ภาควิชาวิทยาแพทย์โภชนาการ คณะหลักสูตรการศึกษาครู
(defun IsWhat (tmp1)
    (setq enType (cdr (assoc '0 (cdr (entget tmp1)))))
    (LAYER_ (cdr (assoc '8 (cdr (entget tmp1)))));;defun
(defun IsLne (tmp1)
    (setq tmp2 (cdr (entget tmp1)))
    (setq pStt (cdr (assoc '10 tmp2)) pEnd (cdr (assoc '11 tmp2)))
    );;defun
(defun IsArc (tmp1)
    (setq tmp2 (cdr (entget tmp1)))
    (setq pCen (cdr (assoc '10 tmp2))
        EndAng (cdr (assoc '51 tmp2))
        radi_ (cdr (assoc '40 tmp2))
        St_Ang (cdr (assoc '50 tmp2))
        pStt (polar pCen St_Ang radi_)
        pEnd (polar pCen EndAng radi_);;defun
(defun M&RotPt (tmp5)
    (setq tmp4 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) angRot) (distance c_tdPt tmp5)))
    (defun Ro&MPt (tmp5)
        (setq tmp3 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) angRot) pi) (distance c_tdPt tmp5)))
    (setq tmp6 (polar tmp4 (* pi 0.5) tmp3)))
(defun Ro&M_2 (tmp5)
    (setq tmp3 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) angRot) pi) (distance c_tdPt tmp5)))
    (setq tmp6 (polar tmp4 (* pi 0.5) tmp3))
    (defun Ro&M_3 (tmp5 tmp7)
        (setq tmp3 (polar '(0.0 0.0) (+ (angle c_tdPt tmp5) angRot) pi) (distance c_tdPt tmp5)))
        (setq tmp6 (polar tmp4 (* pi 0.5) tmp3))
        (setq tmp8 (polar tmp6 0.0 tmp7)))
    (defun inLYoN (tmp1 tmp2 tmp3)
        (setq tmp4 (abs (- (cadr tmp2) (cadr tmp1)))
            (if (equal tmp4 0 preci_)
                (setq disNow 0.0)
                (if (equal tmp4 (+ (abs (- (cadr tmp3) (cadr tmp1))) (abs (- (cadr tmp2) (cadr tmp3))))
                    preci_)
                    (progn
                        (setq disNow (abs (+ (+ (/ (* (- (cadr tmp3) (cadr tmp1)) (- (cadr tmp2) (cadr tmp1)))
                                (car tmp2) (car tmp1)))
                                (car tmp1) (car tmp3))))
                    )
                    (progn
                        (setq disNow 0.0) );; if
                    )
                )
                (if (> disNow disMem) (setq disMem disNow pMem0 tmp3))
            )
        )
        (defun in_L_N2 (tmp1 tmp2 tmp3)
            (setq tmp4 (abs (- (cadr tmp2) (cadr tmp1)))
                (if (equal tmp4 0 preci_)
                    (if (equal (cadr tmp3) (cadr tmp1) preci_)
                        (setq tmp5 (/ (+ (car tmp1) (car tmp2)) 2)
                                    inLne 1)
                        (setq inLne 0))
                    (if (equal tmp4 (+ (abs (- (cadr tmp3) (cadr tmp1))) (abs (- (cadr tmp2) (cadr tmp3))))
                        preci_)
                        (setq inLne 0)
                    )
                )
            )
        )
(setq tmp5 (+ (/ (* (- (car tmp3) (car tmp1)) (- (car tmp2) (car tmp1))) (- (car tmp5) (car tmp1)))
  (inLine 1))
(setq inLine 0)) ; if
);(if (equal tmp4 0 preci_)
)
(defun inAYoN (tmp1 tmp2 tmp3 tmp4)
(if (= tmp4 nil)
(setq InArc 0)
(progn
  (setq tmp5 (- (angle tmp1 tmp3) (angle tmp1 tmp2)))
  (if (<= tmp5 0) (setq tmp5 (+ tmp5 (* 2 pi))))
  (setq tmp6 (- (angle tmp1 tmp4) (angle tmp1 tmp2)))
  (if (< tmp6 0) (setq tmp6 (+ tmp6 (* 2 pi))))
  (if (> tmp6 tmp5) (setq InArc 0) (setq InArc 1))
  );progn
);if
);;defun
(defun XinCir (tmp1 tmp2 tmp3)
(setq tmp4 (car tmp1) tmp5 (cdr tmp1))
(setq tmp6 (- (* tmp2 tmp2) (* (- tmp3 tmp5) (- tmp3 tmp5))))
(if (> tmp6 0)
(progn
  (setq tmp7 (- tmp4 (sqrt tmp6)))
  (setq tmp8 (+ tmp4 (sqrt tmp6)))
  (setq PinCir 1)
)
(setq PinCir 0))
)

