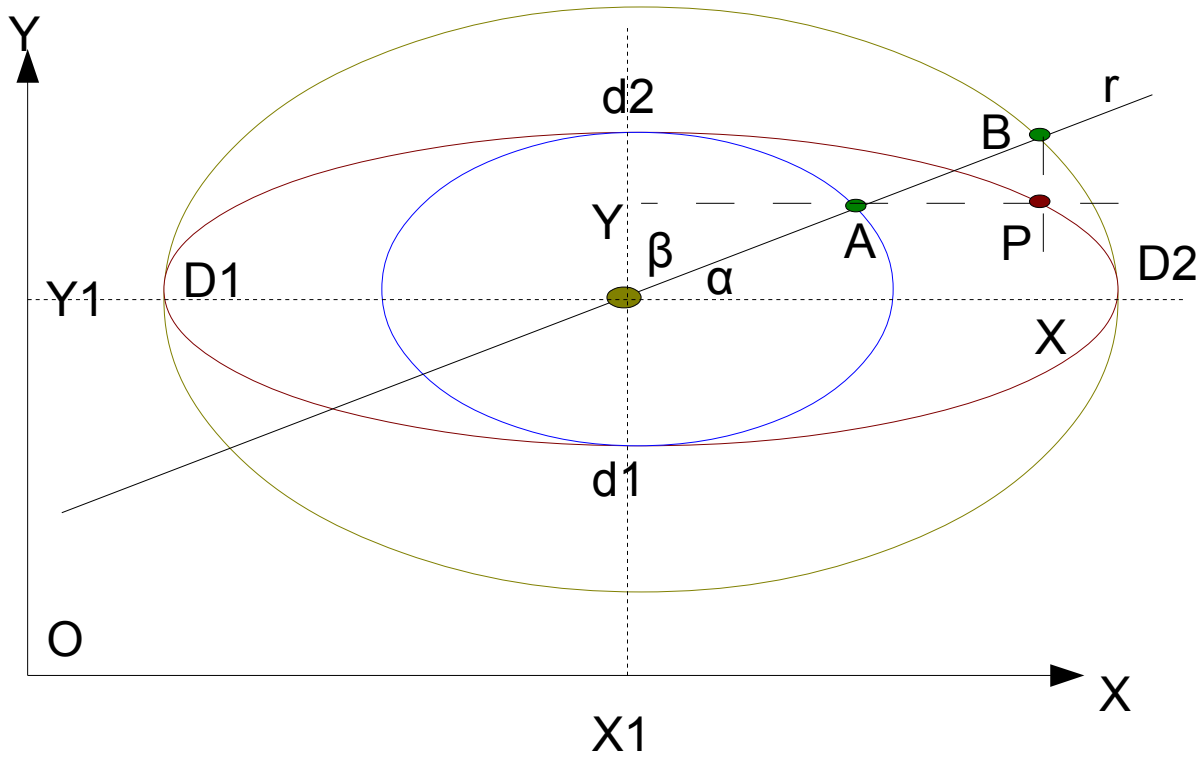
Desenhar uma elipse em si não quer dizer muito, mas pode ser um ponto de partida para a criação de formas complexas artísticas e agradáveis para aos olhos.
Além disso, não é dito que devemos fazer a figura inteira, mas apenas uma parte pode ser útil para projetar peças mais complexas.
Em última análise, daremos a possibilidade de criar uma elipse, todas as variações possíveis dependem de sua imaginação.

Vamos mostrar diferentes abordagens para a realização da elipse
(Neste contexto faleremos em desenhar, mesmo que na realidade é possível fazer uma operação de fresadura).



Vamos começar com os seguintes dados:

X, comprimento do eixo paralelo a X
Y, comprimento do eixo paralelo a Y
α, ângulo inicial

então vamos definir três variáveis que os descrevem

#100=X #101=Y #102=α

Devemos repetir n vezes o cálculo do ponto P
Portanto é preciso realizar uma subrotina o nome da qual é escolhido por você.

Por exemplo O5000

a sub-rotina vai ser chamada assim

M98 P5000 L(n)

n depende do valor do aumento em graus (em radianos se você prefere)

n=360,0 aumento / (desenho elipse completa)

então iremos definir uma variável para o aumento em graus, #103

então obtemos M98 P5000 L[360,0/#103]

Além disso, definimos duas variáveis para o centro da elipse

#104, a posição do centro em X
#105, a posição do centro em Y

Temos definido todos os dados que precisamos!

