

# Kontrol deyimleri

- Kontrol deyimleri, programda tanımlanan işlem adımlarını ve akışı denetlemek, koşullara bağlı olarak deyimlerin/işlemlerin uygulanma sırasını değiştirmek amacıyla kullanılır.
- FORTRAN 77 dili kontrol deyimlerinin başlıcaları şunlardır:

1-Koşulsuz GOTO

2-Koşullu veya hesaplanmış GOTO

3-Aritmetik IF

4-Mantıksal IF

5-Döngü/tekrarlama deyimi DO

6- PAUSE, STOP ve END deyimleri.

# Koşulsuz GOTO Deyimi

- Genel yapısı **GOTO n** şeklinde olan bu deyim, program akışının **n** numaralı deyime gitmesini sağlar. Genellikle IF deyimi yakınlarında kullanılır.

```
      I=0
      ITOPL=0
5     READ (*,*) N
      IF (N.LE.0) GOTO 30
      I=I+1
      ITOPL=ITOPL+N
      GOTO 5
30    IORTL=ITOPL/I
      WRITE (*,*) 'Ortalama =', IORTL
      WRITE (*,*) 'Sayac = ', I
      STOP
      END
```

# Hesaplanmış GOTO Deyimi

Bu deyim bir tam sayı değişkeninin değerine göre programın kontrolünün, çok sayıda deyime aktarılmasını sağlar. Genel yapısı,

**GOTO ( n1, n2, ... , nr ), i**

şeklindedir. Burada  $i$ , bu deyimden önce saptanmış bir tam sayı değişkenidir,  $n$ 'ler ise deyim numaralarıdır.  $i = 1$  ise kontrol  $n1$ , numaralı deyime,  $i = 2$  ise kontrol  $n2$  numaralı deyime geçer,  $i$  tam sayısı için  $1 < i < r$  koşulu sağlanmalıdır,  $i > r$  veya  $i = 0$  için bir alttaki deyime geçilir. **GOTO** sözcüğünü izleyen parantez içinde en az iki deyim numarası bulunmalıdır.

# Hesaplanmış GOTO Deyimi

```
      READ (*, *) I  
      GOTO (10, 20, 30, 40), (I-10)  
10      I=I+1  
          GOTO 50  
20      I=I+2  
          GOTO 50  
30      I=I+3  
          GOTO 50  
40      I=I+4  
50      WRITE (*, *) I  
          END
```

# Aritmetik IF Deyimi

- Bu deyim, aritmetik bir işlem ya da deyim belirlendiği sonuca göre, programın olası 3 deyiminden birine sapmasını sağlar.
- Genel yapısı  $IF ( d ) n1, n2, n3$  şeklindedir. Burada  $d$  bir sabit, bir değişken veya aritmetik bir deyimdir,  $d$  ifadesi kompleks bir sayı veya mantıksal bir ifade olamaz.
- $d < 0$  ise program  $n1$ , numaralı deyime sapar,
- $d = 0$  ise program  $n2$  numaralı deyime sapar.
- $d > 0$  ise program  $n3$ , numaralı deyime sapar.
- Aritmetik IF deyiminde  $n1, n2, n3$  deyim numaraları mutlaka kullanılmalıdır; ancak bu numaraların hepsinin birbirinden farklı olması gerekmez. Aritmetik IF deyimini izleyen ilk deyim, mutlaka bir deyim numarası taşımaktadır.

# Aritmetik IF Deyimi

$x \leq 2.1$  için  $y = 0.5*x + 0.95$

$x > 2.1$  için  $y = 0.7*x + 0.53$

Bu amaçla yazılmış bir program parçası şu şekilde olabilir :

```
IF (X - 2.1) 30, 30, 40  
30    Y = 0.5*X + 0.95  
      GOTO 45  
40    Y = 0.7*X + 0.53  
45    ...
```

Böylece  $x - 2.1 \leq 0$  yani  $x \leq 2.1$  için 30 numaralı deyim işleme girer.  $x - 2.1 > 0$  veya  $x > 2.1$  için program 40 numaralı deyime sapar.

# Mantıksal IF Deyimi

Genel yapısı **IF ( p ) q** şeklindedir. Burada p, MANTIKSAL BİR DEYİM'dir (doğru/yanlış), q bir deyim/işlem'dir.

Eğer p ifadesinin sonucu olumlu ise q deyimi işleme girer; değilse bir alttaki satıra geçilir.

