)	G1	X100	Y50	F500
(3)	G1	X0	Y50	F4000

When working in METRIC the feed is in units of 1 millimetre/minute  
(maximum programmable is 9999 mm/min, or virtually 10 metres/minute)

When working in INCHES the feed is in units of 1/10 inch/minute (maximum  
programmable is 4000 1/10 inch/min (400 inches/minute))

(5) When actually running program, if the cutter is in 'mid-air'  
(ie. not cutting) then the machine needs to go as fast as  
possible to minimise cycle time, (for instance moving to  
component datum to set zero). The code for this rapid  
traverse is G0, so the move to component datum could be  
G0 X700 Y300, as opposed to G1 X700 Y300.

(6) Entering a line of Program:--

G - 1, 2, 3 etc.  
X - co-ordinate.  
Y - co-ordinate.  
Z - for radius value on curve,  
for Z axis movement.  
F - feedspeed  
S - spindle speed  
M - 'M' modes, and end of program with jump address, ADDR=A.  
B - repetition factor (for use with G22, G23)  
C - sub-program number (G98)  
H - dwell in 0.1 seconds (G4)  
T - tool and radius compensation (G41, 42 and 43)

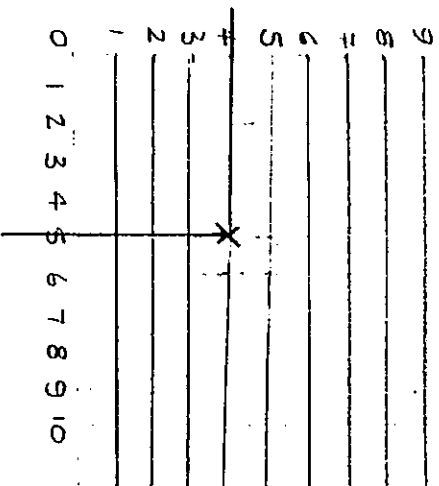
(7) When machine is switched on the control displays two values:--

eg. 6132	8192
amount of memory still available	total amount of memory available

(8) On switching on the memory is protected.

Mode + 9 (Manual Data Input)  
then G37 - Memory enable  
G38 - Memory protect  
G39 - Memory clear, ? enter for Yes.

WHAT IS A POINT?

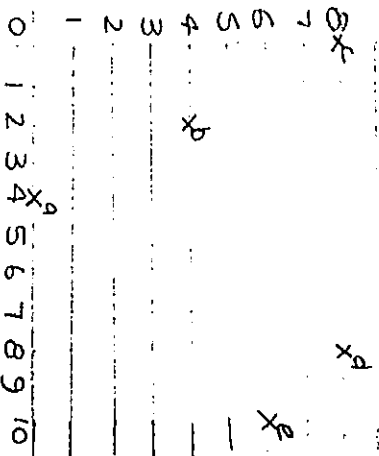


A line which is 5 units along the 'x' axis is known as X5, similarly a line which is 4 up in the 'y' axis is known as Y4.

Where these two lines intersect is known as a point or co-ordinate.

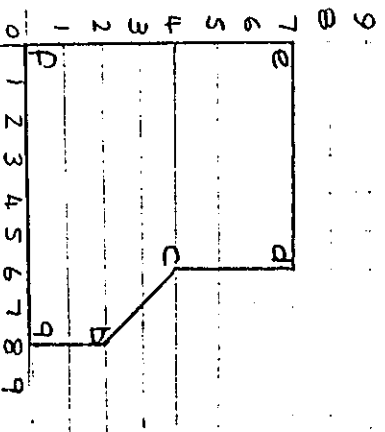
So the point described is known as X5 Y4.

Describe the points below:--



- (a)
- (b)
- (c)
- (d)
- (e)

POINTS FOR A SHAPE

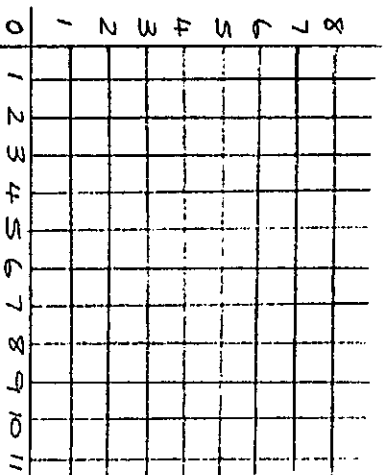


If we draw a shape or component on the graph, all the intersection points of it will lie on points of the graph, thus

- X0 Y0
- (a) X8 Y0
- (b) X8 Y2
- (c) X6 Y4
- (d) X6 Y7
- (e) X0 Y7
- (f) X0 Y0

On the graph below plot the points and draw lines between the points to describe the shape.

Start from X0 Y0  
 X6 Y0  
 X8 Y3  
 X8 Y7  
 X2 Y7  
 X0 Y3  
 X0 Y0



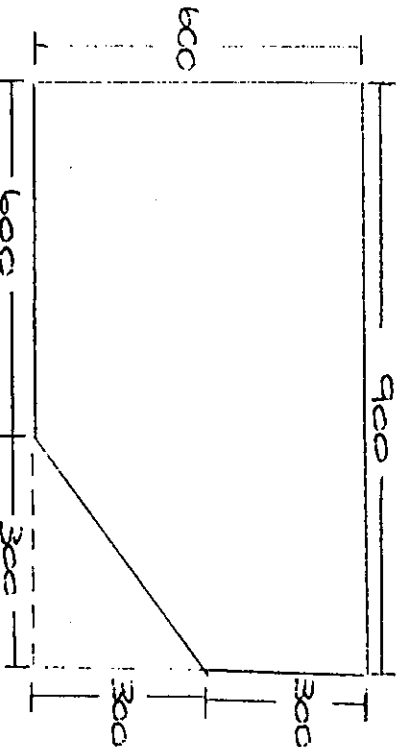
This is the basis of the CNC PROGRAMMING CO-ORDINATE SYSTEM. The table of the router is measured as a grid just like the graph but with a much finer spacing (resolution).

The resolution of the Wadkin Router is .01 mm or 1/1000 inch depending which units you are using.

### STRAIGHT LINES AND CURVES

#### i) Straight Lines

All straight line moves need to be preceded by a G1 code.