Escritura di um programa paramétrico

```
Ø1000 (ELIPSE)
G90G0
(configurações das variáveis)
#100=200(EIXO X)
#101=100 (EIXO Y)
#102=0(ANGULO INICIAL)
#103=1(AUMENTO GRAUS)
#104=100(posição do centro X)
#105=30(posição do centro Y)
```

Nossa ferramenta vai ser um marcador de grande diâmetro

```
T1M6 (DIA MARKER TIP 4)
(POSICIONAMENTO)
G90G0G54X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]S500M3
```

para um ângulo inicial de 0 ° corresponde a

```
G90G0G54X[#104+#100/2]Y[#105]
```

executar as normais operações para baixar a ferramenta

```
G43 H1 Z20
G01 F1000 Z1
F50 Z-0.3
```

chamada de subrotina O5000 n vezes com base no aumento

```
M98P5000L[360,0/#103]
```

repetida a sub-rotina podemos fechar o programa.

```
G90 G0 Z50
G0 G91 G28 Z0
M30
```

Este o subprograma O5000

```
O5000
F500G01X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]
#102=#102+#103(aumento ângulo)
M99
```

A sub-rotina aplica as fórmulas encontradas para aumentar o ângulo.
Aqui está o nosso programa completo

```
Ø1000 (ELLIPSE)
G90G0
(definições de variáveis)
#100=200(EIXO X)
#101=100 (EIXO Y)
#102=0(ANGULO INICIAL)
#103=1(AUMENTO GRAUS)
#104=100(posição do centro X)
#105=30(posição do centro Y)
T1M6(LAPIS DIA 4)
(POSICIONAMENTO)
G90G0G54X [#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]S500M3
G90G0G54X[#104+#100/2]Y[#105]
M98P5000L[360,0/#103]
G90 G0 Z50
G0 G91 G28 Z0
M30
```

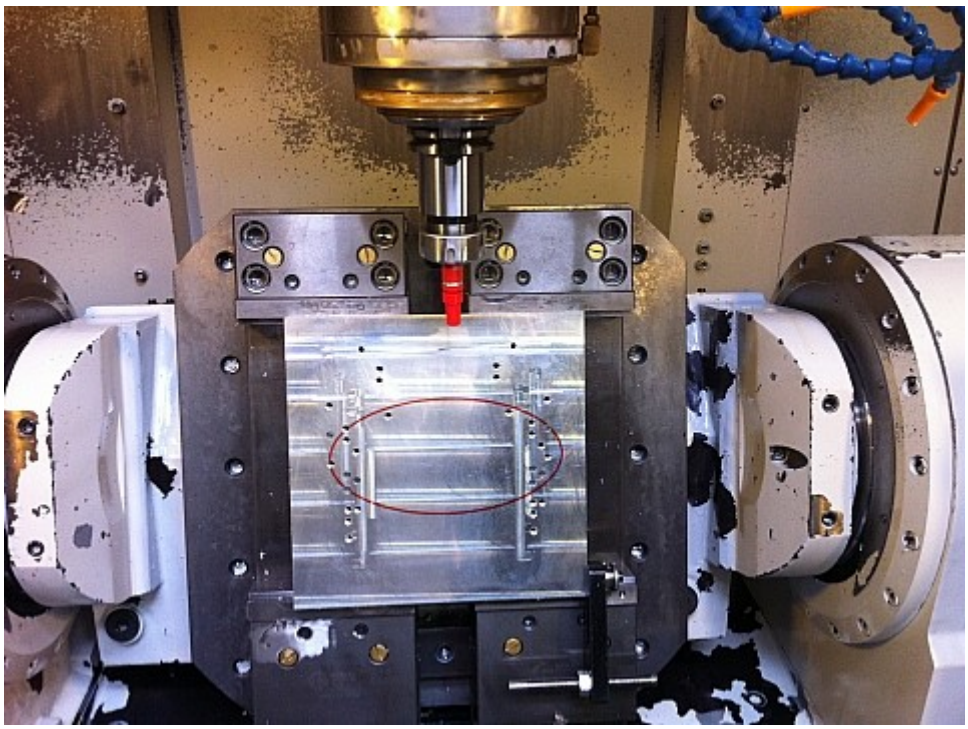
A rotação do fuso (baixa velocidade) è para usar o lápis em toda a superfície.