# Mantıksal bağlantı işlemcileri

| Mantıksal ifade |               | Fortran90/95 | Anlamı                        |
|-----------------|---------------|--------------|-------------------------------|
| .EQ.            | Equal         | $==$         | Eşittir ( $=$ )               |
| .NE.            | Not equal     | $\neq$       | Eşit değildir ( $\neq$ )      |
| .LT.            | Less than     | $<$          | Küçüktür ( $<$ )              |
| .LE.            | Less equal    | $\leq$       | Küçük veya eşittir ( $\leq$ ) |
| .GT.            | Greater than  | $>$          | Büyüktür ( $>$ )              |
| .GE.            | Greater equal | $\geq$       | Büyük veya eşittir ( $\geq$ ) |



# Mantıksal işlemciler

| Mantıksal işlemci | Anlamı |
|-------------------|--------|
| .AND.             | ve     |
| .OR.              | veya   |
| .NOT.             | değil  |

# Mantıksal işlemciler

| x | y | $x \text{ .AND .} y$ |
|---|---|----------------------|
| 1 | 0 | 0                    |
| 0 | 1 | 0                    |
| 0 | 0 | 0                    |
| 1 | 1 | 1                    |

| x | y | $x \text{ .OR .} y$ |
|---|---|---------------------|
| 1 | 0 | 1                   |
| 0 | 1 | 1                   |
| 1 | 1 | 1                   |
| 0 | 0 | 0                   |

| x | $\text{.NOT. } x$ |
|---|-------------------|
| 1 | 0                 |
| 0 | 1                 |

# Blok IF Deyimi

```
IF (ifade) THEN
```

```
.....
```

```
ELSE
```

```
.....
```

```
ENDIF
```

veya

```
IF (ifade1) THEN
```

```
.....
```

```
ELSEIF (ifade2)
```

```
.....
```

```
ELSEIF (ifade3)
```

```
.....
```

```
ELSE
```

```
.....
```

```
ENDIF
```

```
IF (ifade1) THEN
```

```
.....
```

```
IF (ifade2) THEN
```

```
.....
```

```
ENDIF
```

```
.....
```

```
ELSE
```

```
IF (ifade3) THEN
```

```
.....
```

```
ENDIF
```

```
.....
```

```
ENDIF
```

## Soru : İki bilinmeyenli denklem sisteminin çözümü

$$a \mathbf{x} + b \mathbf{y} = c$$

$$d \mathbf{x} + e \mathbf{y} = f$$

şeklinde iki bilinmeyenli bir denklem sistemi verilmektedir (x ve y bilinmeyenleri, a, b, c, d, e, f bilinen değerleri göstermektedir). Denklem sisteminin yani bilinmeyenlerin çözümü için;

$$\mathbf{x} = [(c \cdot e) - (b \cdot f)] / [(a \cdot e) - (b \cdot d)]$$

$$\mathbf{y} = [(a \cdot f) - (c \cdot d)] / [(a \cdot e) - (b \cdot d)]$$

ifadeleri elde edilebilir. Aşağıdaki denklem sistemi için bilinmeyenleri bulunuz.

$$0.2038 \mathbf{x} + 0.1218 \mathbf{y} = 0.2014$$

$$0.4071 \mathbf{x} + 0.2436 \mathbf{y} = 0.4038$$

denklemleri için x ve y değerlerini bulan Fortran Programını Yazınız.

# İki bilinmeyenli denklem sisteminin çözümü

|                   |   |
|-------------------|---|
| Başla             | Programın Yazımı :                              |
| A'yi tanıtır      | A=0.2038  |
| B'yi tanıtır      | B=0.1218  |
| D'yi tanıtır      | D=0.4071  |
| E'yi tanıtır      | E=0.2436  |
| C'yi tanıtır      | C=0.2014  |
| F'yi tanıtır      | F=0.4038  |
| X'i hesaplar.     | $X = ((C * E) - (B * F)) / ((A * E) - (B * D))$ |
| Y'yi hesaplar.    | $Y = ((A * F) - (C * D)) / ((A * E) - (B * D))$ |
| “Görünüm”         | WRITE(*,*)' <<< VERILEN DENKLEMİN KOKLERI>>>'   |
| “Görünüm”         | WRITE(*,*)' '                                   |
| X ve Y'yi yazdır. | WRITE(*,*)'X=',X,' VE ',Y=',Y,' DIR.'           |
| “Görünüm”         | WRITE(*,*)'=====',                              |
| “Görünüm”         | *'====='  |
| Dur               | STOP  |
| Son               | END   |