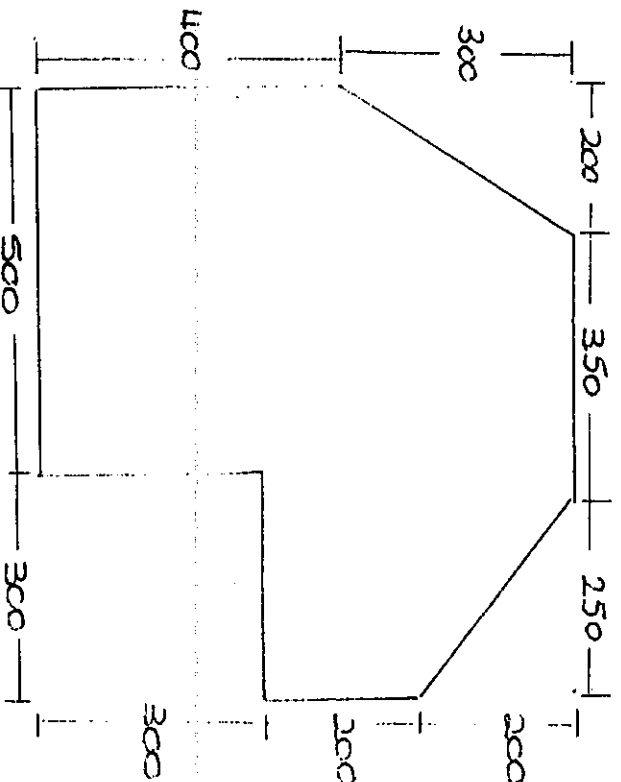


To describe this component shape (with the graph taken off) with all straight line moves would be as follows:-

```

G1 X0 Y0
G1 X600 Y0
G1 X900 Y300
G1 X900 Y600
G1 X0 Y600
G1 X0 Y0
  
```

Write down the points to describe the shape below:-



## 11) Curves

All curves need to be preceded by a G2 or G3 code depending whether the circular movement is clockwise or anti-clockwise.

G2 = Clockwise circular movement

G3 = Anti-clockwise circular movement

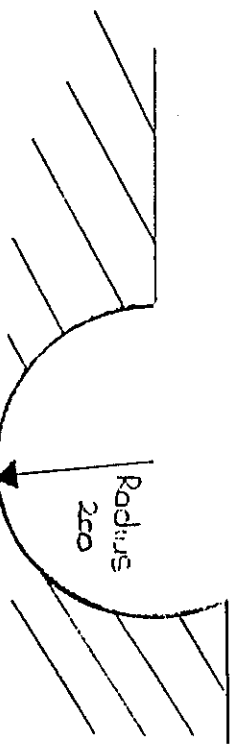
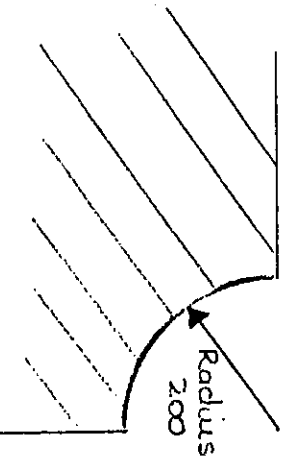
The radius is also required, this is programmed as a + or - Z value. The control automatically assumes that the value is positive unless otherwise stated, ie only the negative (-) sign is required.

Rule for radius value:-

If the angular movement of the curve is 180 degrees or more then the radius is programmed as a positive value, eg. Z200.

If the angular movement of the curve is less than 180 degrees then the radius is programmed as a negative value, eg. Z-150.

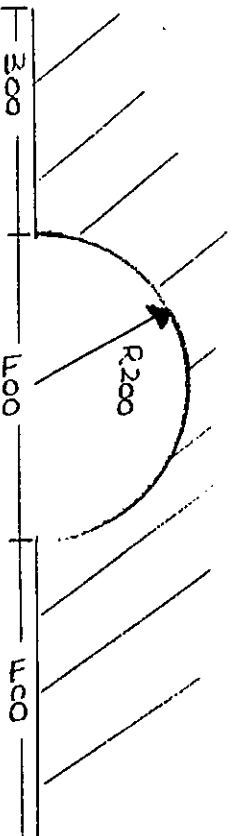
EG. (1) Less than  $180^\circ$  Z-200      (2) Greater than  $180^\circ$  Z200



The X and Y co-ordinates are the end point of the arc.

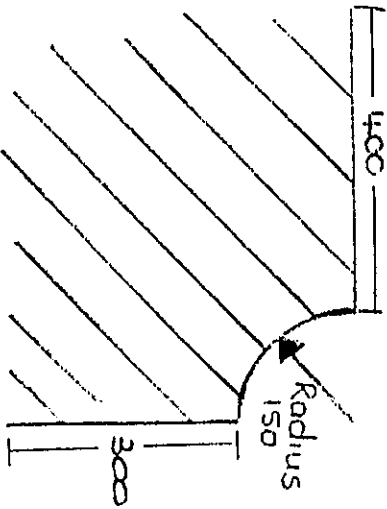
Example 1

```
G1 X300 Y0
G2 X700 Y0 Z200
G1 X1100 Y0
```



G2: Clockwise movement  
X700 Y0: End point of the arc  
Z200: Radius of 180 degrees, so positive value

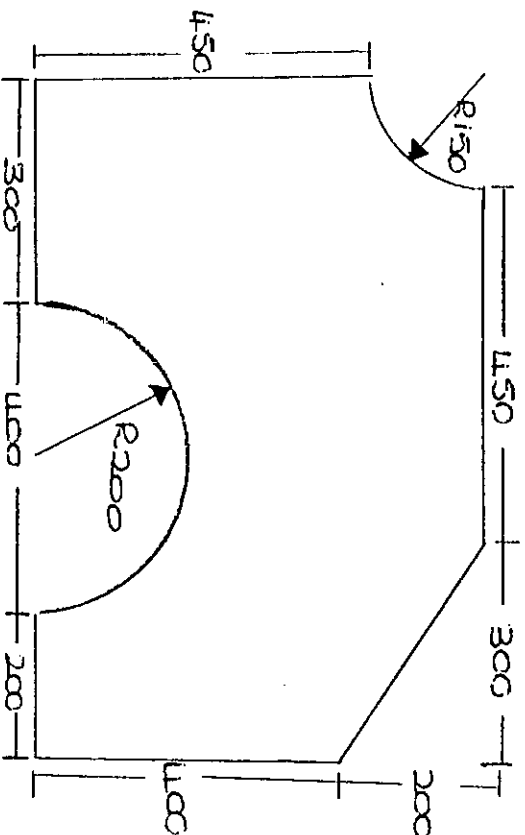
Example 2



```
G1 X400 Y0
G3 X550 Y-150 Z-150
G1 X550 Y-450
```

