

Appendix (A)

Program No. (1)

```
;;;Lower die block (2D)
(defun c:Lower_Die_Block_2D(/ ang len p1 p2 p3 p4 )
(initget 1)
(setq p1 (getpoint "\nFirst corner of Lower Die Block: "))
(setvar "ORTHOMODE" 1)
(setvar "osmode" 0)
(initget 1)
(setq p2 (getpoint p1 "\nSecond corner of Lower Die Block: "))
(setq len (distance p1 p2) ang (angle p1 p2))
(setq p3 (polar p2 (+ ang (/ pi 2)) len ))
(setq p4 (polar p1 (+ ang (/ pi 2)) len ))
(setvar "cmdecho" 0)
(command ".pline" p1 p2 p3 p4 "C")
(alert "Lower_Die_Block_2D created!")
(princ)
)
```

Program No. (2)

```
;;;Lower die block(2D with circle)
(defun c:Lower_Die_Block_2DC(/ ang len p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 H
K )
(initget 1)
(setq p1 (getpoint "\nFirst corner of square: "))
(setvar "ORTHOMODE" 1)
(setvar "osmode" 0)
(initget 1)
(setq p2 (getpoint p1 "\nSecond corner of square: "))
(setq len (distance p1 p2) ang (angle p1 p2))
(setq p3 (polar p2 (+ ang (/ pi 2)) len ))
(setq p4 (polar p1 (+ ang (/ pi 2)) len ))
(setq H (distance p1 p3))
(setq K (distance p2 p4))
(setq p5 (polar p1 (+ ang (/ pi 4)) (/ H 2)))
```

```

(setq p6 (polar p1 (+ ang (/ pi 4)) (/ H 8)))
(setq p7 (polar p1 (+ ang (/ pi 4)) (/ (* 7 H) 8)))
(setq p8 (polar p2 (+ ang (/ (* 3 pi) 4)) (/ K 8)))
(setq p9 (polar p2 (+ ang (/ (* 3 pi) 4)) (/ (* 7 K) 8)))
(setvar "cmdecho" 0)
(command ".pline" p1 p2 p3 p4 "C")
(command "circle" p6 (/ len 15))
(command "circle" p7 (/ len 15))
(command "circle" p8 (/ len 15))
(command "circle" p9 (/ len 15))
(alert "Lower_Die_Block_2DC created!")
(princ)
)

```

Program No. (3)

```

;;;Lower die block(3D)
(defun c:Lower_Die_Block_3D(/ ang len p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 H
K )
(initget 1)
(setq p1 (getpoint "\nFirst corner of Lower Die Block: "))
(setvar "ORTHOMODE" 1)
(setvar "osmode" 0)
(initget 1)
(setq p2 (getpoint p1 "\nSecond corner of Lower Die Block: "))
(setq len (distance p1 p2) ang (angle p1 p2))
(setq p3 (polar p2 (+ ang (/ pi 2)) len ))
(setq p4 (polar p1 (+ ang (/ pi 2)) len ))
(setq H (distance p1 p3))
(setq K (distance p2 p4))
(setq p5 (polar p1 (+ ang (/ pi 4)) (/ H 2)))
(setq p6 (polar p1 (+ ang (/ pi 4)) (/ H 8)))
(setq p7 (polar p1 (+ ang (/ pi 4)) (/ (* 7 H) 8)))
(setq p8 (polar p2 (+ ang (/ (* 3 pi) 4)) (/ K 8)))
(setq p9 (polar p2 (+ ang (/ (* 3 pi) 4)) (/ (* 7 K) 8)))
(setvar "cmdecho" 0)
(command ".pline" p1 p2 p3 p4 "C")
(command "circle" p6 (/ len 15))