# çözüm

```
program prog
implicit none
real :: x, y, a, b, c, d, e, f, c1, f1
real :: payda
a=0.2038
b=0.1218
c=0.2014
d=0.4071
e=0.2436
f=0.4038
payda = ((a * e) - (b * d))
if(payda /= 0) goto 10
write(*,*)' payda sıfır olduğu için kökler bulunamıyor!'
stop
10  x = ((c * e) - (b * f)) / payda
    y = ((a * f) - (c * d)) / payda
    WRITE(*, 20) x, y
20  FORMAT(' kok degerleri', /, &
          'x=', 1X, F8.3, /, &
          'y=', 1X, F8.3)
    c1=a*x+b*y
    f1=d*x+e*y
    write(*, *)' elde edilen köklere göre eşitlikler '
    write(*, 30) c1, f1
30  FORMAT(' 1.eşitlik =', 1X, F8.3, /, &
          ' 2.eşitlik =', 1X, F8.3)
    STOP
end program prog
```

# Ekran görüntüsü

```
kok degerleri
x=  -2.000
y=   5.000
  elde edilen köklere göre eşitlikler
1.eşitlik =   0.201
2.eşitlik =   0.404
Warning: ieee_inexact is signaling
FORTRAN STOP
Press any key to continue . . .
```

# Döngüler/Tekrarlama

işlemlerde tekrarlama varsa yani veriler/bilgiler için aynı işlemler yapılacaksa bunun için işlem sayısı belli olan ve olmayan durumlar için çeşitli **döngüler** kullanılır. Bu kesimde bunlarla ilgili bilgiler verilecektir



# DO Döngüsü

Kontrol deyimlerinden olan DO deyimi, programın belirli bir kesimindeki döngü ya da döngülerin yapılmasını kolaylaştırıcı bir deyim olup, daha çok indisli değişkenlerle ilgili işlemlerde kullanılır. Genel yapısı,

```
DO n i=m1, m2          veya          DO n i=m1, m2, m3
```

şeklindedir. Burada n, DO döngüsünün son deyim numarasıdır. Bu son deyim IF, GOTO, STOP, RETURN veya başka bir DO deyimi olamaz. Genellikle burada CONTINUE deyimi kullanılır. i sayısı DO deyiminin KONTROL DEĞİŞKENİ olup daima bir tam sayı değişkeni olmalıdır, m1, m2 ve m3, parametreleri birer tam sayı sabiti veya değişkenidir. m3, kullanılmazsa değeri +1 olarak derleyici tarafından belirlenir.  $i \geq m2$ , olduğunda program n numaralı deyim altındaki satıra geçer.

# DO Döngüsü

```
DO 10 I = -5, -8, -1      Çıktı
10      WRITE (*,11) I    -5
11      FORMAT (4I5)      -6
END                        -7
                        -8
```

# DO Döngüsü

|    |                  |              |
|----|------------------|--------------|
|    | DO 10 I = -8, -5 | <u>Çıktı</u> |
|    | WRITE (*,11) I   | -8           |
| 11 | FORMAT (4I5)     | -7           |
| 10 | CONTINUE         | -6           |
|    | END              | -5           |

# WHILE Döngüsü

İşlem sayısı belli olmayan yani ne kadar döngü yapılacağı belli olmayan durumlarda koşula bağlı döngüler kullanılabilir. Koşul sağlandığı durumda döngüden çıkılarak işlem bitirilir.

| WATFOR derleyicisi                         | Fortran 90/95 derleyicisi                  |
|--|--|
| <pre>WHILE (koşul)   .   . END WHILE</pre> | <pre>DO WHILE (koşul)   .   . END DO</pre> |

# WHILE Döngüsü

[*isim:*] **DO** [*etiket*] *do-değişkeni* = alt sınır, üst sınır [ ,adım]  
[ *do bloğu* ]

[*etiket*] **END DO** veya **CONTINUE**

```
DO WHILE (cevap .NE. 'h')  
    WRITE (*, '(A)') ' e/h : '  
    READ (*, '(A)') cevap  
END DO
```

```
I=1  
WHILE (I.LT.10)  
    I=I+1  
END WHILE
```