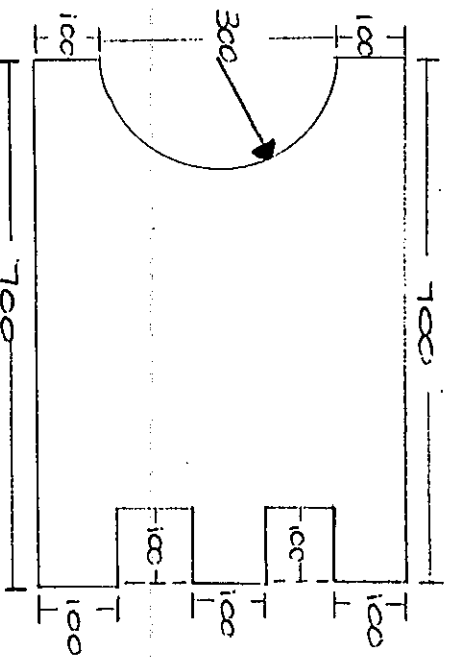
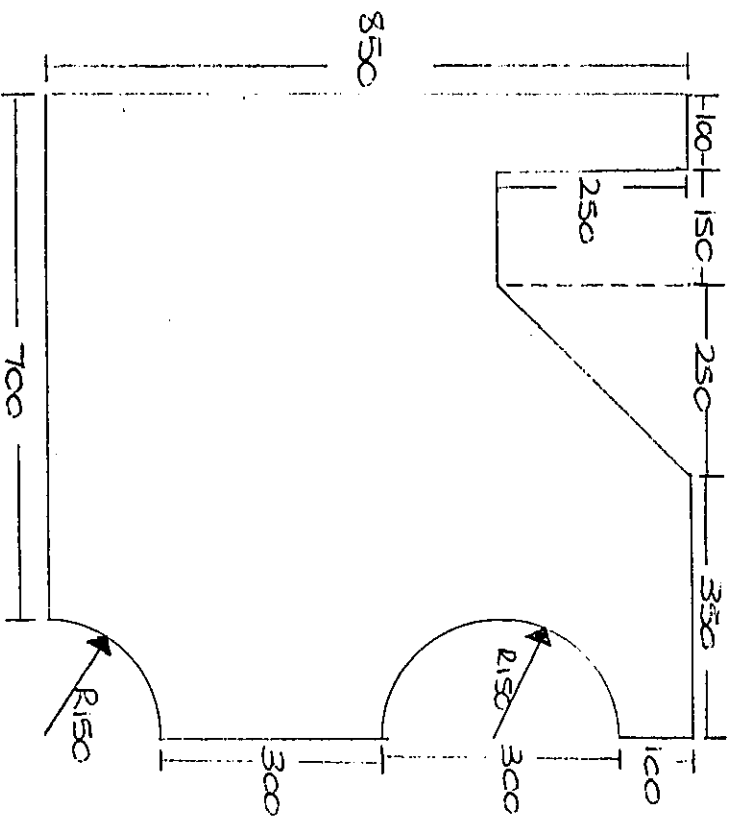
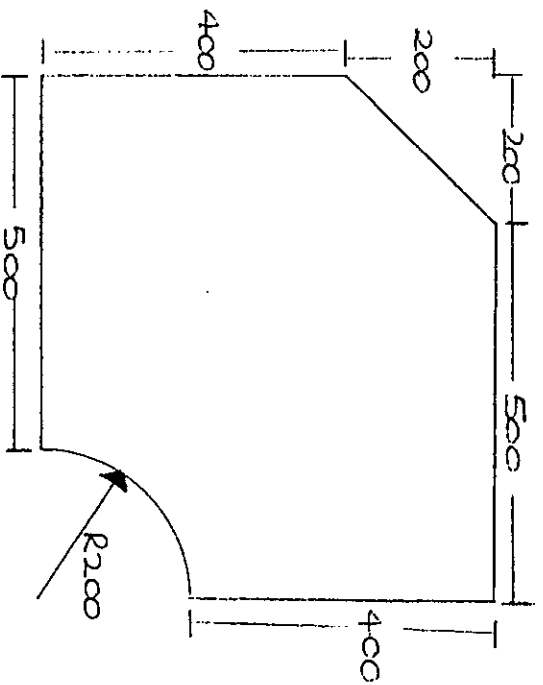
G3: Anti-clockwise movement  
 X550 Y-150: End point of arc  
 Z-150: Radius of less than 180°, so a negative value

Putting these 'G' codes into a complete example:-



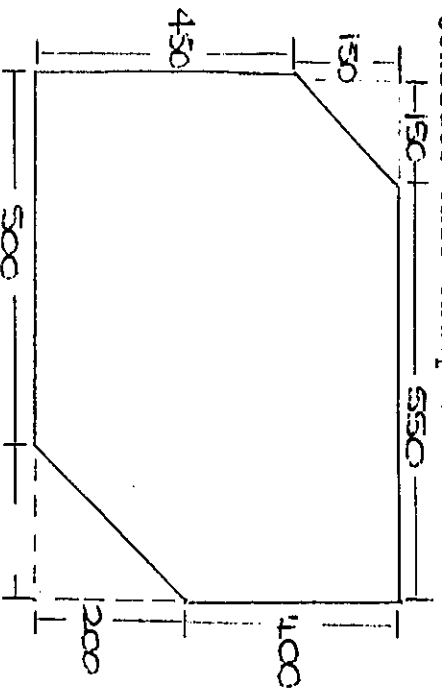
```
1. G78
2. G90
3. G1 X0 Y0
4. G1 X300 Y0 Z200
5. G2 X700 Y0 Z200
6. G1 X900 Y0
7. G1 X900 Y400
8. G1 X600 Y600
9. G1 X150 Y600
10. G2 X0 Y450 Z-150
11. G1 X0 Y0
12. M30 A1
```

Try the following examples:-



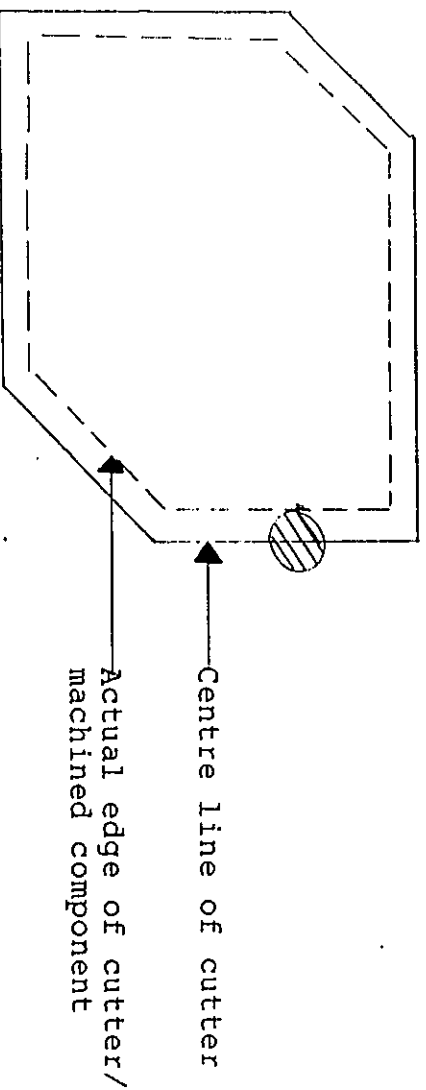
## CUTTER COMPENSATION

Consider this example:-



```
G1 X0 Y0
G1 X500 Y0
G1 X700 Y200
G1 X700 Y600
G1 X150 Y600
G1 X0 Y450
G1 X0 Y0
```

If the above component was cut with a cutter of 10 mm diameter then the component would be cut undersize by 5 mm (the radius of the cutter) because we programmed the **CENTRE LINE** of the cutter.