(defun BA2 ()
(command "UNDO" "m")
(setq osVar (getvar "OSMODE"))(setqvar "OSMODE" 0)
(command "EXPLODE" "all" "")
(princ "\n Select an object in the right side of all, please.")
(setq e2 (ssget))
(setq e1 (ssname e2 0))
(IsWhat e1)
(cond ((= enType "LINE")
  (IsLne e1)
  (if (> (cadr pStt) (cadr pEnd)) (setq direct 2) (setq direct 1))
  (= enType "ARC")
  (IsArc e1)
  (setq pS_L1 (polar pStt (angle pStt pEnd) (/ (distance pStt pEnd) 2)))
  (setq pS_L2 (polar pCen (angle pCen pS_L1) radi_))
  (inAYoN pCen pStt pEnd pS_L2)
  (if (= InArc 0) (setq pS_L2 (polar pCen (+ (angle pCen pS_L1) pi) radi_-))
    (if (> (car pS_L1) (car pS_L2)) (setq direct 2) (setq direct 1))
    );;cond
  (command "PEDIT" e1 "y" "j" "all" "" "")
  (setq areaS (getvar "AREA"))(command "EXPLODE" e1 "")
  (setq c_tdp nil)
  (graphsrc)
  (setq e1 (entnext))
)
(setq sumXbar 0 sumYbar 0 sumELn 0)
(setq preci_ 0.000001)
(while e1
  (setq en_Type (cdr (assoc '0 (entget e1))))
  (cond ((= en_Type "LINE")
    (IsLne e1)
    (setq X_bar (/ (+ (car pStt) (car pEnd)) 2)
     Y_bar (/ (+ (cdr pStt) (cdr pEnd)) 2))
    (setq E_Ln (distance pStt pEnd)))
  (setq cd_Lng (distance pStt pEnd))
  (if (> St_Ang EndAng)
    (setq a_Ang (- (* 2 pi) (~ St_Ang EndAng)))
    (setq a_Ang (~ EndAng St_Ang))))
  (cond ((= en_Type "ARC")
    (IsArc e1)
    (setq cd_Lng (distance pStt pEnd))
    (if (> St_Ang EndAng)
      (setq a_Ang (- (* 2 pi) (~ St_Ang EndAng)))
      (setq a_Ang (~ EndAng St_Ang)))
    (setq X_bar (+ (car pCen)(* radi_ (sin cd_Ag)))
     Y_bar (+ (cdr pCen)(* radi_ (cos cd_Ag))))
    (t nil))
  (setq sumXbar (+ sumXbar (* X_bar E_Ln))
      sumYbar (+ sumYbar (* Y_bar E_Ln)))
  (setq sumELn (+ sumELn E_Ln))
  (setq e1 (entnext e1))
  (while e1
    (setq X_Bar (/ sumXbar sumELn)
     Y_Bar (/ sumYbar sumELn))
    (setq c_tdPt (list X_Bar Y_Bar))
    (setvar "OSMODE" osVar)
  )
)
(defun c:2w ()
  (ba2)
  (setvar "CMDECHO" 0)
  (setq preci_ 0.00001)
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "move" "all" "" c_tdPt "0,0")
  (setq c_tdPt (list 0.0 0.0))
  (command "zoom" "e")
  (command "zoom" "0.3XP")
  (setq det_Ag 0.5)
  (setq angDeg 0)
  (setq a_Min nil a_Max 0.0 mxDiYoN 0 disNow 0.0)
  (princ "n")
  (princ " Please Wait. ")
  (princ "n")
  (while (<= angDeg 180)
    (setq angRot (* (/ pi 180) angDeg))
    (setq disMem 0.0 yMax 0.0 yMin 0.0)
    (setq m 0)
    (setq e1 (entnext))
    ;;STEP1
  )
  (while e1
    (if e1
      (......)
(progn
  (IsWhat e1) (setq enTy_1 enType)
  (cond ((= enTy_1 "LINE") (Islne e1) (M&RotPt pStt) (setq pS_L1 tmp4)
    (M&RotPt pEnd) (setq pE_L1 tmp4)
    (if (> (cadr pS_L1) (cadr pE_L1)) (setq tmp1 pS_L1 pS_L1 pS_L1 pE_L1
      pE_L1 tmp1))
    (if (> (cadr pE_L1) yMax) (setq yMax (cadr pE_L1) pYmax pE_L1
      eMax e1))
    (if (< (cadr pS_L1) yMin) (setq yMin (cadr pS_L1) pYmin pS_L1
      eMin e1))
    ((= enTy_1 "ARC") (IsArc e1) (setq ra_A1 radi_)
      (M&RotPt pCen) (setq pC_A1 tmp4)
      (M&RotPt pStt) (setq pS_A1 tmp4)
      (M&RotPt pEnd) (setq pE_A1 tmp4)
      (setq pChk (polar pC_A1 (* 0.5 pi) ra_A1))
      (inAYoN pC_A1 pS_A1 pE_A1 pChk)
      (if (= InArc 1)
        (if (> (cadr pChk) yMax) (setq yMax (cadr pChk) pYmax pChk
          eMax e1))
        (progn
          (if (> (cadr pS_A1) yMax) (setq yMax (cadr pS_A1) pYmax
              pS_A1 eMax e1))
          (if (> (cadr pE_A1) yMax) (setq yMax (cadr pE_A1) pYmax
              pE_A1 eMax e1))))
        (setq pChk (polar pC_A1 (* 1.5 pi) ra_A1))
        (inAYoN pC_A1 pS_A1 pE_A1 pChk)
        (if (= InArc 1)
          (if (< (cadr pChk) yMin)(setq yMin (cadr pChk) pYmin pChk eMin
              e1))
          (progn
            (if (< (cadr pS_A1) yMin) (setq yMin (cadr pS_A1) pYmin
                pS_A1 eMin e1))
            (if (< (cadr pE_A1) yMin) (setq yMin (cadr pE_A1) pYmin
                pE_A1 eMin e1))))
          );;;cond
          (setq e1 (entnext e1))
        );;; porgn
      );;; if e1
    )
  );;;STEP2
  (setq LstSuj nil e1 (entnext))
  (while e1
    (if e1
      (progn
        (IsWhat e1) (setq enTy_1 enType)
        (cond ((= enTy_1 "LINE")
          (Islne e1) (M&RotPt pStt) (setq pS_L1 tmp4)
          (M&RotPt pEnd) (setq pE_L1 tmp4)
          (if (or (and (equal pS_L1 pYmax precl_1) (equal pE_L1 pYmin preci_1))
              (and (equal pS_L1 pYmin precl_1) (equal pE_L1 pYmax preci_1)))
            (progn
              (setq pChk (list (/ (+ (car pS_L1) (car pE_L1)) 2) (/ (+ (cadr pS_L1) (cadr pE_L1))
                2))
              (setq tmp1 (append (list '0) (list pS_L1) (list pChk)))
              (setq tmp2 (append (list '0) (list pChk) (list pE_L1)))
            )
            (progn
              (setq pChk (list (/ (+ (car pS_L1) (car pE_L1)) 2) (/ (+ (cadr pS_L1) (cadr pE_L1))
                2))
              (setq tmp1 (append (list '0) (list pS_L1) (list pChk)))
              (setq tmp2 (append (list '0) (list pChk) (list pE_L1)))
            )
          )
        )
      )
    )
  )
)
(setq tmp3 nil)
)
(progn
  (setq tmp1 (append (list '0) (list pS_L1) (list pE_L1)))
  (setq tmp2 nil tmp3 nil)
)
)

(= enTy_1 "ARC")
(isArc e1) (setq ra_A1 radi_)
(m&rotpt pCen) (setq p_c_A1 tmp4)
(m&rotpt pStt) (setq pS_A1 tmp4)
(m&rotpt pEnd) (setq pE_A1 tmp4)
(setq pChk (polar p_c_A1 (* 0.5 pi) ra_A1))
(if (equal pChk pYmax preci_)
  (progn
    (inAYoN p_c_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (if (equal pS_A1 pChk preci_) (setq pCut1 nil)
       (if (equal pE_A1 pChk preci_) (setq pCut1 nil)
        (setq pCut1 pChk)))
     (setq pCut1 nil))
  (setq pCut1 nil))
);if
(setq pChk (polar p_c_A1 (* 1.5 pi) ra_A1))
(if (equal pChk pYmin preci_)
  (progn
    (inAYoN p_c_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (if (equal pS_A1 pChk preci_) (setq pCut2 nil)
       (if (equal pE_A1 pChk preci_) (setq pCut2 nil)
        (setq pCut2 pChk)))
     (setq pCut2 nil))
  (setq pCut2 nil))
);if
(if pCut1
  (if pCut2
    (if (> (angle p_c_A1 pS_A1) (* 0.5 pi)) (< (angle p_c_A1 pS_A1) (* 1.5 pi)))
     (if (= direct 1)
       (progn
         (setq tmp1 (append (list p_c_A1) (list pS_A1) (list pCut2)))
         (setq tmp2 (append (list p_c_A1) (list pCut2) (list pCut1)))
         (setq tmp3 (append (list p_c_A1) (list pCut1) (list pE_A1)))
       (progn
         (setq tmp1 (append (list p_c_A1) (list pCut1) (list pE_A1)))
         (setq tmp2 (append (list p_c_A1) (list pCut2) (list pCut1)))
         (setq tmp3 (append (list p_c_A1) (list pS_A1) (list pCut2)))
         );if
     (if (= direct 1)
       (progn
         (setq tmp1 (append (list p_c_A1) (list pS_A1) (list pCut1)))
         (setq tmp2 (append (list p_c_A1) (list pCut2) (list pCut1)))
         (setq tmp3 (append (list p_c_A1) (list pCut2) (list pE_A1)))
       (progn
         (setq tmp1 (append (list p_c_A1) (list pS_A1) (list pCut1)))
         (setq tmp2 (append (list p_c_A1) (list pCut2) (list pCut1)))
         (setq tmp3 (append (list p_c_A1) (list pCut2) (list pE_A1)))
       )
     )
   )
  )
);if