# WHILE Döngüsü

```
C234567
```

```
C sinus fonksiyonu
```

```
C  $\sin(x) = x - x^3/3! + x^5/5! - \dots$ 
```

```
C serisi ile çözümlenmektedir
```

```
    X=3.14159/6.
```

```
C terim sayısı
```

```
    N=9
```

```
C faktöriyel hesabı için değişken
```

```
    K=1
```

```
C toplam değişkeni
```

```
    T=0
```

```
    J=1
```

```
    WHILE (K.LT.N)
```

```
        I=1
```

```
        IFAKT=1
```

```
        WHILE (I.LE.K)
```

```
            IFAKT=IFAKT*I
```

```
            I=I+1
```

```
        END WHILE
```

```
        T=T+J*X** (K) /FLOAT (IFAKT)
```

```
        K=K+2
```

```
        J=J* (-1)
```

```
    END WHILE
```

```
    WRITE (*,10) T, SIN (X)
```

```
10    FORMAT (1X,2F8.3)
```

```
    STOP
```

```
    END
```

# LOOP... UNTIL Döngüsü

İşlem sayısı belli olmayan yani ne kadar döngü yapılacağı belli olmayan durumlarda koşula bağlı döngülerden biriside LOOP UNTIL döngüsüdür. Mantıksal ifade olan koşul sağlandığı durumda döngüden çıkılarak işlem bitirilir (WATFOR-77).

LOOP

.

.

UNTIL (koşul)

# LOOP... UNTIL Döngüsü

```
C234567
C EXPONANSİYEL IN HESAPLANMASI
C  $e(x) = 1 + x/1! + x*x/2! + x*x*x/3! + \dots$ 
  X=1.5
  N=10
  K=1
  T=1.0
  LOOP
    I=1
    IFAKT=1
    LOOP
      IFAKT=IFAKT*I
      I=I+1
    UNTIL (I.GT.K)
    T=T+X**K/FLOAT(IFAKT)
    WRITE(*,*)T,K,IFAKT
    K=K+1
  UNTIL (K.GT.N)
  WRITE(*,10)T,EXP(X)
10  FORMAT(1X,2F8.3)
  STOP
  END
```

```
program prog
  implicit none
! EXPONANSİYEL IN HESAPLANMASI
!  $e(x) = 1 + x/1! + x*x/2! + x*x*x/3! + \dots$ 
  integer :: i, n=10, k=1, faktoriyel
  real :: x, t=1.0
  x=1.5
  n=10
  k=1
  do while (k.LE.n)
    i=1
    faktoriyel=1
    do while (i <= k)
      faktoriyel=faktoriyel*i
      i=i+1
    end do
    t = t + x**k/FLOAT(faktoriyel)
    write(*,*)t, k, faktoriyel
    k = k+1
  end do
  write(*,10)t, EXP(x)
  format(1X,2F8.3)
  STOP
end program prog
```



# Çarpım tablosunun hazırlanması

1. Başla
2.  $I=1$
3. Eğer  $I>10$  ise Git 8
4.  $J=2*I$
5. Yaz 2, 'x', I, '=', J
6.  $I=I+1$
7. Git 3
8. Son

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

# Çarpım tablosu

C234567

C234567

I=1

LOOP

I1=1\*I

WRITE(\*,\*) I, '\*1=', I1

I=I+1

UNTIL (I.GT.10)

STOP

END

I=1

DO WHILE (I.LE.10)

I1=1\*I

WRITE(\*,\*) I, '\*1=', I1

I=I+1

END DO

STOP

END

1\*1= 1

2\*1= 2

3\*1= 3

4\*1= 4

5\*1= 5

6\*1= 6

7\*1= 7

8\*1= 8

9\*1= 9

10\*1= 10

# KOŞULLU DÖNGÜLER

$m=8.40$  kg lık bir cisim sürtünmesiz eğik bir düzlem üzerinden kaymaktadır. Eğik düzlemin yatayla yaptığı açığı  $0^\circ$  den  $90^\circ$  ye kadar  $5$  er derece artırarak cisim üzerine etki eden kuvveti ( $n=m*g*\cos?$ ) ve cismin ivmesini ( $a=g*\sin?$ ) do ... while döngüsünü kullanarak hesaplanması:

1. Başla
2.  $teta=0.0$
3.  $m=8.40$
4.  $g=10.0$
5. Eğer  $teta>90$  ise Git 11
6.  $n=m*g*\cos(teta*3.14/180.0)$
7.  $a=g*\sin(teta*3.14/180.0)$
8. yaz  $teta, n, a$
9.  $teta=teta+5.0$
10. Git 5
11. Son

```
PROGRAM EgikDuzlem
IMPLICIT NONE
REAL :: teta=0.0, n, a
REAL :: m=8.40, g=10.0
WRITE(*,*) ' teta(o)   n(Newton)   a(n/s^2) '
WRITE(*,*) ' -----   -----   ----- '
DO WHILE (teta<=90)
    n = m*g*cos(teta*3.14/180)
    a = g*sin(teta*3.14/180)
    WRITE(*,10) teta, n, a
10 FORMAT(1x, F6.1, 2X, F9.2, 2X, F8.2)
    teta=teta+5
END DO
END PROGRAM EgikDuzlem
```

| teta (o) | n (Newton) | a (n/s^2) |
|----------|------------|-----------|
| -----    | -----      | -----     |
| 0.0      | 84.00      | 0.00      |
| 5.0      | 83.68      | 0.87      |
| 10.0     | 82.73      | 1.74      |

# SELECT CASE Deyimi

sorgulamada birden fazla olası durum karşısında kullanılacak deyimdir.

```
[case-adı : ] SELECT CASE [case-ifadesi]  
  CASE [case-seçici]  
    blok  
  [CASE DEFAULT  
    blok]  
END SELECT [case-adı]
```

# SELECT CASE Deyimi

```
SELECT (koşul) FROM
CASE durum1
.
.
CASE durum2
.
.
.
END SELECT
```

```
READ(*,*) I
SELECT (I) FROM
CASE 1
    CALL SUB1()
CASE 2
    CALL SUB2()
CASE DEFAULT
END SELECT
```

```
SELECT CASE (NET_GELIR)
CASE (50000:)
    VERGI_ORANI=0.28
CASE (25000:49999)
    VERGI_ORANI=0.14
CASE DEFAULT
    VERGI_ORANI_RATE=0.05
END SELECT
```

# SELECT CASE

```
program prog
implicit none
CHARACTER*1  Sec
CHARACTER*44 :: Sec1, Sec2, Sec3, Sec4
Sec1 = "A. Avrupa turu, 5 ülke, 5 gece, 6 gündüz.      "
Sec2 = "B. Afrika turu, 8 ülke, 7 gece, 8 gündüz.      "
Sec3 = "C. Uzak doğu turu, 3 ülke, 5, gece, 6 gündüz."
Sec4 = "D. Amerika turu, 2 ülke, 15, gece, 16 gündüz."
WRITE (*, *) Sec1
WRITE (*, *) Sec2
WRITE (*, *) Sec3
WRITE (*, *) Sec4
WRITE (*, *) "-----"
WRITE (*, *) " SEÇENEĞİNİZ"
READ  (*, *) Sec
secenek : SELECT CASE (Sec)
CASE ('A', 'a')
    WRITE (*, *) Sec1, " 999€ "
CASE ('B', 'b')
    WRITE (*, *) Sec2, " 1299 US Dolar"
CASE ('C', 'c')
    WRITE (*, *) Sec3, " 2599 US Dolar "
CASE ('D', 'd')
    WRITE (*, *) Sec4, " 1799 US dolar"
CASE DEFAULT
    WRITE (*, *) "Cevabınız seçenekler dışındadır."
END SELECT Secenek
end program prog
```

```
A. Avrupa turu, 5 3lke, 5 gece, 6 g3nd3z.
B. Afrika turu, 8 3lke, 7 gece, 8 g3nd3z.
C. Uzak dou turu, 3 3lke, 5, gece, 6 g3zd3z
D. Amerika turu, 2 3lke, 15, gece, 16 g3zd3z
```

```
-----
SEÄENE°|N|Z
```

```
c
```

```
C. Uzak dou turu, 3 3lke, 5, gece, 6 g3zd3z 2599 US Dolar
Press any key to continue . . .
```



# Sorular

1. Bir sayının (A) karekökünü hesaplamak için kullanılan yöntemlerden birisi  $R_{i+1} = (R_i^2 + A) / (2R_i)$  denklemi ile verilebilir. Denklemdaki  $R_i$  karekök değeri için ilk tahmini değerdir ve genellikle A'nın yarısı olarak alınır. Yukarıdaki denklemi kullanarak A'nın karekökünü hesaplayan algoritma ve fortran programını yazınız.