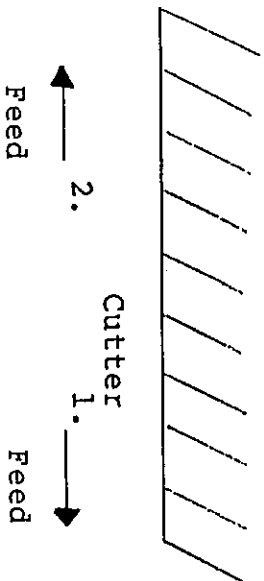
To enable us to still program the component as previous, but still get a true sized component we use **CUTTER COMPENSATION**.

This compensates for the size of the cutter, ie. moves away from the programmed path a set amount (1/2 the diameter of the cutter (radius)).

This amount is stored in Mode +

There are 32 tool stores and to call up a tool store within a program you enter the tool store number under T.

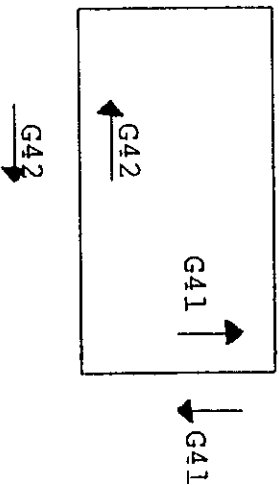
**RULE (1)**



If you can imagine yourself stood on top of the cutter, facing the direction of the feed:-

1. You are on the **RIGHT-HAND** side of the workpiece you use **G42**
2. You are on the **LEFT-HAND** side of the workpiece you use **G41**

**RULE (2)**



Cutting **anti-clockwise** round the outside of the component use **G42**.

Cutting **clockwise** round the outside of the component use **G41**.

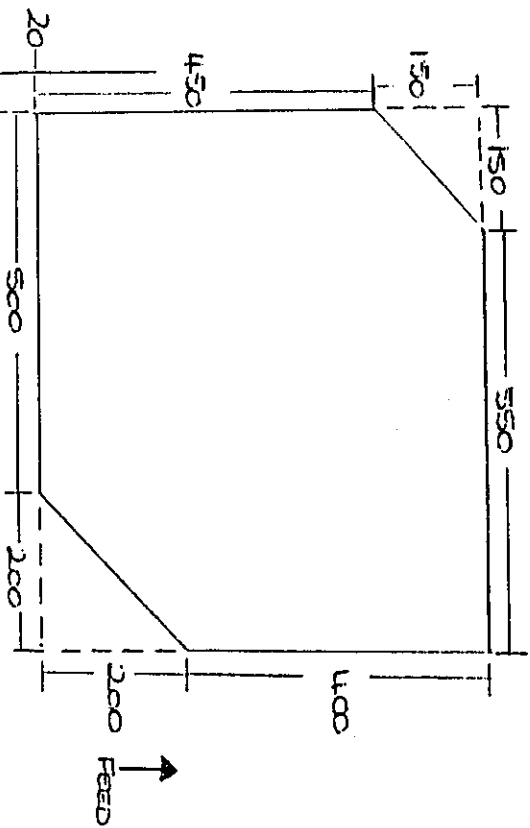
Cutting **anti-clockwise** round the inside of the the component use **G41**.

Cutting **clockwise** round the inside of the component use **G42**.

**TO CANCEL COMPENSATION** we use **G40**.

Cutter compensation must be put on and cancelled with a **straight line** move ( of a distance which is more than the radius of the cutter being used) before and after cutting respectively.

Consider the previous example:-



```
G78  
G90  
G1 X-20 Y0
```

(Start of main program)  
(Absolute programming)  
(A move away from the component so that compensation can be set, this must be at least the distance of the radius of the cutter being used.)

```
G42 T1 - 32  
  
G1 X0 Y0
```

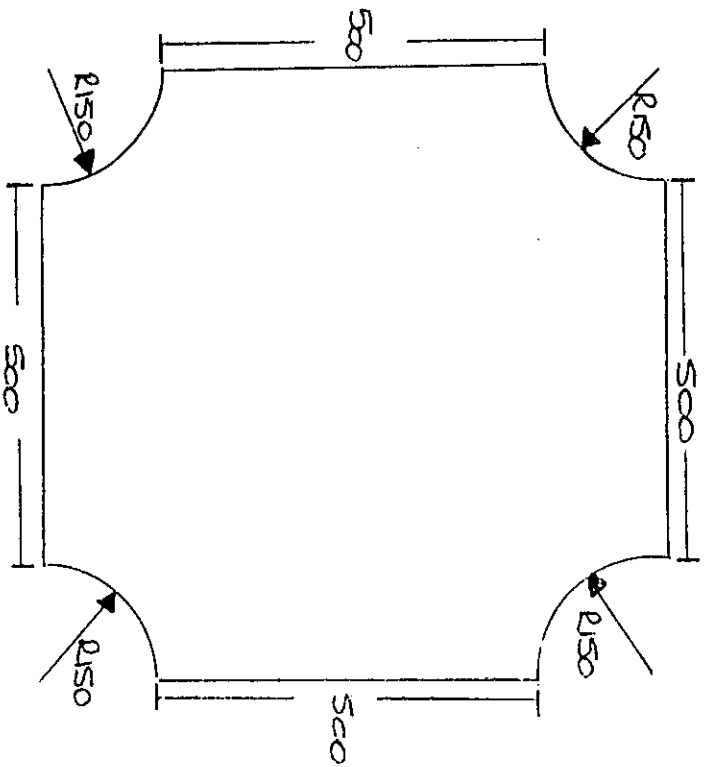
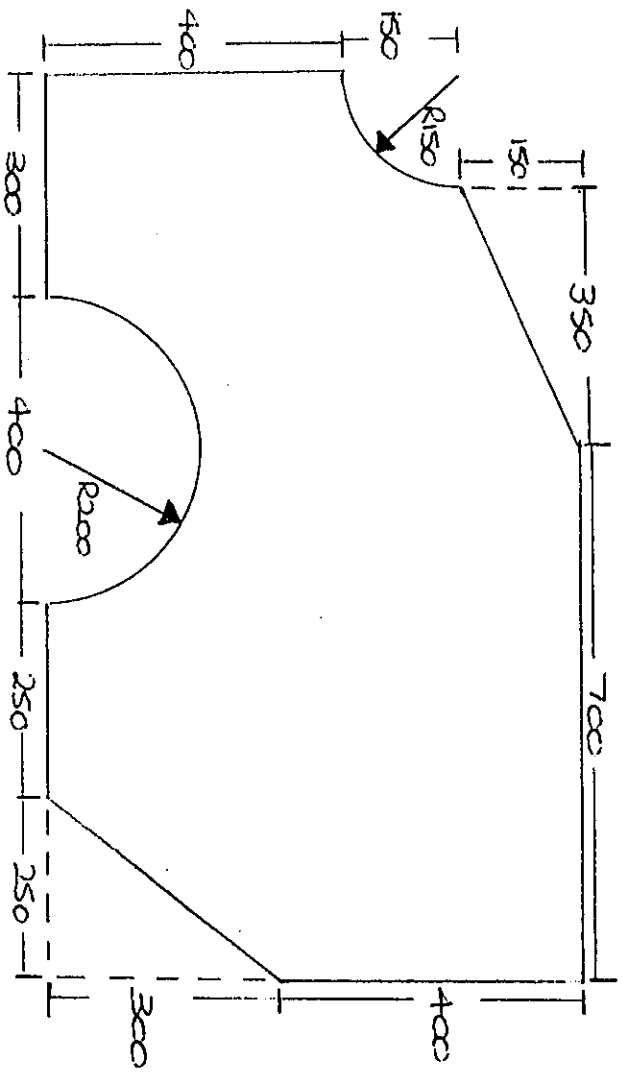
(G42 - cutter compensation to the right of the workpiece (anti-clockwise round the outside) using the chosen tool store number. It is not necessary to know the exact size of the cutter at this moment, it can be entered later.  
(Straight line move to activate cutter compensation)

```
G1 X500 Y0  
G1 X700 Y200  
G1 X700 Y600  
G1 X150 Y600  
G1 X0 Y450  
G1 X0 Y0  
G40  
G1 X0 Y-20  
M30 A1
```