Se preferir, você pode usar um loop WHILE

```
Ø1000 (ELIPSE)
G90G0
(configurações das variáveis)
#100=200(EIXO X)
#101=100 (EIXO Y)
#102=0(ANGULO INICIAL)
#103=1(AUMENTO GRAUS)
#104=100(posição do centro X)
#105=30(posição do centro Y)
T1M6(LAPIS DIA 4)
G90G0G54X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]S500M3
G43H1Z20
G01F1000Z1
F50Z-0.3
WHILE[#102NE360]DO1
G01F500X[#104+#100/2*COS[#102]]Y[#105+#101/2*SIN[#102]]
#102=#102+#103
End1
G90G0Z50
G91G28G0Z0
M30
```

Estes alguns exemplos de como você pode desenhar uma elipse.
Você pode mudar o programa para desenhar as parcelas desejadas e as posições no plano. É uma indicação aproximada, nenhuma verificação de erro.

Um possível resultado é mostrado na figura



O tamanho e a forma da elipse dependem dos valores que você escolher.
Se você definir um valor de X=100 e Y=200 vai ter uma elipse, como da figura mas rodada de 90°.

Claro que você pode obter figuras complexas e artísticas, utilizando rotação ou ou alterando a posição da elipse, executando apenas algumas partes ....

Neste caso, dado que a solução do problema é mais longa, é melhor criar uma macro que pode tirar qualquer elipse no plano sem intervir cada vez no programa (variável).
A criação de uma macro requer mais esforço, mas os resultados serão, sem dúvida, recompensados.

Por que escrever um macro?

Um programa paramétrico (geralmente) resolve um particolar problema, é rápido e conveniente, mas nem sempre é fácil de manusear.
Quando nos deparamos com uma infinidade de soluções que nós sabemos o princípio, mas não a solução deve-se transformar o programa paramétrico em macro:
este, uma vez definido e testado, permite a solução do "sem fim" problemas, é prático, pois que não intervém diretamente no programa, exceto através dos argumentos, o que torna mais seguro. Também é possível detectar erros dos parâmetros, o que é viável para programas paramétricos, mas que normalmente não é feito.
Nós tentamos esboçar uma macro possível!

Entre as várias maneiras para chamar uma macro utilizeremos o tipo G.
Vamos usar o codicgo 450, número associado com o parâmetro (6054) para o programa O9014.
Nada impede de usar uma chamada com o G65.

G450 X Y Z U W I J A F   a nossa macro

(X #24 X centro da elipse)
(Y #25 Y elipse centro)
(Z #26 valor de Z para escitura)

(U #21 eixo X)
(V #22 eixo Y)
(I #4   ângulo inicial)
(A #1   aumento ângulo)
(J #5   ângulo final)
(F #9   avanço)

Primeiro passo: verificar a presença dos argumentos

```
IF[#21EQ#0]THEN#3000=1(FALTA EIXO X ELIPSE)
IF[#22EQ#0]THEN#3000=2(FALTA EIXO Y ELIPSE)
IF[#1EQ#0]THEN#3000=5(FALTA ângulo INICIAL)
IF[#9EQ#0]THEN#3000=7(FALTA AVANCO)
```

Você é livre para substituir os alarmes, mas tente ser simples e concreto, e não como neste exemplo que, à primeira vista parece bom, mas depois de meses pode não significar nada.

```
IF[#1EQ#0]THEN#3000=5(NOT DEF A)
```

Os argumentos de uma macro nem sempre são todos obrigatórios, alguns podem ter valores padrão para usar quando eles são omitidos na chamada macro:
é o exemplo de I e J ou de X, Y e Z.
O padrão elipse vai ser desenhada de forma completa e posicionada com seu centro no centro da origem de trabalho a um valor do plano Z igual a zero.

Considerando isso, a macro pode ser chamada com o seguinte formato:

```
G450 U V A F
```

A escolha é feita tendo em conta duas considerações:
- ilustrar o uso de técnicas de controle
- facilitar a chamada da macro com o uso de poucos argumentos.