(setq tmp1 (append (list pC_A1) (list pCut2) (list pE_A1)))
(setq tmp2 (append (list pC_A1) (list pCut1) (list pCut2)))
(setq tmp3 (append (list pC_A1) (list pS_A1) (list pCut1)))) );if
)
(if (= direct 1)
(progn
 (setq tmp1 (append (list pC_A1) (list pS_A1) (list pCut1)))
 (setq tmp2 (append (list pC_A1) (list pCut1) (list pE_A1)))
 (setq tmp3 nil))
(progn
 (setq tmp1 (append (list pC_A1) (list pCut1) (list pE_A1)))
 (setq tmp2 (append (list pC_A1) (list pS_A1) (list pCut1)))
 (setq tmp3 nil))
)
);if pCut2
(if (= direct 1)
(progn
 (setq tmp1 (append (list pC_A1) (list pS_A1) (list pCut2)))
 (setq tmp2 (append (list pC_A1) (list pCut2) (list pE_A1)))
 (setq tmp3 nil))
(progn
 (setq tmp1 (append (list pC_A1) (list pCut2) (list pE_A1)))
 (setq tmp2 (append (list pC_A1) (list pS_A1) (list pCut2)))
 (setq tmp3 nil))
)
(progn
 (setq tmp1 (append (list pC_A1) (list pS_A1) (list pE_A1)))
 (setq tmp2 nil) (setq tmp3 nil))
);if pCut2
; if pCut1
)
);cond
(cond ((and (= tmp3 nil) (= tmp2 nil))
 (if LstSuj (setq LstSuj (append LstSuj (list tmp1))) (setq LstSuj (list tmp1))) )
((= tmp3 nil)
 (if LstSuj (setq LstSuj (append LstSuj (list tmp1) (list tmp2))) (setq LstSuj (append (list tmp1) (list tmp2))) )
(t (if LstSuj (setq LstSuj (append LstSuj (list tmp1) (list tmp2) (list tmp3))) (setq LstSuj
 (append (list tmp1) (list tmp2) (list tmp3)))) )
)
)
(setq e1 (entnext e1))
);;;; porgn
);;;; if e1
)

(setq n 0)
(setq nMax (length LstSuj))
(setq Lstpnil nil)
(setq step 0 swit 0 flag nil)
(while (< n nMax)
 (setq tmp0 (nth n LstSuj))
 (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
 (if (or (and (equal tmp2 pYmax preci_) (equal tmp3 pYmin preci_))
 (and (equal tmp2 pYmin preci_) (equal tmp3 pYmax preci_)) )
(progn

(setq p_C_A1 tmp1 p_S_A1 tmp2 p_E_A1 tmp3)
(setq ra_A1 (distance tmp1 tmp2))
(setq pChk (polar p_S_A1 (angle p_C_A1 p_E_A1) (/ (distance p_S_A1 p_E_A1) 2)))
(setq p_M_A1 (polar p_C_A1 (angle p_C_A1 pChk) ra_A1))
(if (= direct 1)
  (progn
    (setq tmLst1 (append (list p_C_A1) (list p_S_A1) (list p_M_A1)))
    (setq tmLst2 (append (list p_C_A1) (list p_M_A1) (list p_E_A1)))))
(progn
  (setq tmLst1 (append (list p_C_A1) (list p_M_A1) (list p_E_A1)))
  (setq tmLst2 (append (list p_C_A1) (list p_S_A1) (list p_M_A1))))
)
(setq tmLst1 tmp0 tmLst2 nil)
)
(if LstTmp
  (if (= tmLst2 nil) (setq LstTmp (append LstTmp (list tmLst1)))
    (setq LstTmp (append LstTmp (list tmLst1) (list tmLst2)))))
(if (= tmLst2 nil) (setq LstTmp (list tmLst1))
  (setq LstTmp (append (list tmLst1) (list tmLst2)))))
(setq n (+ n 1))
)
;while
(setq lstMem LstTmp)
(setq LstSuj LstTmp)
(setq n 0)
(setq nMax (length LstSuj))
(setq LstTmp nil LstLt nil LstRgh nil LstL_V nil LstR_V nil)
(setq step 0 swit 0 flag nil)
(while (< n nMax)
  (setq tmp0 (nth n LstSuj))
  (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
  (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_) 
    (equal tmp2 pYmin preci_) (equal tmp3 pYmin preci_))
    (progn
      (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_)) 
        (setq MxOrMn 1) (setq MxOrMn 0))
      (setq tmp4 (nth (+ n 1) LstSuj))
    )
  (if tmp4
    (progn
      (setq tmp5 (cadr tmp4) tmp6 (caddr tmp4))
      (if (or (equal tmp5 pYmax preci_) (equal tmp6 pYmax preci_) 
        (equal tmp5 pYmin preci_)) (equal tmp6 pYmin preci_))
      (if (or (equal tmp5 pYmax preci_) (equal tmp6 pYmax preci_))
        (if (= MxOrMn 1)
          (if (= swit 2) (setq swit 1) (setq swit 2))
          (setq swit 1))
        (if (= MxOrMn 0)
          (if (= swit 2) (setq swit 1) (setq swit 2))
          (setq swit 1))
      )
    )
  )
)
(setq swit 1)
)
(progn
  (if (= swit 2) (setq swit 1) (setq swit 2))
  )
(if (= swit 1)
  (progn
    (if (or (equal tmp2 pYmax preci_) (equal tmp3 pYmax preci_))
        ;true
        (if (= direct 1)
            (progn
              (if (= step 3) (setq flag 31))
              (setq step 1))
            (progn
              (if (= step 4) (setq flag 42))
              (setq step 2))
          );if
          (if (= direct 1)
            (progn
              (if (= step 1) (setq flag 13))
              (setq step 3))
            (progn
              (if (= step 2) (setq flag 24))
              (setq step 4))
          );if
        );if
      );progn
  );progn
);if

(cond ((= step 0) (if (LstTmp (setq LstTmp (append LstTmp (list tmp0)))) (setq LstTmp (list tmp0))))
  ((= step 1) (if (LstLft (setq LstLft (append LstLft (list tmp0)))) (setq LstLft (list tmp0))))
  ((= step 2) (if (LstRgh (setq LstRgh (append LstRgh (list tmp0)))) (setq LstRgh (list tmp0))))
  ((= step 3) (if (LstR_V (setq LstR_V (append LstR_V (list tmp0)))) (setq LstR_V (list tmp0))))
  ((= step 4) (if (LstL_V (setq LstL_V (append LstL_V (list tmp0)))) (setq LstL_V (list(tmp0))))
  (setq n (+ n 1))
);while