```

```

(command "circle" p7 (/ len 15))
(command "circle" p8 (/ len 15))
(command "circle" p9 (/ len 15))
(setq pt_list (list p1 p2 p3 p4))
(command ".region" (ssget "_CP" pt_list) "")")
(command "_view" "_swiso" "")
(command ".extrude" (ssget "_CP" pt_list) """/ len 6)""")"
(command ".zoom" "all" "")
(setq pt1_list (list p1 p2))
(setq pt2_list (list p5 p6 p7 p8 p9))
(command "circle" p6 (/ len 10))
(command "circle" p7 (/ len 10))
(command "circle" p8 (/ len 10))
(command "circle" p9 (/ len 10))
(command ".extrude" (ssget "_F" pt2_list) """/ len 16) """)"
(command ".subtract" (ssget "_f" pt1_list)"""(ssget "_f" pt2_list)""")"
(command ".zoom" "all" "")
(setvar "cmdecho" 1)
(alert "Lower_Die_Block_3D Created!")
(princ)
)

```

Program No. (4)

```

(vl-load-com)
(defun c:guide_pin ()
  (setq p1      (getpoint    "\nPick centre point of guide pin:"))
  (setq d1      (getreal     "\nguide pin diameter:"))
  (setq gh      (getint      "\nHeight of guide pin:"))

  (command "circle" p1 "d" d1)
  (command "extrude" "L" "" gh ""))
  (command ".shade" "complete" "" _g "")"

  (command "_view" "_swiso" "")
  (command ".zoom" "all" "")
  (alert "guide_pin Created!")
  (princ)
)

```

)

Program No. (5)

```
;;;Assy_Die_Blocks (3D)
(vl-load-com)
(defun c:Assy_Die_Blocks(/ ang len p1 p2 p3 p4 p5 p6 p7 p8 p9 p10 p11 p12 p13 H
K )
(initget 1)
(setq p1 (getpoint "\nFirst corner of square: "))
(setvar "ORTHOMODE" 1)
(setvar "osmode" 0)
(initget 1)
(setq AFW 200)
(setq p2 (getpoint p1 "\nSecond corner of square: "))
(setq len (distance p1 p2) ang (angle p1 p2))
(setq p3 (polar p2 (+ ang (/ pi 2)) len ))
(setq p4 (polar p1 (+ ang (/ pi 2)) len ))
(setq H (distance p1 p3))
(setq K (distance p2 p4))
(setq p5 (polar p1 (+ ang (/ pi 4)) (/ H 2)))
(setq p6 (polar p1 (+ ang (/ pi 4)) (/ H 8)))
(setq p7 (polar p1 (+ ang (/ pi 4)) (/ (* 7 H) 8)))
(setq p8 (polar p2 (+ ang (/ (* 3 pi) 4)) (/ K 8)))
(setq p9 (polar p2 (+ ang (/ (* 3 pi) 4)) (/ (* 7 K) 8)))
(setvar "cmdecho" 0)
(command "Layer" "M" "Edge" "")
(command ".pline" p1 p2 p3 p4 "C")
(command "Layer" "M" "Circle1" "" )
(command "circle" p6 (/ len 15))
(command "circle" p7 (/ len 15))
(command "circle" p8 (/ len 15))
(command "circle" p9 (/ len 15))
(setq pt_list (list p1 p2 p3 p4))
(command "Layer" "M" "Region1" "" "Color" "green")
(command ".region" (ssget "_CP" pt_list) "")
(command "_view" "_swiso" "")
```

```

(command ".extrude" (ssget "_CP" pt_list) """/ len 6)"""
(command ".zoom" "all" """
(setq pt1_list (list p1 p2))
(setq pt2_list (list p5 p6 p7 p8 p9))
(command ".subtract" (ssget "_f" pt1_list)"""(ssget "_f" pt2_list)""")
(command "color" "red" """
(command "circle" p6 (/ len 16))
(command "circle" p7 (/ len 16))
(command "circle" p8 (/ len 16))
(command "circle" p9 (/ len 16))
(setq pt3_list (list p6 p7 p8 p9 ))
(command ".extrude" (ssget "_F" pt3_list) """ (/ len 2) """
(command ".zoom" "all" """
(command "Color" "yellow" """
(command ".copy" (ssget "_F" pt1_list)"""
(setvar "cmdecho" 1)
(princ)
)

```

Program No. (6)

```

;;;gear (3D)
(vl-load-com)
;This function converts Degrees to Radians.
(defun dtr (x)
;define degrees to radians function
(* pi (/ x 180.0))
;divide the angle by 180 then
;multiply the result by the constant PI
) ;end of function
*****;
(princ) ;load cleanly

;input from the data
(defun input ()
    (setq p1          (getpoint           \nPick centre point of gear))