(É sempre possível utilizar a função de compensação origem G52, ou utilizar G91)

Assim, para as variáveis I, J, X, Y e Z temos de agir de forma diferente.

Se #24, #25 e #26 não estão presentes na chamada macro terão o valor zero.
(Origem elipse = origem de trabalho, e plano Z de elipse = 0)

```
IF[#24EQ#0]THEN#24=0
IF[#25EQ#0]THEN#25=0
IF[#26EQ#0]THEN#26=0
```

Se o #5 e #6 não estão presentes na chamada terão como valor respectivamente 0 e 360 (elipse completa).

```
IF[#4EQ#0]THEN#4=0
IF[#5EQ#0]THEN#5=360
```

Segundo passo: determinar o intervalo de valores aceitáveis.

X e Y representam o centro de uma elipse, não podemos fazer suposições sobre sua localização, que pode ser em qualquer lugar do plano.

Z é o plano da escrita, pode tomar qualquer valor: zero positivo ou negativo.

U, eixo X. Para a existência da elipse, este valor deve ser positivo e diferente de zero U> 0

V, eixo Y. Para a existência da elipse, este valor deve ser positivo e diferente de zero V> 0

I ângulo inicial. O ângulo deve estar entre 0° e 360°, 0<=I<=360. Você pode escolher se considerar os ângulos negativos, você precisará alterar o intervalo de valores aceitáveis e as relações que considerá-los.

A aumento ângulo. Do ponto de vista matemático (A) deve ser contido em um ângulo de volta (360°). Porem (A) não deve ter altos valores de incremento angular, pois maiores aumentos correspondem menores pontos calculados, em seguida, pior aproximação da elipse.
Lembremo-nos que estamos a desenhar uma elipse, se os pontos estão distantes nossa figura não é mais uma curva, mas uma linha quebrada.

Devemos, portanto, impor um valor máximo de aumento além do qual não podemos ir. Se você fizer os testes vai achar o melhor valor.

Nós definimos 2° como o valor máximo 0<A<=2

O sinal do nosso ângulo vai ser por entanto sempre positivo.

J, ângulo final. O valor deve estar entre 0 e 360. 0<I<=360 certificando-se que I e J sejam diferentes. I≠J

F, avanço. Temos que considera-lo positivo e diferente de zero F> 0

A faixa de valores foi determinada:

```
IF[#21LE0]THEN#3000=8(EIXO X ERRADO)
IF[#22LE0]THEN#3000=9(EIXO Y ERRADO)
IF[#4LT0]THEN#3000=10(ANGULO INICIAL <0)
IF[#4GT360]THEN#3000=11(ANGULO INICIAL >360)
IF[#5LT0]THEN#3000=12(ANGULO FINAL <1)
IF[#5GT360]THEN#3000=13(ANGULO FINAL >360)
IF[#4EQ#5]THEN#3000=14(I,J COINCIDENCIA ANGULOS)
IF[#1LE0]THEN#3000=15(AUMENTO ANGULO NEGATIVO)
IF[#1GT2]THEN#3000=16(AUMENTO ANGULO >2)
IF[#9LE0]THEN#3000=17(AVANCO ERRADO)
```

Isto é o mínimo necessário para o funcionamento adequado. Claro que você pode introduzir qualquer tipo de controle que pode determinar de forma inequívoca o valor da argumentos. Ou você pode verificar certos valores de diferentes maneiras.

Por exemplo,
você pode controlar o valor de ângulo superior a 360°,
você pode controlar ângulos negativos,
o avanço pode ser aceito com valor negativo.

Existem vários tipos de controles que podemos fazer; aqui vamos mostrar o princípio, você pode usar ou criar os mais adequados.

Terceiro passo: salvar os dados modais.

As condições de trabalho não devem ser alteradas.
O operador não espera que a macro mude as condições de trabalho, por isso é aconselhável se você alterar um estado modal voltar para o original.