# Sorular

Bir sayının karekökünün  $R_{i+1} = (R_i^2 + A) / (2R_i)$  ye göre hesaplanması.

1. Başla
2. A değişkenine değer giriniz,
3.  $R = A / 2.0$
4.  $I = 0$
5. Eğer  $I > 20$  ise Git 9
6.  $R = (R * R + A) / (2 * R)$
7.  $I = I + 1$
8. Git 5
9. R değişkenindeki değeri yaz,
10. Son

```
program prog
  implicit none
  real :: A, R
  integer :: I=0
  write(*, *) 'karekökü hesaplanacak sayı:'
  read(*, *) A
  R=A/2.0
5  if (I>20) goto 10
  R=(R*R+A)/(2*R)
  I=I+1
  go to 5
10 write(*, 15) A, R
15 format(1X, F6.2, ' nın karekökü =', F6.2)
end program prog
```

```
karekökü hesaplanacak sayı:
```

```
7
```

```
7.00 nın karekökü = 2.65
```

```
Press any key to continue . . .
```



# Sorular

2. Aşağıdaki verileri bir dosyadan okuyup yazıcıya yollayan FORTRAN programını yazınız.

```
X=0.1234567E3      I=158
Y=0.4682E0        J=303
Z=0.876543E-1     K=5
```

```
program prog
implicit none
real A, X, Y, Z
integer I, J, K, L
open(1, FILE='veri.txt', status='old')
do
  read(1, 10, end=15) A, L
10  format (e12.7, i3)
  X=A
  I=L
  write(*, 13) X, I
13  format(1x, e12.7, 2x, i3)
end do
15  close(1)
end program prog
```

```
.1234567E+03  158
.4682000E+00  303
.8765430E-01   5
Press any key to continue . . .
```

# Sorular

3. Bir elipsin alanı ve çevresinin uzunluğu eksenlerinin yarı uzunlukları olan  $a$  ve  $b$  değerlerine bağlı olarak Alan =  $\pi * a * b$ , Çevre =  $2 * \pi * \text{sqr}t((a^2+b^2)/2)$  şeklinde verilebilir. Bu denklemlere göre elipsin eksenlerinin yarı uzunlukları girilince/verilince elipsin alanını ve çevre uzunluğunu hesaplayan FORTRAN programını yazınız.

1. Başla

2.  $a=10.0$

3.  $b=8.0$

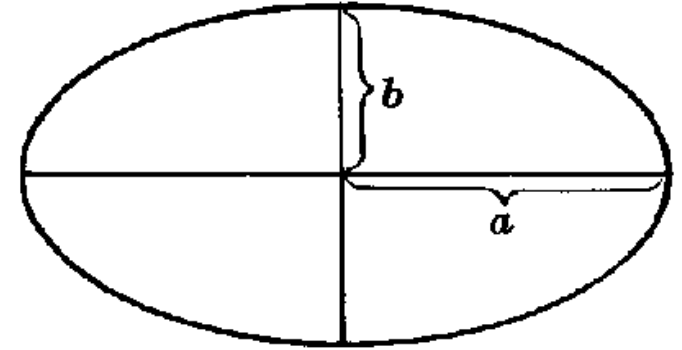
4.  $PI=3.14159$

5. Alan= $\pi*a*b$

6. Çevre= $2*\pi*\text{sqr}t((a^2+b^2)/2)$

7.  $a$ ,  $b$ , Alan ve Çevre değişkenlerindeki değerleri yaz.

9. Son



$$\text{Alan} = \pi ab \quad k = \sqrt{a^2 - b^2}/a$$

$$\begin{aligned} \text{Çevre} &= 4a \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \theta} d\theta \\ &= 2\pi \sqrt{\frac{1}{2}(a^2 + b^2)} \quad (\text{yaklaşık olarak}) \end{aligned}$$

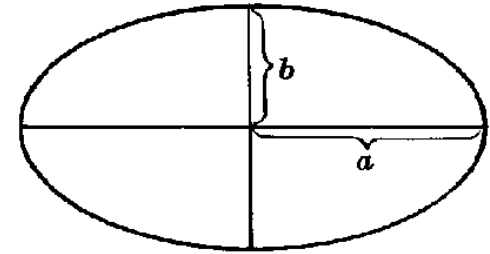
# Sorular

```
program Elips
implicit none
real :: a=10.0, b=8.0, PI=3.14159
real :: Alan, Cevre
Alan=pi*a*b
Cevre=2*pi*sqrt((a**2+b**2)/2)
print *, ' Elipsin köşegen uzunlukları = ', a, b
print *, ' Elipsin çevresi = ', Cevre
print *, ' Elipsin alanı = ', Alan
stop
end program Elips
```