(cond ((= flag 31) (if (and LstLft LstTmp) (setq LstLft (append LstLft LstTmp))))
  ((= flag 24) (if (and LstL_V LstTmp) (setq LstL_V (append LstL_V LstTmp))))
  ((= flag 13) (if (and LstR_V LstTmp) (setq LstR_V (append LstR_V LstTmp))))
  ((= flag 42) (if (and LstRgh LstTmp) (setq LstRgh (append LstRgh LstTmp))))
  (if (and LstR_V (= LstRgh nil)) (setq LstRgh (reverse LstR_V))
  (if (and LstL_V (= LstLft nil)) (setq LstLft (reverse LstL_V))
  (if (and LstRgh (= LstR_V nil)) (setq LstR_V (reverse LstRgh))
  (if (and LstLft (= LstL_V nil)) (setq LstL_V (reverse LstLft))
  (setq n 0)
  (setq nMax (length LstRgh))
  (setq disMem 0.0)
  (while (< n nMax)
    (setq tmp0 (nth n LstRgh))
    (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
    (if (> (= (cadr tmp1) nil) (setq enTy_1 "LINE") (setq enTy_1 "ARC"))
      (cond (= enTy_1 "LINE") (if (> (cadr tmp2) (cadr tmp3)) (setq pS_L1 tmp3 pE_L1 tmp2)
        (setq pS_L1 tmp2 pE_L1 tmp3))
      (setq ra_A1 (distance tmp1 tmp2)) (setq pC_A1 tmp1) (setq pS_A1 tmp2) (setq pE_A1 tmp3))
(setq n2 0)
(while (< n2 nMax)
  (setq tmp0 (nth n2 LstR_V))
  (setq tmp1 (car tmp0))
  (if (= (cadr tmp1) nil)  (setq enTy_2 "LINE") (setq enTy_2 "ARC") )
  (cond ( (= enTy_2 "LINE")
           (RoM_2 (cadr tmp0)) (setq tmp1 tmp6)
           (RoM_2 (caddr tmp0)) (setq tmp2 tmp6)
           (if (> (cadr tmp1) (cadr tmp2)) (setq pS_L2 tmp2 pE_L2 tmp1) (setq pS_L2 tmp1 pE_L2 tmp2))
         )
  (= enTy_2 "ARC")
  (RoM_2 (cadr tmp0)) (setq pC_A2 tmp6)
  (RoM_2 (caddr tmp0)) (setq pS_A2 tmp6)
  (RoM_2 (caddr tmp0)) (setq pE_A2 tmp6)
  (setq ra_A2 (distance pC_A2 pS_A2))
  )
  (cond ((and (= enTy_1 "LINE") (= enTy_2 "LINE")
           (if (or (equal (cadr pE_L1) (cadr pS_L2) preci_1) (equal (cadr pE_L2) (cadr pS_L1) preci_1))
             (setq xxx 1)
             (if (or (and (>=(cadr pE_L1) (cadr pS_L2)) <= (cadr pS_L1) (cadr pE_L2))
                     (and (>= (cadr pE_L2) (cadr pS_L1)) <= (cadr pS_L2) (cadr pE_L1)))
               (progn
                 (inLYoN pS_L1 pE_L1 pS_L2)
                 (inLYoN pS_L1 pE_L1 pE_L2)
                 (inLYoN pS_L2 pE_L2 pS_L1)
                 (inLYoN pS_L2 pE_L2 pE_L1)
               )
               (progn
                 (or (and (= enTy_1 "LINE") (= enTy_2 "ARC")
                       (and (= enTy_1 "ARC") (= enTy_2 "LINE")
                        (if (= enTy_1 "ARC") (setq pS_L1 pS_L2 pE_L1 pE_L2
                            pS_A2 pS_A1 pE_A2 pE_A1 pC_A2 pC_A1 ra_A2 ra_A1)
                          (setq pChk (polar pC_A2 (+ (angle pS_L1 pE_L1)(* 0.5 pi)) ra_A2))
                          (inAYoN pC_A2 pS_A2 pE_A2 pChk)
                          (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
                          (setq pChk (polar pC_A2 (+ (angle pE_L1 pS_L1)(* 0.5 pi)) ra_A2))
                          (inAYoN pC_A2 pS_A2 pE_A2 pChk)
                          (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
                          (XinCr pC_A2 ra_A2 (cadr pS_L1))
                          (if (= PinCr 1)
                            (progn
                              (setq pChk (list tmp7 (cadr pS_L1)))
                              (inAYoN pC_A2 pS_A2 pE_A2 pChk)
                              (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
                              (setq pChk (list tmp8 (cadr pS_L1))
                              (inAYoN pC_A2 pS_A2 pE_A2 pChk)
                              (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
                            ))
                          (XinCr pC_A2 ra_A2 (cadr pE_L1))
                          (if (= PinCr 1)
                            (progn
                              (setq pChk (list tmp7 (cadr pE_L1)))
                              (inAYoN pC_A2 pS_A2 pE_A2 pChk)
                            ))
                        )))
                      )))
        )))
))
(if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
(setq pChk (list tmp8 (cadr pE_L1)))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
)
(inLYoN pS_L1 pE_L1 pS_A2)
(inLYoN pS_L1 pE_L1 pE_A2)
)
((and (= enTy_1 "ARC") (= enTy_2 "ARC"))
(setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
(setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
(if (<= C_CinY (+ ra_A1 ra_A2))
(progn
(setq tmp2 (/ C_CinY (+ 1 (/ ra_A2 ra_A1))))
(if (> (cadr pC_A1) (cadr pC_A2))
(setq Y_Bar (- (cadr pC_A1) tmp2))
(setq Y_Bar (+ (cadr pC_A1) tmp2)))
(XinCir pC_A1 ra_A1 Y_Bar)
(setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
(XinCir pC_A2 ra_A2 Y_Bar)
(setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
(inAYoN pC_A1 pS_A1 pE_A1 pCut1)
(if (= InArc 1)
(progn
(inAYoN pC_A2 pS_A2 pE_A2 pCut4)
(if (= InArc 1)
(progn
(setq disNow (abs (- (car pCut1)(car pCut4)))))
(if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
(setq mxDiYoN 1)
))
}
)
(inAYoN pC_A1 pS_A1 pE_A1 pCut2)
(if (= InArc 1)
(progn
(inAYoN pC_A2 pS_A2 pE_A2 pCut3)
(if (= InArc 1)
(progn
(setq disNow (abs (- (car pCut2)(car pCut3)))))
(if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
(setq mxDiYoN 1)
))
))
)
)
)
(progn
(setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
(setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
(setq tmp1 (abs (- ra_A2 ra_A1))
(if (<= C_CinY tmp1)
(progn
(setq tmp2 (/ (* ra_A1 (abs (- (cadr pC_A2)(cadr pC_A1)))) tmp1))
(if (<= ra_A1 ra_A2)
(if(< (cadr pC_A1)(cadr pC_A2))(setq Y_Bar(-(cadr pC_A1)tmp2))(setq Y_Bar(+ (cadr pC_A1)tmp2)))
    (if(< (cadr pC_A2)(cadr pC_A1))(setq Y.Bar(-(cadr pC_A1)tmp2))(setq Y.Bar(+ (cadr pC_A1)tmp2)))
)
(XinCir pC_A1 ra_A1 Y.Bar)
(setq pCut1 (list tmp7 Y.Bar) pCut2 (list tmp8 Y.Bar))
(XinCir pC_A2 ra_A2 Y.Bar)
(setq pCut3 (list tmp7 Y.Bar) pCut4 (list tmp8 Y.Bar))
(inAYoN pC_A1 pS_A1 pE_A1 pCut1)
(if (= InArc 1)
    (progn
        (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
        (if (= InArc 1)
            (progn
                (setq disNow (abs (- (car pCut1)(car pCut3)))
                (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
                (setq mxDiYoN 1)
            )
        )
    )
)(inAYoN pC_A1 pS_A1 pE_A1 pCut2)
(if (= InArc 1)
    (progn
        (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
        (if (= InArc 1)
            (progn
                (setq disNow (abs (- (car pCut2)(car pCut4)))
                (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
                (setq mxDiYoN 1)
            )
        )
    )
)(if (= mxDiYoN 0)
    (progn
        (XinCir pC_A2 ra_A2 (cadr pS_A1))
    (if (= PinCir 1)
        (progn
            (setq pChk (list tmp7 (cadr pS_A1)))
            (inAYoN pC_A2 pS_A2 pE_A2 pChk)
            (if (= InArc 1)
                (progn
                    (setq disNow (abs - tmp7 (car pS_A1)))
                    (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
                )
            )
        (setq pChk (list tmp8 (cadr pS_A1)))
        (inAYoN pC_A2 pS_A2 pE_A2 pChk)
        (if (= InArc 1)
            (progn
                (setq disNow (abs - tmp8 (car pS_A1)))
                (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
            )
        )
    )
)(XinCir pC_A2 ra_A2 (cadr pE_A1))
(if (= PinCir 1)
(progn
  (setq pChk (list tmp7 (cadr pE_A1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)
    (progn
      (setq disNow (abs (- tmp7 (car pE_A1))))
      (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
    )
  )
  (setq pChk (list tmp8 (cadr pE_A1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)
    (progn
      (setq disNow (abs (- tmp8 (car pE_A1))))
      (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
    )
  )
)(XinCir pC_A1 ra_A1 (cadr pS_A2))
(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (progn
        (setq disNow (abs (- tmp7 (car pS_A2))))
        (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
      )
    )
    (setq pChk (list tmp8 (cadr pS_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (progn
        (setq disNow (abs (- tmp8 (car pS_A2))))
        (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
      )
    )
)(XinCir pC_A1 ra_A1 (cadr pE_A2))
(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pE_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (progn
        (setq disNow (abs (- tmp7 (car pE_A2))))
        (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
      )
    )
    (setq pChk (list tmp8 (cadr pE_A2)))
    (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
      (progn
        (setq disNow (abs (- tmp8 (car pE_A2))))
        (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
      )
    )
  )
)(progn
  (setq mxDiYoN 0)
);(if (= mxDiYoN 0)
(setq n2 (+ n2 1))
) ;while
(setq n (+ n 1))
)
(setq d_toShf disMem)
(setq n 0)
(setq nMax (length LstLft))
(setq disMem 0.0)
(while (< n nMax)
 (setq tmp0 (nth n LstLft))
 (setq tmp1 (car tmp0) tmp2 (cadr tmp0) tmp3 (caddr tmp0))
 (if (= (cadr tmp1) nil) (setq enTy_1 "LINE") (setq enTy_1 "ARC") )
 (cond ((= enTy_1 "LINE") (if (> (cadr tmp2) (cadr tmp3)) (setq pS_L1 tmp3 pE_L1 tmp2)
 (setq pS_L1 tmp2 pE_L1 tmp3)) )
 ((= enTy_1 "ARC") (setq ra_A1 (distance tmp1 tmp2)) (setq pC_A1 tmp1) (setq pS_A1 tmp2) (setq pE_A1 tmp3))
 ) ;cond
(setq n2 0)
(while(< n2 nMax)
 (setq tmp0 (nth n2 LstL_V))
 (setq tmp1 (car tmp0))
 (if (= (cadr tmp1) nil) (setq enTy_2 "LINE") (setq enTy_2 "ARC") )
 (cond ((= enTy_2 "LINE")
 (Ro8M_3 (cadr tmp0) d_toShf) (setq tmp1 tmp8)
 (Ro8M_3 (cadr tmp0) d_toShf) (setq tmp2 tmp8)
 (if (> (cadr tmp1) (cadr tmp2)) (setq pS_L2 tmp2 pE_L2 tmp1) (setq pS_L2 tmp1 pE_L2 tmp2))
((= enTy_2 "ARC")
 (Ro8M_3 (cadr tmp0) d_toShf) (setq pC_A2 tmp8)
 (Ro8M_3 (cadr tmp0) d_toShf) (setq pS_A2 tmp8)
 (Ro8M_3 (cadr tmp0) d_toShf) (setq pE_A2 tmp8)
 (setq ra_A2 (distance pC_A2 pS_A2)) ) ) ;cond
 (cond ((and (= enTy_1 "LINE") (= enTy_2 "LINE")
 (if (or (equal (cadr pE_L1) (cadr pS_L2) preci_) (equal (cadr pE_L2) (cadr pS_L1)
 preci_)))
 (setq xxx 1)
 (if (or (and (>=(cadr pE_L1) (cadr pS_L2)) (<=(cadr pS_L1) (cadr pE_L2)))
 (and (>=(cadr pE_L2) (cadr pS_L1)) (<=(cadr pS_L2) (cadr pE_L1))))
 (progn
 (inLYoN pS_L1 pE_L1 pS_L2)
 (inLYoN pS_L1 pE_L1 pE_L2)
 (inLYoN pS_L2 pE_L2 pS_L1)
 (inLYoN pS_L2 pE_L2 pE_L1) ) ;progn
 )
 )
 )
(or (and (= enTy_1 "LINE") (= enTy_2 "ARC")
 (and (= enTy_1 "ARC") (= enTy_2 "LINE")
 (if (= enTy_1 "ARC") (setq pS_L1 pS_L2 pE_L1 pE_L2
 pS_A2 pS_A1 pE_A2 pE_A1 pC_A2 pC_A1 ra_A2 ra_A1)
 (setq pChk (polar pC_A2 (+ (angle pS_L1 pE_L1)(* 0.5 pi)) ra_A2))
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
)
(setq pChk (polar pC_A2 (+ (angle pE_L1 pS_L1)(* 0.5 pi)) ra_A2))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
(XinCir pC_A2 ra_A2 (cadr pS_L1))
(if (= PinCir 1))
(progn
  (setq pChk (list tmp7 (cadr pS_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  (setq pChk (list tmp8 (cadr pS_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
)
(XinCir pC_A2 ra_A2 (cadr pE_L1))
(if (= PinCir 1))
(progn
  (setq pChk (list tmp7 (cadr pE_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  (setq pChk (list tmp8 (cadr pE_L1)))
  (inAYoN pC_A2 pS_A2 pE_A2 pChk)
  (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
)
(inLYoN pS_L1 pE_L1 pS_A2)
(inLYoN pS_L1 pE_L1 pE_A2)
(and (= enTy_1 "ARC") (= enTy_2 "ARC")
  (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2)))))
  (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
  (if (<= C_CinY (+ ra_A1 ra_A2)))
(progn
  (setq tmp2 (/ C_CinY (+ 1 (/ ra_A2 ra_A1))))
  (if (> (cadr pC_A1) (cadr pC_A2))
    (setq Y_Bar (- (cadr pC_A1) tmp2))
    (setq Y_Bar (+ (cadr pC_A1) tmp2)))
  (XinCir pC_A1 ra_A1 Y_Bar)
  (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
  (XinCir pC_A2 ra_A2 Y_Bar)
  (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
  (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
  (if (= InArc 1))
  (progn
    (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
    (if (= InArc 1)
      (progn
        (setq disNow (abs (- (car pCut1)(car pCut4))))
        (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
        (setq mxDiYOY 1)
      )
    ))
  (inAYoN pC_A1 pS_A1 pE_A1 pCut2)
  (if (= InArc 1)
    (progn
      (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
      (if (= InArc 1)