Tanto quanto sabemos, a nossa macro não intervém sobre as origens de trabalho, planos, compencaçao raio o cumprimento.....
Por segurança vamos salvar o estado de G90, G0.
Não é necessário salvar estes valores (neste programa), você pode tomar uma posição final para garantir a segurança. É sempre bom salvar o estado modal destas variáveis, porque não sabemos ainda o que vamos escrever (o pior que pode

acontecer é escrever algumas linhas de código: é preferível poucas linhas que dá mais garantia do que uma situação inesperada).

```
#2=#4001(G00,1,2,3,33)
#3=#4003(G90,91)
```

A fim de mostrar o conceito vamos salvar os valores dos eixos, trazendo a máquina nas condições de trabalho antes da chamanda macro

```
#10=#5001(X ATUAL)
#11=#5002(Y ATUAL)
#12=#5003(Z ATUAL)
```

E' importante realizar estas três etapas em cada macro que você escreve, fará seu trabalho mais seguro e menos inesperados.
(Controle presença argumentos, controle valores de argumentos, backup de dados modal)

Podemos, então, continuar com a escritura da macro (até agora nem mesmo um movimento)

Temos que seguir o conceito formulado no programa paramétrico paralelo, possivelmente adaptando-o e fazendo controles.

```
(POSICIONAMENTO)
G90G0X[#24+#21/2*COS[#4]]Y[#25+#22/2*SIN[#4]]
(Z-SEGURANÇA)
G43H1Z[#26+5]
(ABORDAGEM SEGURA)
G01F500Z[#26+1]
(ABORDAGEM A Z)
F50Z#26
(AVAMCO)
F#9
```

Neste ponto, temos de decidir o rumo da viagem. Os métodos de escolha, como sempre, são variados. Nós usaremos os valores de #4 e #5 para determinar a nossa escolha.

#4 ->> ângulo inicial #5 ->> ângulo final

Para executar um movimento anti-horário será #4 menor de #5, para um movimento no sentido horário o contrário. (Lembre-se que consideramos a gama de valores de 0° a 360°)

```
IF[#5LT#4]THEN#1=-#1
```

Inverter o sinal de #1 (ângulo de crescimento) pois a #5 (ângulo final) acaba por ser menor do que #4 (o ângulo inicial), o que indica uma rotação no sentido horário.

Escritura ciclo.
É claro que o loop WHILE terminará quando os dois ângulos são os mesmos.
Podemos comparar a distância entre eles e determinar a coincidência, podemos considerar uma variável e determinar seu valor com base na saída para fora do loop, podemos calcular e considerar as repetições ou o valor final do ângulo ....
Hoje vou mostrar um método diferente que tenta usar um loop infinito.
Não sabemos o número de incrementos angular, muito menos o número de repetições; decidimos usar um ciclo repetitivo sem fim do qual sairemos sob determinadas condições (os dois ângulos iguais).
Esta escolha é oferecida novamente para espírito académico e sinta-se livre para modifica-la.

Um loop infinito realiza-se assim

```
DO n
---
---
END n
```

E' preciso prestar muita atenção a este ciclo, porque a condição de saída deve ser criada especificamente.

Também prestar atenção ao controle do aumento que não é dito é um múltiplo da variação angular (#5 - #4), o que levaria a suposições erradas.

Suponha que o ângulo inicial é de 1,34 °, o final de 23,56 ° e que o aumento é de 0,5 °.
A diferença entre 23,56 e 1,34 não é um múltiplo de 0,5.
Devemos, portanto, introduzir uma verificação para determinar com base na posição atual e futura como comportar-se para não exceder os limites.

Nossa abordagem é a seguinte:
verificar se o aumento do valor atual não excede o valor final.
Se isso acontecer, (estamos no último aumento) vamos forçar o valor final do ângulo (o aumento será menor do que o definido, mas irá garantir a correta elipse).

```
IF[ABS[#5-#4]LT[ABS[#1]]]THEN#4=#5
```

Se a diferença dos ângulos (o próximo e atual) em valor absoluto (sem sinal) é menor do que o aumento (que passou o ângulo final), então vamos definir o aumento como esta diferença.

Vamos tentar!

```
(definição variáveis )
(#6=1 CICLO OK)
(#6=0 STOP CICLO)
#6=1
(CICLO DE CONSTRUÇÃO ELIPSE)
DO1
(POSIcIONAMENTO)
X[#24+#21/2*COS[#4]]Y[#25+#22/2*SIN[#4]]
(CONTROLE CICLO, #6=0-->> OUT)
IF[#6EQ0]THENGOTO100
(AUMENTO ANGOLAR)
#4=#4+#1
(CONTROLE AUMENTO ANGOLAR)
(SE EXCEDEU O LIMITE)
IF[ABS[#5-#4]LT[ABS[#1]]]THEN#4=#5
(ULTIMO AUMENTO, FINAL CICLO)
IF[#4EQ#5]THEN#6=0
END1
(REINICIAR VALORES MODAIS)
N100G90G0Z#12
X#10Y#11
G#3G#2
M99
```

Releia a macro, faça as alterações apropriadas .... e experimente.
A macro-proposta é apenas a base para modificações e melhorias.