<https://ideone.com>

Success #stdin #stdout 0s 14912KB

```
Elipsin köşegen uzunlukları = 10.0000000 8.00000000
Elipsin çevresi = 56.8966179
Elipsin alanı = 251.327209
```



$$\text{Alan} = \pi ab \quad k = \sqrt{a^2 - b^2}/a$$

$$\begin{aligned} \text{Çevre} &= 4a \int_0^{\pi/2} \sqrt{1 - k^2 \sin^2 \theta} d\theta \\ &= 2\pi \sqrt{\frac{1}{2}(a^2 + b^2)} \quad (\text{yaklaşık olarak}) \end{aligned}$$

# Sorular

4. Derece, dakika ve saniye cinsinden girilen bir açı değerini derece, radyan ve gradyene çeviren FORTRAN programını yazınız.

```
ACI=DER+DAK/60.0+SAN/3600.0
```

```
RADYAN=ACI*3.141593/180.0
```

```
GRAD=ACI*400.0/360.0
```

```
program AciHesabi
```

```
implicit none
```

```
real :: aci, der, dak, san
```

```
real :: rad, gra
```

```
read (*, *) der, dak, san
```

```
aci=der+dak/60.0+san/3600.0
```

```
rad=aci*3.141593/180.0
```

```
gra=aci*400.0/360.0
```

```
print *, aci, ' derece ', rad, ' radyandır.
```

```
'
```

```
print *, aci, ' derece ', gra, ' grandyendir.'
```

```
grandyendir.'
```

```
stop
```

```
end program AciHesabi
```

Success #stdin #stdout 0s 14912KB

45.3488884

derece

0.791487455

radyandır.

45.3488884

derece

50.3876534

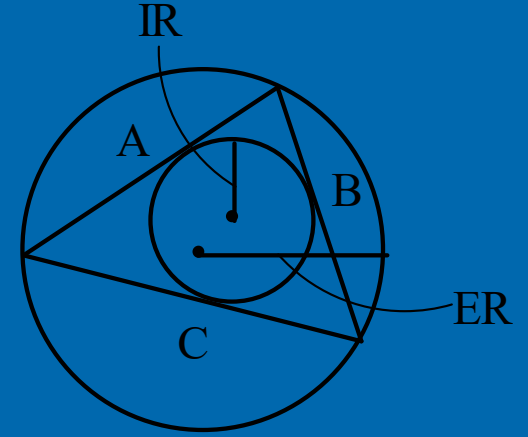
grandyendir.



# Sorular

5. Bileşik faiz P-ana para, n-yıl sayısı, i-faiz oranı (desimal), m-bir yıl içindeki tekrarlama sayısı olmak üzere  $A=P*(1+i/m)^{n*m}$  denklemi ile verilebilir. Bu denkleme göre faiz hesabı yapan algoritma ve FORTRAN programını yazınız.

# Sorular



6 Kenarlarının uzunluğu A, B, C olan bir üçgenin içine ve dışına çemberler çizilmiştir (IR içteki çemberin yarıçapı, ER dıştaki çemberin yarıçapı). Denklemdeki  $S=(A+B+C)/2$  dir. Kenar uzunlukları girilen üçgenin içindeki ve dışındaki çemberlerin yarıçaplarını A, B ve C uzunluklarına göre hesaplayan FORTRAN programını yazınız. Programı  $A=B=C=1$  ve  $A=65.23$ ,  $B=39.77$  ve  $C=58.05$  değerleri için hesap makinesi ile elde ettiğiniz sonuçlarla karşılaştırınız.

$$IR = \sqrt{\frac{(S - A) (S - B) (S - C)}{S}} \quad ER = \frac{ABC}{4\sqrt{S(S - A) (S - B) (S - C)}}$$

1. Başla

2. "Üçgenin kenar uzunluklarını giriniz :", A,B,C

3.  $IR = \sqrt{(S-A)(S-B)(S-C)/S}$

4.  $ER = ABC / (4 * \sqrt{S(S-A)(S-B)(S-C)})$

5. "İçteki çemberin yarıçapı : ", IR

6. "Dıştaki çemberin yarıçapı : ", ER

7. Dur