(progn
  (setq disNow (abs (- (car pCut2)(car pCut3))))
  (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
  (setq mxDiYoN 1)
)
)
)
)
(if (= mxDiYoN 0)
(progn
  (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
  (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
  (setq tmp1 (abs (- ra_A2 ra_A1)))
  (if (<= C_CinY tmp1)
    (progn
      (setq tmp2 (/ (* ra_A1 (abs (- (cadr pC_A2)(cadr pC_A1)))) tmp1))
      (if (= ra_A1 ra_A2)
        (if (< (cadr pC_A1)(cadr pC_A2))(setq Y_Bar(+ (cadr pC_A1)tmp2)))
          (setq Y_Bar(- (cadr pC_A1)tmp2)))
          (setq Y_Bar(+ (cadr pC_A1)tmp2)))
      (XinCir pC_A1 ra_A1 Y_Bar)
      (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
      (XinCir pC_A2 ra_A2 Y_Bar)
      (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
      (inAYOn pC_A1 pS_A1 pE_A1 pCut1)
      (if (= InArc 1)
        (progn
          (inAYOn pC_A2 pS_A2 pE_A2 pCut3)
          (if (= InArc 1)
            (progn
              (setq disNow (abs (- (car pCut1)(car pCut3))))
              (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
              (setq mxDiYoN 1)
            ))
          ))
      (inAYOn pC_A1 pS_A1 pE_A1 pCut2)
      (if (= InArc 1)
        (progn
          (inAYOn pC_A2 pS_A2 pE_A2 pCut4)
          (if (= InArc 1)
            (progn
              (setq disNow (abs (- (car pCut2)(car pCut4))))
              (if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
              (setq mxDiYoN 1)
            ))
          ))
      ))
    (if (= mxDiYoN 0)
      (progn
        (XinCir pC_A2 ra_A2 (cadr pS_A1))
        (if (= PinCir 1)
          (progn
            ))
(setq pChk (list tmp7 (cdr pS_A1)))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)
 (progn
   (setq disNow (abs (- tmp7 (car pS_A1))))
   (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
   ))

(setq pChk (list tmp8 (cdr pS_A1)))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)
 (progn
   (setq disNow (abs (- tmp8 (car pS_A1))))
   (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
   ))