(Code for cancelling compensation)  
(Straight line move to activate cancellation)  
(End of main program)

NB:- It is good machining practice to feed on to and off the component in a straight line as possible for compensation, this saves the cutter having to do any unnecessary moves.

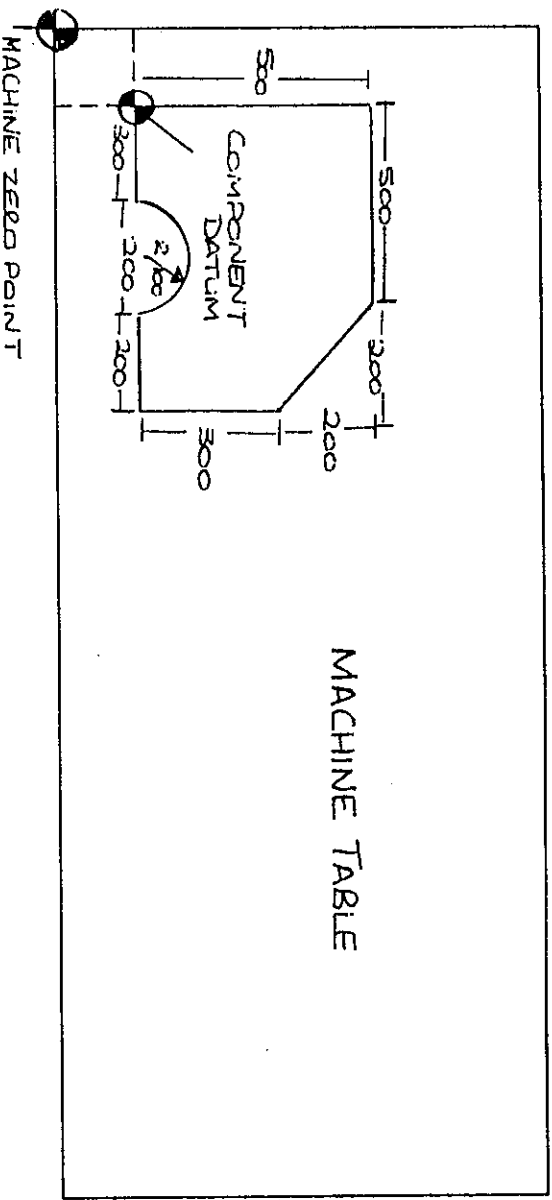
Try these two examples using the previous example as a guide line:-



### COMPONENT DATUM POINTS (MOVING ZERO POINTS)

We have used the bottom left hand corner of the workpiece as the component datum (ie. all dimensions are relative to this point).

In practice the component would be situated somewhere on the machine table.



The component datum has a distance from the bottom left hand corner of the machine table (Machine Zero Point). In this case its 50 along the 'X' axis and 30 up in the 'Y' axis.

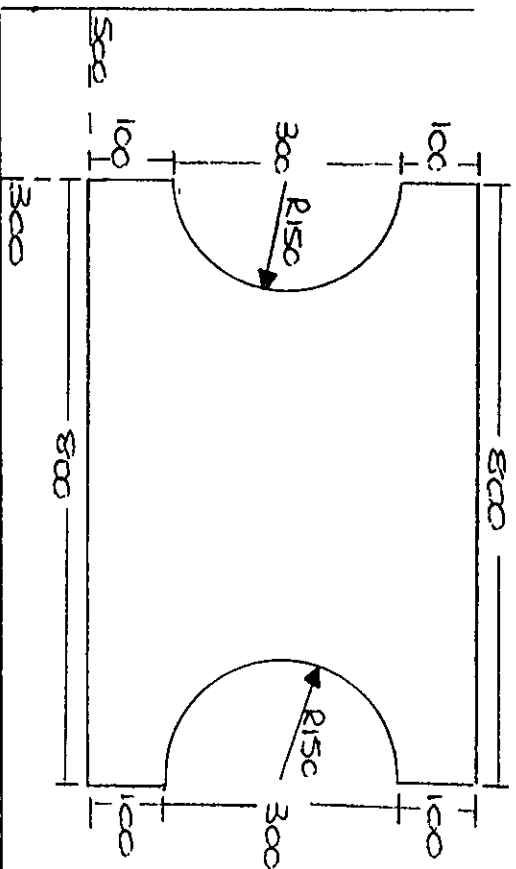
If there wasn't the 'moving zero point' facility the program would have to have 50 added on to every 'X' dimension and 30 on to every 'Y' dimension.

```

eg:  G1 X50 Y30          (Previously X0 Y0 )
      G1 X350 Y30        (Previously X300 Y0 )
      G2 X550 Y30        (Previously X500 Y0 )
      G1 X750 Y30        (Previously X700 Y0 )
      G1 X750 Y330       (Previously X700 Y300)
      G1 X550 Y530       (Previously X500 Y500)
      G1 X50 Y530        (Previously X0 Y500)
      G1 X50 Y30         (Previously X0 Y0 )
  
```

What we need to do is set the bottom left hand corner of the workpiece (or the desired co-ordinates) to equal X0 Y0 when on the machine table.