```

```

(setq n          (getint      \nNumber of teeth:))
(setq mod        (getreal    \nModule:))
(setq gh         (getint      \nHeight of gear:))
(setq d1         (getreal    \nShaft diameter:))
)

```

;Calculating: inner circle, outer circle, pitch circle, teeth height

```

(defun calculating1 ()
  (setq oc          (* (+ n 2)mod))
  (if (>= mod 1.25) (setq ht 2.25) (setq ht 2.4))
  (setq h          (* ht mod))
  (setq ic         (- oc(* 2 h)))
  (setq pc         (* n mod))
  (setq cp         (* pi mod)))

```

;Error messages

```

(defun error_messages ()
  (if (>= d1 ic) (alert "Shaft diameter is to large!"))
  (setq x1 0))
)

```

```

(defun calculating2 ()
  (setq h1          (/ oc 2))
  (setq h11         (/ ic 2))
  (setq h2          (* 0.14175 cp))
  (setq h22         (* 0.2835 cp))
  (setq h3          (* 2 h2))
  (setq h4          (/ h3 ic))
  (setq h5          (* 2 oc))
  (setq h6          (* h1 1.2))
  (setq h7          (*(* h6 h6)2))
  (setq h8          (sqrt h7))
  (setq p2          (list (car p1)           (+ (cadr p1) h1)))
  (setq p21         (list (car p1)           (+ (cadr p1) h11)))
  (setq p3          (list (+ (car p2) h2)      (cadr p2)))
  (setq p4          (list (+ (car p21) h22) (cadr p21)))

```

```

(setq p5          (list (+ (car p4) h3)           (- (cadr p4) h4)))
      (list (+ (car p4) h3)           (- (cadr p4) h4)))
      (polar p1 5.497787144 h8))
      (polar p1 2.35619449 h8))

      (list (* (car p1) 1)           (* (cadr p1) 1)))
      (list (* (car p1)1000)         (* (cadr p1)1000)))
      (command "zoom" "w" w1 w2)
)

```

;Drawing

```

(defun draw ()
;;;      (command "circle" p1 "d" oc)
;;;      (command "circle" p1 "d" ic)
;;;      (command "circle" p1 "d" pc)
      (command "pline" p2 p3 p4 ""))
      (setq s1 (entlast))
      (command "mirror" "L" "" p1 p2 "n" "")
      (setq s2 (entlast))
      (command "pedit" "L" "J" s1 s2 "")
      (setq s21(entlast))
      (command "array" "L" "" "p" p1 n "" "Y")
      (setvar           osmode 0)
      (setvar "osmode" 1)
      (command "pline" p4 p5 "")
      (setvar "osmode" 0)
      (command "array" "L" "" "p" p1 n "" "Y")
      (setq s3 (entlast))
      (command "zoom" "e")
      (command "_view" "_top")
      (command "pedit" s21 "J" "C" z1 z2 "")
      (command "extrude" "L" "" gh "")
      (command ".region" "l")
      (command "circle" p1 "d" d1)
      (command "extrude" "L" "" gh "")
      (command ".region" "l")
      (setq pt1_list (list p2 p3 p4 p5))
      (command ".subtract" (ssget "_f" pt1_list) "L")
)

```

```
(setq p6 (polar p1 (dtr 90.0) (/ d1 8)))
(setq p7 (polar p6 (dtr 180.0) (/ d1 1.75)))
(setq p8 (polar p7 (dtr 270.0) (/ d1 4)))
(setq p9 (polar p8 (dtr 0.0) (/ d1 1.75)))
(command "pline" p1 p6 p7 p8 p9 "c")
(command "extrude" "L" "" gh "")
(command ".subtract" (ssget "_f" pt1_list)"" "L" "")
(command "_view" "_swiso" "")
(command "hide")
)
```

```
;Major program
(defun c:gear ()
  (input)
  (calculating1)
  (calculating2)
  (error_messages)
  (draw)
  (alert "Gear Created!")
(princ)
)      ;(prompt "\nType Gear to start the Lisp")
```

```
;;
;      s = entities
;      p = punten
;      w = zoom
;      z = selectie voor pedit final
;      h = help
```