(XinCir pC_A2 ra_A2 (cdr pE_A1))
(if (= PinCir 1)
 (progn
   (setq pChk (list tmp7 (cdr pE_A1))))
   (inAYoN pC_A2 pS_A2 pE_A2 pChk)
   (if (= InArc 1)
     (progn
       (setq disNow (abs (- tmp7 (car pE_A1))))
       (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
       ))
   (setq pChk (list tmp8 (cdr pE_A1)))
   (inAYoN pC_A2 pS_A2 pE_A2 pChk)
   (if (= InArc 1)
     (progn
       (setq disNow (abs (- tmp8 (car pE_A1))))
       (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
       ))

(XinCir pC_A1 ra_A1 (cdr pS_A2))
(if (= PinCir 1)
 (progn
   (setq pChk (list tmp7 (cdr pS_A2))))
   (inAYoN pC_A1 pS_A1 pE_A1 pChk)
   (if (= InArc 1)
     (progn
       (setq disNow (abs (- tmp7 (car pS_A2))))
       (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
       ))
   (setq pChk (list tmp8 (cdr pS_A2)))
   (inAYoN pC_A1 pS_A1 pE_A1 pChk)
   (if (= InArc 1)
     (progn
       (setq disNow (abs (- tmp8 (car pS_A2))))
       (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
       ))

(XinCir pC_A1 ra_A1 (cdr pE_A2))
(if (= PinCir 1)
 (progn
   (setq pChk (list tmp7 (cdr pE_A2))))
   (inAYoN pC_A1 pS_A1 pE_A1 pChk)
   (if (= InArc 1)
     (progn
       (setq disNow (abs (- tmp7 (car pE_A2))))
       (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
       ))
(inAYoN pC_A1 pS_A1 pE_A1 pChk)
(if (= InArc 1)
  (progn
    (setq disNow (abs (- tmp7 (car pE_A2))))
    (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
  ))
(setq pChk (list tmp8 (cadr pE_A2)))
(inAYoN pC_A1 pS_A1 pE_A1 pChk)
(if (= InArc 1)
  (progn
    (setq disNow (abs (- tmp8 (car pE_A2))))
    (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
  ))
);;progn
(setq mxDiYoN 0)
); (if (= mxDiYoN 0)
  ;;progn
  ;;if
  ;)
  
(setq n2 (+ n2 1))
(if (> nMax 4)
  (progn
    (setq XX (rem n2 4))
    (cond ((= XX 0) (princ "\n") (princ "\n") (princ "\n Please wait. / ") (princ "\n")
          (princ "\n "))
          ((= XX 1) (princ "\n") (princ "\n") (princ "\n Please wait. - ") (princ "\n") (princ "\n")
          (princ "\n "))
          ((= XX 2) (princ "\n") (princ "\n") (princ "\n Please wait. \| \| ") (princ "\n") (princ "\n")
          (princ "\n "))
          ((= XX 3) (princ "\n") (princ "\n") (princ "\n Please wait. | ") (princ "\n") (princ "\n")
          (princ "\n ")))
  )
);while n2
(setq n (+ n 1))
(princ "\n")
);while n
(setq d_Area disMem)
(setq hgh (abs (- yMax yMin)))
(setq areaN_ (* d_Area hgh))
(if (or (< areaN_ a_Min) (= a_Min nil)) (setq a_Min areaN_ agM_Min d_Area d_ShMm d_toShf))

yMinMm yMin

(dsM Мин disMem pM_Min pMm0 yMaxMm yMax

(setq angDeg (+ angDeg det_Ag))
) ; while (<= angRot
(princ "\n") (princ "\n") (princ "\n Please wait. * ")(princ "\n ")(princ "\n")
(setq angMeD (/ (* agM_Min 180) pi))
(setq bstAg1 angMeD)
(setvar "OSMODE" osVar)
(princ "\n") (princ "\n")
(princ "\n") (princ " Press any key to continue, please."
(princ "\n") (princ "\n")
(read-char)

(tes2w())

(defun tes2w ()
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (setq gap (getreal "\n GAP <0> = "))
  (if (= gap nil) (setq gap 0))
  (setq agM_MinD (/ (* agM_Min 180) pi))
  (setq dsMarry (* (- 1)(+ (* 2 gap) d_AreaMm)))
  (setq tmp3 (+ yMaxMm yMinMm))
  (setq tmp2 (polar (0 0) (* pi 0.5) tmp3))
  (setq tmp4 (+ gap d_ShfMm))
  (setq tmp1 (polar tmp2 0 tmp4))
  (graphscr)
  (command "EXPLODE" "all" "")
  (command "EDIT" "l" "y" "j" "all" """)
  (command "ROTATE" "all" "0,0" agM_MinD)
  (command "COPY" "p" "0,0" "0,0")
  (command "ROTATE" "p" "0,0" "180")
  (command "MOVE" "p" "0,0" tmp1)
  (command "ARRAY" "all" "R" "1" "3" dsMarry)
  (setvar "OSMODE" osVar)
  (princ "(n)" (princ "")
  (princ "(n)" (princ " Press any key to see data.")
  (princ "(n)")
  (read-char)
  (read-char)
  (read-char)
  (setq tmp1 (/ (* areaS2 a_Min) 100))
  (setq tmp2 (- yMaxMm yMinMm))
  (textscr)
  (princ "(n)" (princ "**************************************************************************")
  (princ "(n)" (princ "**************************************************************************")
  (princ "(n)" (princ "")
  (princ "(n)" (princ " This is the data for 2 way strip layout.")
  (princ "(n)" (princ "")
  (princ "(n)" (princ " The best angle = ") (princ agM_MinD) (princ " degree.")
  (princ "(n)" (princ " Distance for feed = ") (princ dsM_Min) (princ " mm. <no gap>")
  (princ "(n)" (princ " Strip width = ") (princ tmp2) (princ " mm. <no gap>")
  (princ "(n)" (princ " The best Area = ") (princ a_Min) (princ ",<2 workpiece>")
  (princ "(n)" (princ " The efficiency = ") (princ tmp1) (princ " % <no gap>")
  (princ "(n)" (princ "")
  (princ "(n)" (princ "**************************************************************************")
  (princ "(n)" (princ "** If the function is not correct or **")
  (princ "(n)" (princ "** have some problem, please tell me. ***")
  (princ "(n)" (princ "** ***")
  (princ "(n)" (princ " S.Singhapant ***")
  (princ "(n)" (princ " Tel 074322901 073917164 ***")
  (princ "(n)" (princ "**************************************************************************")
  (princ "(n)" (princ "")
  (princ "(n)" (princ " Press any key to continue,please.")
  (princ "(n)" (princ "")
  (read-char)
  (graphscr))
(defun b1 ()
  (command "UNDO" "m")
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "EXPLODE" "all" "m")
  (setq e1 (entnext))(command "EDIT" e1 "y" "j" "all" "m" "m")
  (setq e1 (entlast))(command "list" e1 "m")
  (setq areaS (getvar "AREA"))(command "EXPLODE" e1 "m")
  (setq c_tdPt nil)
  (graphscr)
  (setq e1 (entnext))
  (setq sumXbar 0 sumYbar 0 sumELn 0)
  (while e1
    (setq en_Type (cdr (assoc '0 (entget e1))))
    (cond ((= en_Type "LINE")
      (IsLine e1)
      (setq X_bar (/ (+ (car pStt) (car pEnd)) 2)
            Y_bar (/ (+ (cadr pStt) (cadr pEnd)) 2))
      (setq E_Ln (distance pStt pEnd))
      ((= en_Type "ARC")
        (IsArc e1)
        (setq cd_Lng (distance pStt pEnd))
        (if (> St_Ang EndAng)
          (setq a_Ang (- (* 2 pi) (- St_Ang EndAng)))
          (setq a_Ang (- EndAng St_Ang)) )
        (setq E_Ln (* radi_a_Ang))
        (setq a_Ang2 (/ a_Ang 2))
        (setq X_b1a (/ (* radi_ (sin a_Ang2)) a_Ang2)
            cd_Ag (- (angle pStt pEnd) (* pi 0.5))
            (setq X_bar (+ (car pCen)(* radi_ (cos cd_Ag)))
            Y_bar (+ (cadr pCen)(* radi_ (sin cd_Ag)))))
    (t nil) )cond
    (setq sumXbar (+ sumXbar (* X_bar E_Ln))
          sumYbar (+ sumYbar (* Y_bar E_Ln)))
    (setq sumELn (+ sumELn (* E_Ln E_Ln)))
    (setq e1 (entnext e1)) ;while
    (setq X_Bar (/ sumXbar sumELn)
          Y_Bar (/ sumYbar sumELn))
    (setq c_tdPt (list X_Bar Y_Bar))
    (setvar "OSMODE" osVar)
    )