USE OF G92



```
G78
G90
G0 X500 Y300
G92 X0 Y0
```

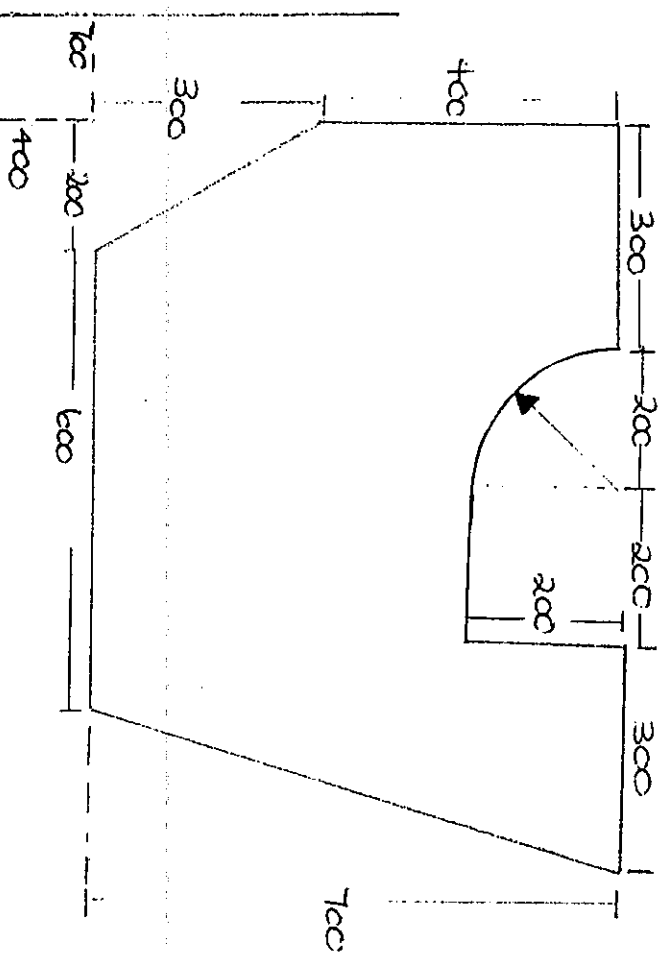
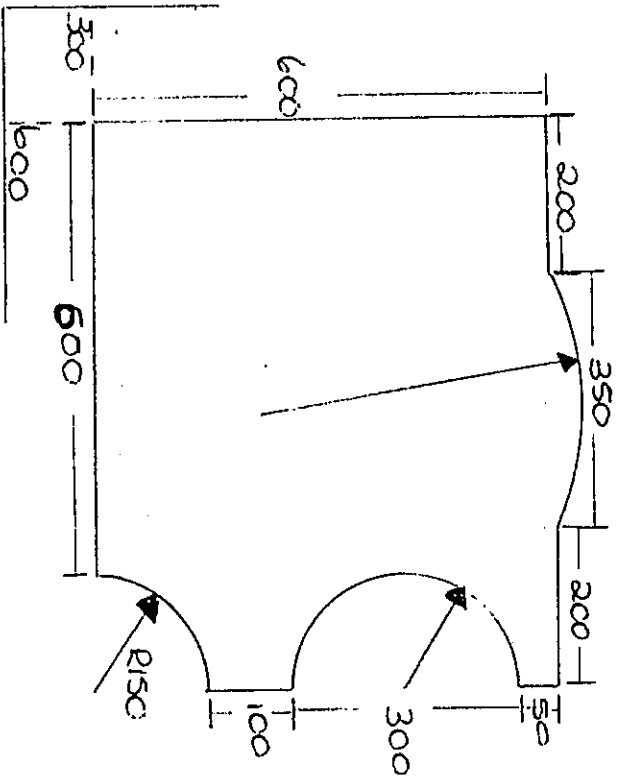
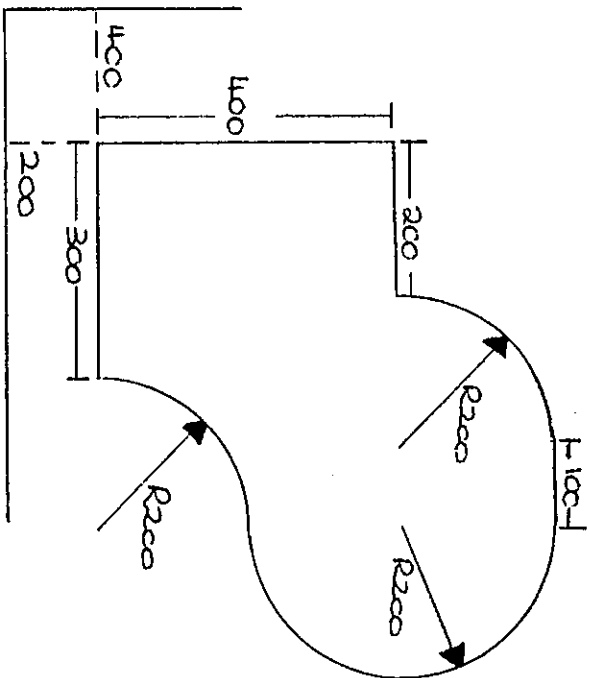
(Start of main program)  
 (Absolute programming)  
 (This move takes us to the component datum point relative to the machine zero point).  
 (The G92 will set the X and Y co-ordinates to whatever you want them to be, in this and most cases it is X0 Y0, so now the component can be programmed as before)  
 (Move away from the component to enable us to set cutter compensation)  
 (Cutter compensation to the right of the workpiece)  
 (Straight line move to activate compensation)

```
G1 X-20 Y0
G42 T2 Y0
G1 X0 Y0
G1 X800 Y0
G1 X800 Y100
G2 X800 Y400 Z150
G1 X800 Y500
G1 X0 Y500
G1 X0 Y400
G2 X0 Y100 Z150
G1 X0 Y0
G40
G1 X0 Y-20
G1 X0 Y0
```