Querendo frezar devemos considerar outros fatores: controle dos valores de Z, e gestão de compensação, o controle de repetições (passadas) …

Muito interessante é a realização de uma elipse no torno seja no plano XZ que em YZ.
Além disso, podemos considerar as rotações, o que implica controles adicionais: o cálculo do ponto e rotação em torno da origem.
Aqui eu quis enunciar o princípio para realizar uma elipse, tudo está em suas mãos (se precisar de ajuda me chame).

Quando você escreve um programa, seja paramétrico ou macro, está tentado querer terminar rapidamente, sem escrever comentários, fazendo uso de complexas fórmulas matemáticas, usando as variáveis que não se lembra o significado ....

Agir desta forma:

- Um programa super-comentado
- Um programa sem comentários para colocar na cnc

O primeiro é para você para decifrar a macro, para conhecer cada símbolo ...
O segundo é o que irá por em seu controle: você não terá que intervir diretamente no programa então não vai precisar de comentários.

Bem, a última etapa é a prova!
Normalmente nunca uma macro funciona da primeira vez ...
Você só tem que experimentar, corrigir, tentar novamente etc ..

<u>para atingir o objectivo: desenhar uma elipse!</u>

Relatamos nossa macro descomentada

```
O9014(ELLISSE)
(G450 X Y Z U W I J A F)
(X#24  X centro da elipse)
(Y#25  Y elipse centro)
(Z#26  valor de Z para escitura)
(U#21  eixo X)
(V#22  eixo Y)
(I#4   ângulo inicial)
(A#1   aumento ângulo)
(J#5   ângulo final)
(F#9   avanço)
IF[#21EQ#0]THEN#3000=1(FALTA EIXO X ELIPSE)
IF[#22EQ#0]THEN#3000=2(FALTA EIXO Y ELIPSE)
IF[#1EQ#0]THEN#3000=5(FALTA ângulo INICIAL)
IF[#9EQ#0]THEN#3000=7(FALTA AVANCO)
IF[#24EQ#0]THEN#24=0
IF[#25EQ#0]THEN#25=0
IF[#26EQ#0]THEN#26=0
IF[#4EQ#0]THEN#4=0
IF[#5EQ#0]THEN#5=360
IF[#21LE0]THEN#3000=8(EIXO X ERRADO)
IF[#22LE0]THEN#3000=9(EIXO Y ERRADO)
IF[#4LT0]THEN#3000=10(ANGULO INICIAL <0)
IF[#4GT360]THEN#3000=11(ANGULO INICIAL >360)
IF[#5LT0]THEN#3000=12(ANGULO FINAL <1)
IF[#5GT360]THEN#3000=13(ANGULO FINAL >360)
IF[#4EQ#5]THEN#3000=14(I,J COINCIDENCIA ANGULOS)
IF[#1LE0]THEN#3000=15(AUMENTO ANGULO NEGATIVO)
IF[#1GT2]THEN#3000=16(AUMENTO ANGULO >2)
IF[#9LE0]THEN#3000=17(AVANCO ERRADO)
#2=#4001
#3=#4003
#10=#5001
#11=#5002
#12=#5003
```

```
G90G0X[#24+#21/2*COS[#4]]Y[#25+#22/2*SIN[#4]]
G43H1Z[#26+5]
G01F500Z[#26+1]
F50Z#26
IF[#5LT#4]THEN#1=-#1
#6=1
DO1
X[#24+#21/2*COS[#4]]Y[#25+#22/2*SIN[#4]]F#9
IF[#6EQ0]THENGOTO100
#4=#4+#1
IF[ABS[#5-#4]LT[ABS[#1]]]THEN#4=#5
IF[#4EQ#5]THEN#6=0
END1
N100G90G0Z#12
X#10Y#11
G#3G#2
M99
```