(defun c1w ()
  (ba1)
  (setq preci_ 0.00001)
  (setq osVar (getvar "OSMODE"))(setvar "OSMODE" 0)
  (command "move" "all" "m" c_tdPt "0,0")
  (setq c_tdPt (list 0.0 0.0))
  (command "zoom" "e")
  (command "zoom" "0.3XP")
  (setq det_Ag 1.0)
  (setq angDeg 0)
  (setq a_Min nil a_Max 0.0 mxDiYoN 0 disNow 0.0)
  (setq Lstang nil)
  (setq n 1)
  (while (<= angDeg 180)
    (setq angRot (*/ (pi 180) angDeg))
    )
(setq disMem 0.0 yMax 0.0 yMin 0.0)
(setq e1 (entnext))
(while e1
 (if e1
   (progn
     (setq e2 e1)
     (IsWhat e1) (setq enTy_1 enType)
     (cond ((= enTy_1 "LINE") (progn
                               (IsLne e1) (M&RotPt pStt) (setq pS_L1 tmp4)
                               (M&RotPt pEnd) (setq pE_L1 tmp4)
                               (if (> (cdr pS_L1) (cdr pE_L1)) (setq tmp1 pS_L1 pS_L1 pE_L1 pE_L1 tmp1))
                               (if (> (cdr pE_L1) yMax) (setq yMax (cdr pE_L1)))
                               (if (< (cdr pS_L1) yMin) (setq yMin (cdr pS_L1))))
                               (if (> (cdr pE_L1) yMax) (setq yMax (cdr pE_L1)))))
     (if (> (cdr pChk) yMax) (setq yMax (cdr pChk)))
     (progn
       (if (> (cdr pS_A1) yMax) (setq yMax (cdr pS_A1)))
       (if (> (cdr pE_A1) yMax) (setq yMax (cdr pE_A1))))
     (setq pChk (polar pC_A1 (* 0.5 pi) ra_A1))
     (inAYoN pC_A1 pS_A1 pE_A1 pChk)
     (if (= InArc 1)
       (if (> (cdr pChk) yMax) (setq yMax (cdr pChk)))
       (progn
         (if (> (cdr pS_A1) yMax) (setq yMax (cdr pS_A1)))
         (if (> (cdr pE_A1) yMax) (setq yMax (cdr pE_A1))))
       (setq pChk (polar pC_A1 (* 1.5 pi) ra_A1))
       (inAYoN pC_A1 pS_A1 pE_A1 pChk)
       (if (= InArc 1)
         (if (< (cdr pChk) yMin) (setq yMin (cdr pChk)))
         (progn
           (if (< (cdr pS_A1) yMin) (setq yMin (cdr pS_A1)))
           (if (< (cdr pE_A1) yMin) (setq yMin (cdr pE_A1))))
         (setq pChk (polar pC_A1 (* -0.0 pi) ra_A1))
         (inAYoN pC_A1 pS_A1 pE_A1 pChk)
         (if (= InArc 1)
           (progn
             (setq pChk (polar pC_A1 pi ra_A1))
             (inAYoN pC_A1 pS_A1 pE_A1 pChk)
             (if (= InArc 1)
               (progn
                 (setq disNow (* ra_A1 2))
                 (if (> disNow disMem) (setq disMem disNow pMem0 disChk))))))
   (setq e2 (entnext 2))
   (if e2
     (progn
     (IsWhat e2) (setq enTy_2 enType)
     (cond ((= enTy_2 "LINE")
               (progn
                 (IsLne e2) (M&RotPt pStt) (setq pS_L2 tmp4)
                 (M&RotPt pEnd) (setq pE_L2 tmp4)
                 (if (> (cdr pS_L2) (cdr pE_L2)) (setq tmp1 pS_L2 pS_L2 pE_L2 pE_L2 tmp1))
                 (if (> (cdr pE_L2) yMax) (setq yMax (cdr pE_L2)))))
     (setq e2 (entnext e2))
   )
  ))
(if (< (cadr pS_L2) yMin) (setq yMin (cadr pS_L2)))
)

((= enTy_2 "ARC")
 (progn
 (IsArc e2)(setq ra_A2 radi_)
 (M&RotPt pCen)(setq pC_A2 tmp4)
 (M&RotPt pStt)(setq pS_A2 tmp4)
 (M&RotPt pEnd)(setq pE_A2 tmp4)
 (setq pChk (polar pC_A2 (* 0.5 pi) ra_A2))
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)
 (if (> (cadr pChk) yMax)(setq yMax(cadr pChk)))
 (progn
 (if (> (cadr pS_A2) yMax)(setq yMax(cadr pS_A2)))
 (if (> (cadr pE_A2) yMax)(setq yMax(cadr pE_A2))))
 (setq pChk (polar pC_A2 (* 1.5 pi) ra_A2))
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)
 (if (< (cadr pChk) yMin)(setq yMin(cadr pChk)))
 (progn
 (if (< (cadr pS_A2) yMin)(setq yMin(cadr pS_A2)))
 (if (< (cadr pE_A2) yMin)(setq yMin(cadr pE_A2))))
 (setq pChk (polar pC_A2 0 ra_A2))
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)
 (progn
 (setq pChk (polar pC_A2 pi ra_A2))
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)
 (progn
 (setq disNow (* ra_A2 2))
 (if (> disNow disMem) (setq disMem disNow pMem0 pChk)))
))))

((= enTy_2 "CIRCLE") (setq XXX nil)) ; ; ; ; cond
 (cond ((and (= enTy_1 "LINE") (= enTy_2 "LINE")
 (if (or (and (>=(cadr pE_L1) (cadr pS_L2))(<= (cadr pS_L1) (cadr pE_L2)))
 (and (>=(cadr pE_L2) (cadr pS_L1))(<= (cadr pS_L2) (cadr pE_L1))))
 (progn
 (inLYoN pS_L1 pE_L1 pS_L2)
 (inLYoN pS_L1 pE_L1 pE_L2)
 (inLYoN pS_L2 pE_L2 pS_L1)
 (inLYoN pS_L2 pE_L2 pE_L1)
 )
 )))
 (or (and (= enTy_1 "LINE") (= enTy_2 "ARC")
 (and (= enTy_1 "ARC") (= enTy_2 "LINE")))
 (progn
 (if (= enTy_1 "ARC")
 (setq pS_L1 pS_L2 pE_L1 pE_L2)
 pS_A2 pS_A1 pE_A2 pE_A1 pC_A2 pC_A1 ra_A2 ra_A1))
 (setq pChk (polar pC_A2 (+ (angle pS_L1 pE_L1)(* 0.5 pi)) ra_A2)
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk)
 (setq pChk (polar pC_A2 (+ (angle pE_L1 pS_L1)(* 0.5 pi)) ra_A2))
 (inAYoN pC_A2 pS_A2 pE_A2 pChk)
 (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
 (XinCir pC_A2 ra_A2 (cadr pS_L1)))