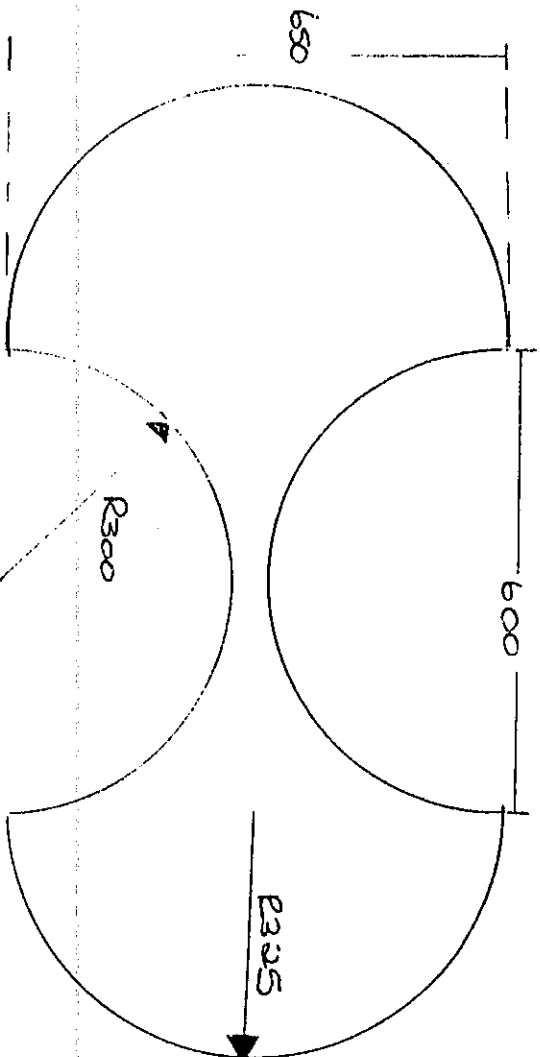
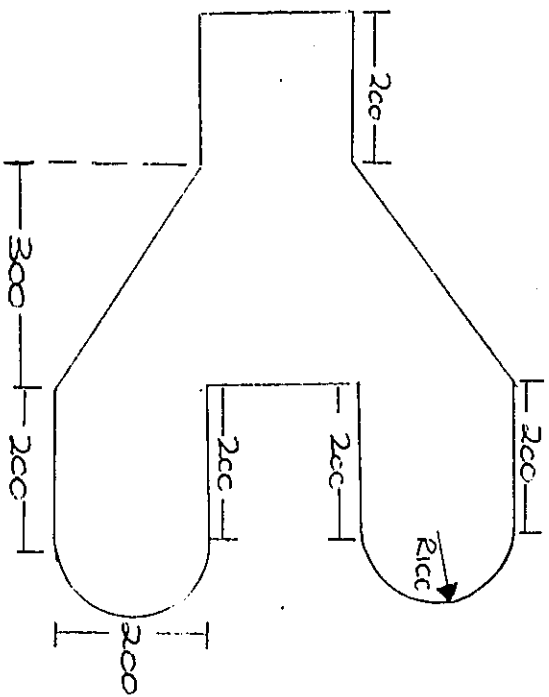
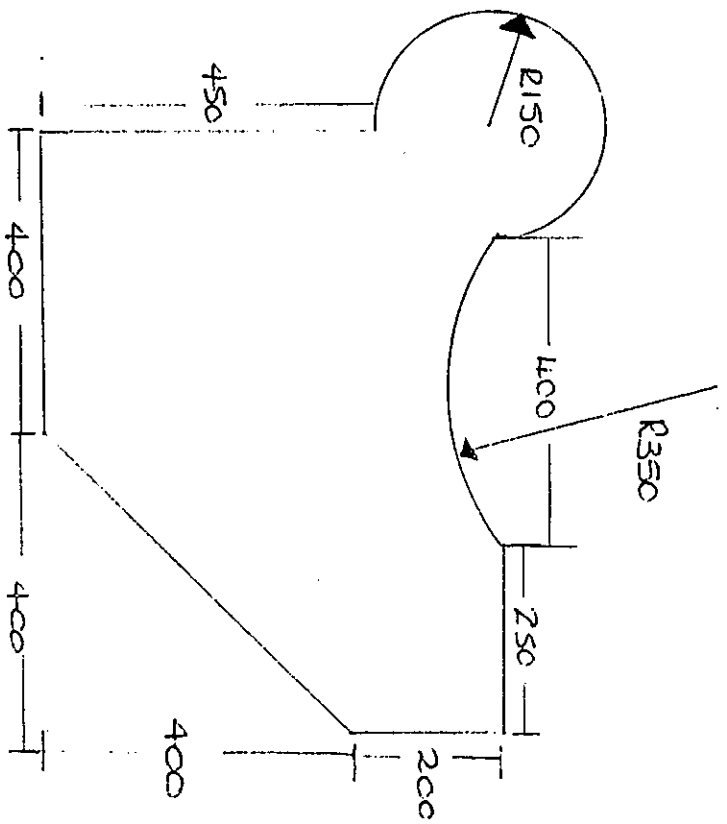
```
G92 X500 Y300
M30 A1
```

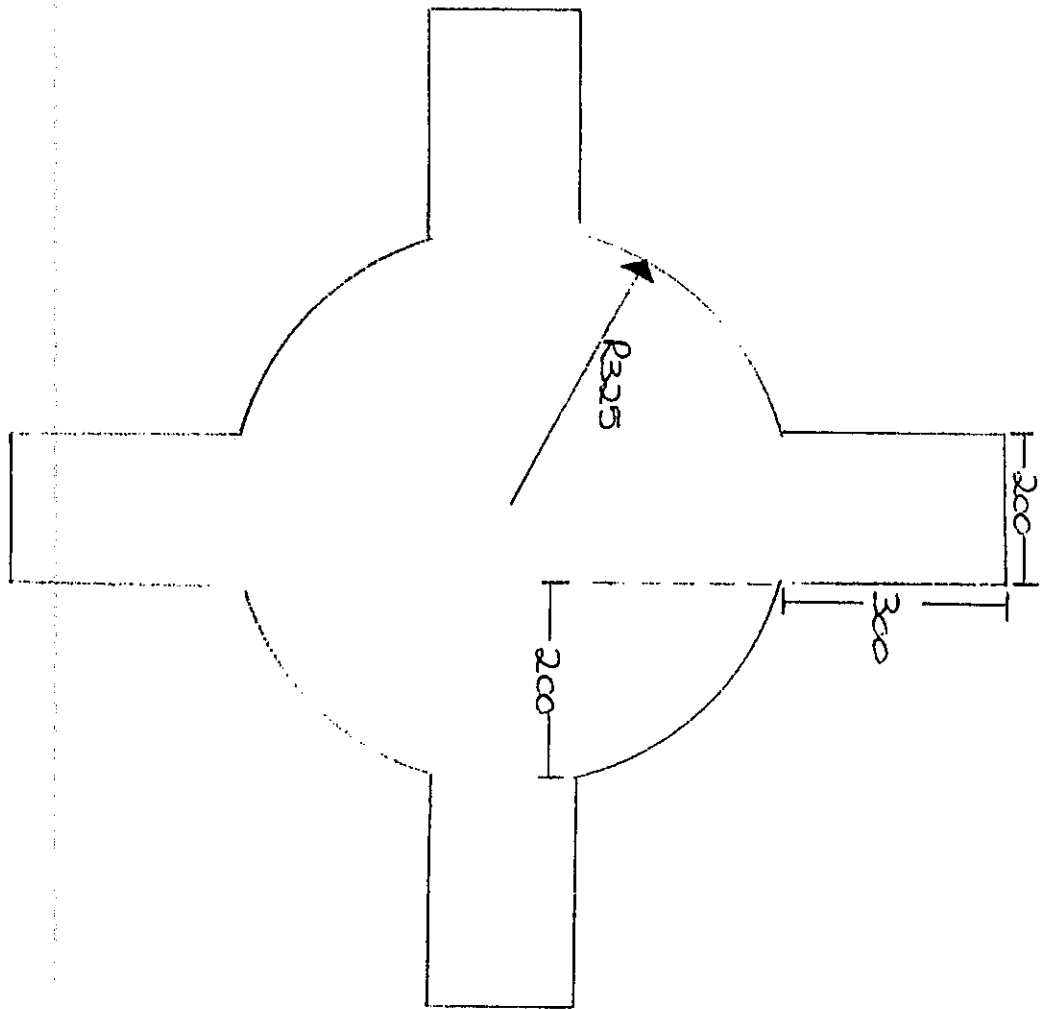
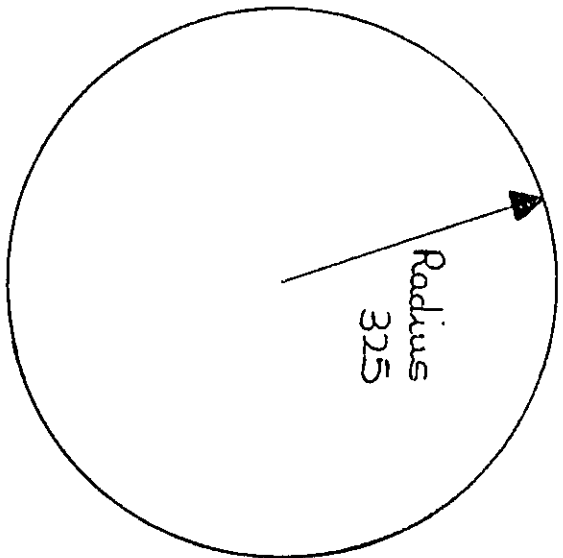
(Cancel compensation)  
 (Straight line move to activate cancellation)  
 (Back to the component datum point so that we can reset its true position relative to the machine zero point)  
 (We are giving the X and Y co-ordinates their true position, notice that they must correspond exactly with the values set at the beginning)  
 (End of main program)

Examples - with Component Datum Points.



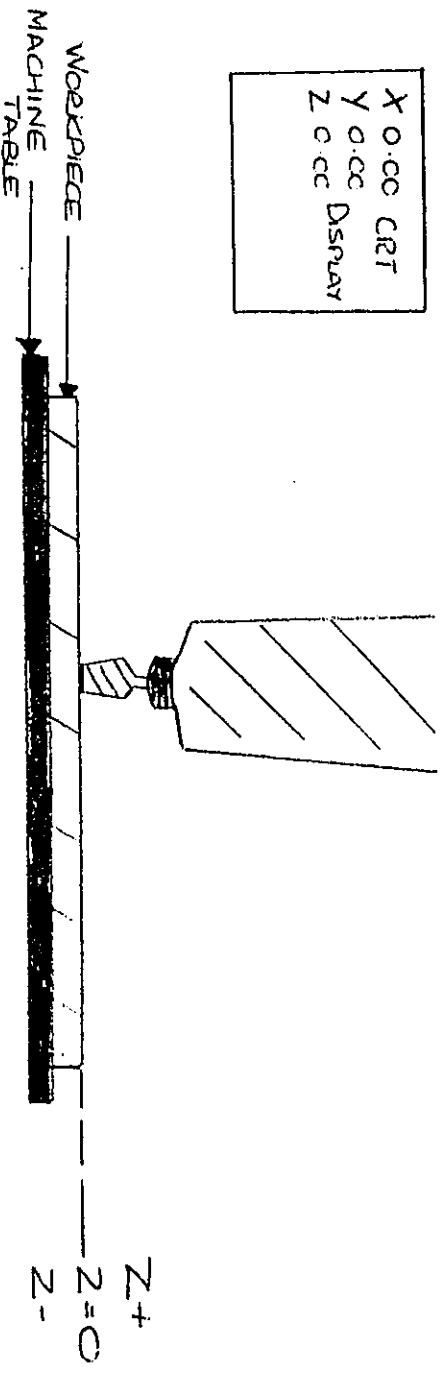
EXAMPLES.





## TOOL LENGTH COMPENSATION


X	O.CC	CRT
Y	O.CC	Display
Z	O.CC	Display



How to set tool length compensation.

Bring cutter down so that it is just touching the top of the workpiece using the manual panel and handwheel (vacuum clamping must be on if vacuum pumps are switched on).

Note the reading of the 'Z' axis on the CRT display.

Go into Mode + , enter the number of the tool store you require and then under the 'L' column enter the Z reading as a negative value.

When the tool store is called up into the program it will set the top of the workpiece to equal zero.

eg.

G78			(Start of main program)
G90			(Absolute programming)
G0	X200	Y200	(Rapid traverse to component datum)
G92	X0	Y0	(Set component datum to equal zero)
G0	X-20	Y0	(Move away to start feed in position)
G42	T1		(Call up of cutter compensation using Tool store 1.)
G1	X-10	Y0	(Straight line move to activate compensation)
M8		F5000	(Select left hand head down (on a two headed machine))
G43	Z5		(Call up cutter length offset to Z=5 using tool 1.)
G1	Z-10	F500	(Feed down to cutter depth)
G1	X0	Y0	
G1	X100	Y0	
G1	X100	Y100	
G1	X0	Y100	
G1	X0	Y-10	
G0	Z5		(Retract cutter to above the workpiece)
G40	X0	Y-20	(Cancel cutter compensation with straight line move)
M9			(Retract left hand head)
G44	Z0		(Cancel tool length offset to machine zero)
G0	X0	Y0	(Re-position over component zero)
G92	X200	Y200	(Re-reference control with respect to machine absolute)
G0	X0	Y800	(Off load position)
M30	A?		(End of main program with jump address)