(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pS_L1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
    (setq pChk (list tmp8 (cadr pS_L1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  )
)
(XinCir pC_A2 ra_A2 (cadr pE_L1))
(if (= PinCir 1)
  (progn
    (setq pChk (list tmp7 (cadr pE_L1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
    (setq pChk (list tmp8 (cadr pE_L1)))
    (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)(inLYoN pS_L1 pE_L1 pChk))
  )
)
(inLYoN pS_L1 pE_L1 pS_A2)
(inLYoN pS_L1 pE_L1 pE_A2)
)

((and (= enTy_1 "ARC") (= enTy_2 "ARC"))
 (progn
    (setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2))))
    (setq C_CinX (abs (- (car pC_A1) (car pC_A2))))
    (if (<= C_CinY (+ ra_A1 ra_A2))
      (progn
        (setq tmp2 (/ C_CinY (+ 1 (/ ra_A2 ra_A1))))
        (if (> (cadr pC_A1) (cadr pC_A2))
          (setq Y_Bar (- (cadr pC_A1) tmp2))
          (setq Y_Bar (+ (cadr pC_A1) tmp2)))
        (XinCir pC_A1 ra_A1 Y_Bar)
        (setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
        (XinCir pC_A2 ra_A2 Y_Bar)
        (setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
        (inAYoN pC_A1 pS_A1 pE_A1 pCut1)
        (if (= InArc 1)
          (progn
            (inAYoN pC_A2 pS_A2 pE_A2 pCut4)
            (if (= InArc 1)
              (progn
                (setq disNow (abs (- (car pCut1)(car pCut4)))))
                (if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
                (setq mxDiYoN 1)
              ))
            )
          )
        )
      )
    )
  )
)
(inAYoN pC_A1 pS_A1 pE_A1 pCut2)
(if (= InArc 1)
  (progn
    (inAYoN pC_A2 pS_A2 pE_A2 pCut3)
    (if (= InArc 1)
      (progn
        (setq disNow (abs (- (car pCut2)(car pCut3))))
      )
    )
  )
)
(if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
(setq mxDiYoN 1)
)
))
) ;;; progn
) ;;; if
(if (= mxDiYoN 0)
(progn
(setq C_CinY (abs (- (cadr pC_A1) (cadr pC_A2)))))
(setq C_CinX (abs (- (car pC_A1) (car pC_A2)))))
(setq tmp1 (abs (- ra_A2 ra_A1)))
(if (<= C_CinY tmp1)
(progn
(setq tmp2 (/ (* ra_A1 (abs (- (cadr pC_A2) (cadr pC_A1))))) tmp1))
(if (<= ra_A1 ra_A2)
(if (< (cadr pC_A1) (cadr pC_A2)) (setq Y_Bar (- (cadr pC_A1) tmp2))
(setq Y_Bar (+ (cadr pC_A1) tmp2))))
(if (< (cadr pC_A2) (cadr pC_A1)) (setq Y_Bar (- (cadr pC_A1) tmp2))
(setq Y_Bar (+ (cadr pC_A1) tmp2))))
)
(XinCir pC_A1 ra_A1 Y_Bar)
(setq pCut1 (list tmp7 Y_Bar) pCut2 (list tmp8 Y_Bar))
(XinCir pC_A2 ra_A2 Y_Bar)
(setq pCut3 (list tmp7 Y_Bar) pCut4 (list tmp8 Y_Bar))
(inAYoN pC_A1 pS_A1 pE_A1 pCut1)
(if (= InArc 1)
(progn
(inAYoN pC_A2 pS_A2 pE_A2 pCut3)
(if (= InArc 1)
(progn
(setq disNow (abs (- (car pCut1) (car pCut3))))
(if (> disNow disMem) (setq disMem disNow pMem0 pCut1))
(setq mxDiYoN 1)
))
)
)
(inAYoN pC_A1 pS_A1 pE_A1 pCut2)
(if (= InArc 1)
(progn
(inAYoN pC_A2 pS_A2 pE_A2 pCut4)
(if (= InArc 1)
(progn
(setq disNow (abs (- (car pCut2) (car pCut4))))
(if (> disNow disMem) (setq disMem disNow pMem0 pCut2))
(setq mxDiYoN 1)
))
))
)
)
(if (= mxDiYoN 0)
(progn
(XinCir pC_A2 ra_A2 (cadr pS_A1))
(if (= PinCir 1)
(progn
(setq pChk (list tmp7 (cadr pS_A1)))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)
    (progn
      (setq disNow (abs (- tmp7 (car pS_A1))))
      (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
    ))

(setq pChk (list tmp8 (cadr pS_A1)))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)
    (progn
      (setq disNow (abs (- tmp8 (car pS_A1))))
      (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
    ))

)(XinCir pC_A2 ra_A2 (cadr pE_A1))
(if (= PinCir 1)
    (progn
      (setq pChk (list tmp7 (cadr pE_A1))))
      (inAYoN pC_A2 pS_A2 pE_A2 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp7 (car pE_A1))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
        ))

(setq pChk (list tmp8 (cadr pE_A1)))
(inAYoN pC_A2 pS_A2 pE_A2 pChk)
(if (= InArc 1)
    (progn
      (setq disNow (abs (- tmp8 (car pE_A1))))
      (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
    ))

)(XinCir pC_A1 ra_A1 (cadr pS_A2))
(if (= PinCir 1)
    (progn
      (setq pChk (list tmp7 (cadr pS_A2))))
      (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp7 (car pS_A2))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
        ))

(setq pChk (list tmp8 (cadr pS_A2)))
(inAYoN pC_A1 pS_A1 pE_A1 pChk)
(if (= InArc 1)
    (progn
      (setq disNow (abs (- tmp8 (car pS_A2))))
      (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
    ))

)(XinCir pC_A1 ra_A1 (cadr pE_A2))
(if (= PinCir 1)
    (progn
      (setq pChk (list tmp7 (cadr pE_A2))))
      (inAYoN pC_A1 pS_A1 pE_A1 pChk)
    (if (= InArc 1)
        (progn
          (setq disNow (abs (- tmp7 (car pE_A2))))
          (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
        ))
(progn
  (setq disNow (abs (- tmp7 (car pE_A2))))
  (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
)
(setq pChk (list tmp8 (cadr pE_A2))
(inAYoN pC_A1 pS_A1 pE_A1 pChk)
(if (= InArc 1)
  (progn
    (setq disNow (abs (- tmp8 (car pE_A2))))
    (if (> disNow disMem) (setq disMem disNow pMem0 pChk))
  ))
) ;progn
(setq mxDiYoN 0)
(if (= mxDiYoN 0)
  ));progn

);cond
(if (> n 50)
  (progn
    (setq XX (rem n 4))
    (cond ((= XX 0) (princ "\n") (princ "\n") (princ "\n Please wait. / ") (princ "\n ")
          (princ "\n")
          (princ "\n")
          (princ "\n")
          (princ "\n")
          (princ "\n")
          (princ "\n")
          (princ "\n")
          (princ "\n "))
    )))
  ));progn
);if e2
);while e2
(setq e1 (entnext e1))
); ;; porgn
);; if e1
);;while e1
(setq hgh (abs (- yMax yMin)))
(setq areaN_ (* disMem hgh))
(if (or (< areaN_ a_Min)(= a_Min nil)) (setq a_Min areaN_ agM_Min angRot
 dsM_Min disMem pM_Min pMem0 yMaxMm yMax

  yMinMm yMin))
(setq angDeg (+ angDeg det_Ag))
); while (<= angRot 360)
(setq angMeD (/ (* agM_Min 180) pi))
(setq bstAg1 angMeD)
(setvar "OSMODE" osVar)
(princ "\n") (princ "\n")
(princ "\n") (princ " Press any key to continue, please. ")
(princ "\n") (princ "\n")
(read-char)
(tes) )