

## HAFTA 7

### İÇİÇE FOR-REPEAT DÖNGÜLERİ

**ÖRNEK 1:**  $f(x) = e^x$  fonksiyonunun değerini hesaplamak için aşağıdaki eşitlik verilmektedir.

$$f(x) = e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{k=1}^{\infty} \frac{x^k}{k!}$$

Kaç adet terimin toplanacağı ve  $e$  sayısının kuvveti bilgisi dış ortamdan alınmaktadır. Buna göre  $e$  sayısının tamsayı kuvvetinin değerini hesaplamak için yukarıdaki fonksiyonu kullanan algoritmayı tasarlayınız.

```
PRINT 'Kaç adet terimin toplanacağını giriniz:'
READ N
PRINT 'e sayısının kaçınıcı kuvvetini hesaplamak istiyorsunuz?'
READ X
t←1+X
FOR I←2 TO N DO
    fac←1
    FOR J←2 TO I DO
        fac←fac*J
    REPEAT
        t←t+X^I/fac
REPEAT
PRINT 'e sayısının',X, '. Kuvveti=',t
END
```

#### 2. YOL

```
PRINT 'Kaç adet terimin toplanacağını giriniz:'
READ N
PRINT 'e sayısının kaçınıcı kuvvetini hesaplamak istiyorsunuz?'
READ X
t←1+X
fac←1
FOR I←2 TO N DO
    t←t+X^I/(fac*I)
REPEAT
PRINT 'e sayısının',X, '. Kuvveti=',t
END
```

X ve N	I	t	fac
X=4; 5		1+4=5	1
	2	5+4^2/(1*2)	1*2
	3	5+4^2/(1*2)+4^3/(1*2*3)	1*2*3
	4	5+4^2/(1*2)+4^3/(1*2*3)+4^4/(1*2*3*4)	1*2*3*4
	5	5+4^2/(1*2)+4^3/(1*2*3)+4^4/(1*2*3*4)+4^5/(1*2*3*4*5)	1*2*3*4*5

**ÖRNEK 2:** Aşağıdaki toplamın değerini girilen x ve y değerleri için hesaplayan algoritmayı tasarlayınız.

$$\sum_{i=1}^x \sum_{j=1}^y \frac{j^2}{2i-1}$$

```
PRINT 'x ve y değerlerini giriniz:'
READ x,y
t←0
FOR I←1 TO x DO
    FOR J←1 TO y DO
        t←t+J^2/(2*I-1)
    REPEAT
REPEAT
PRINT 'Toplamın değeri=',t
END
```

**ÖRNEK 3:** Bir mobilya firması, 3 tür büfe üretimi yapmakta ve bunları 7 farklı şehirde bulunan mağazalarında satmaktadır. Firma en az satışı yapılan modeli üretimden kaldırmak istemektedir. Bu amaçla 30 gün süreyle, 3 modelin satış miktarını takip etme kararı almıştır. Buna göre firmanın ürettiği modellerden en az satılanı bulan algoritmayı tasarlayınız.

```
M1←0;M2←0;M3←0
FOR I←1 TO 30 DO
    FOR J←1 TO 7 DO
        FOR K←1 TO 3 DO
            CASE
                :K=1:PRINT I,'. gün',J,'. Mağazada kaç adet satış yapıldı?';
                    READ N
                    M1←M1+N
                :K=2:PRINT I,'. gün',J,'. Mağazada kaç adet satış yapıldı?';
                    READ N
                    M2←M2+N
                :ELSE:PRINT I,'. gün',J,'. Mağazada kaç adet satış yapıldı?';
                    READ N
                    M3←M3+N
            ENDCASE
        REPEAT
    REPEAT
REPEAT
IF M1<M2 & M1<M3 THEN PRINT 'En az M1 modeli satılmıştır.' ENDIF
IF M2<M1 & M2<M3 THEN PRINT 'En az M2 modeli satılmıştır.' ENDIF
IF M3<M1 & M3<M1 THEN PRINT 'En az M3 modeli satılmıştır.' ENDIF
END
```

## 2. YOL

```
M1←0;M2←0;M3←0
FOR I←1 TO 30 DO
    FOR J←1 TO 7 DO
        FOR K←1 TO 3 DO
            IF K=1 THEN
                PRINT I,'. gün',J,'. Mağazada kaç adet satış yapıldı?';
```

```

        READ N
        M1←M1+N
    ELSE IF K=2 THEN
        PRINT I,‘. gün’,J,‘. Mağazada kaç adet satış yapıldı?’;
        READ N
        M2←M2+N
    ELSE
        PRINT I,‘. gün’,J,‘. Mağazada kaç adet satış yapıldı?’;
        READ N
        M3←M3+N
    ENDIF
REPEAT
REPEAT
REPEAT
IF M1<M2 & M1<M3 THEN PRINT ‘En az M1 modeli satılmıştır.’ ENDIF
IF M2<M1 & M2<M3 THEN PRINT ‘En az M2 modeli satılmıştır.’ ENDIF
IF M3<M1 & M3<M1 THEN PRINT ‘En az M3 modeli satılmıştır.’ ENDIF
END

```

**ÖRNEK 4:**  $a < b < c$  olmak üzere  $a^2+b^2=c^2$  eşitliğini sağlayan sayılar Pisagor üçlüsü olarak adlandırılır. Dış ortamdan sadece  $a$  sayısı istenmektedir.  $a+b+c < 1000$  koşulunu sağlayan Pisagor üçlülerini görüntüleyen ve kaç adet üçlü olduğunu bulan algoritmayı tasarlayınız.

```

PRINT ‘a sayısını giriniz:’
READ a
t←0
FOR I←a TO 998 DO
    FOR b←(I+1) TO 999 DO
        FOR c←(b+1) TO 1000 DO
            IF (I^2+b^2=c^2)&(I+b+c<1000) THEN
                t←t+1
            ENDIF
        REPEAT
    REPEAT
REPEAT
PRINT ‘İstenilen koşulları sağlayan’,t,‘adet üçlü bulunmaktadır.’
END

```

## WHILE-DO Döngü Deyimi

WHILE-DO döngü deyimi, işlem ya da işlemlerin, koşul sağlandığı sürece yapılmasını sağlamaktadır. Deyimin genel yapısı aşağıdaki şekildedir:

```

WHILE <Koşul> DO
    Yapılacak İşlem/İşlemler
END

```

Algoritma ilerlerken WHILE döngüsü ile karşılaşıldığında, eğer döngüdeki koşul sağlanıyorsa, WHILE ve END arasındaki işlem(ler) yapılır. Bu süreç, koşul sağlandığı sürece tekrarlı olarak devam eder. Koşul sağlanmadığında ise, WHILE ve END arasındaki işlemler yapılmadan döngüden çıkılır ve algoritma END'den sonraki noktadan devam eder. Eğer WHILE/DO döngü deyimi içerisinde GOTO ya da EXIT deyimleri varsa, bu deyimler, WHILE'daki koşul sağlanıyor olsa bile döngüden çıkılmasını sağlayacaktır.

Önceden de bahsedildiği üzere, FOR-REPEAT döngüsü, genellikle işlem sayısı belirli olduğunda ya da işlemler bir koşula bağlı olmaksızın tekrarlı olarak yapıldığında tercih edilmektedir. Koşula bağlı tekrarlı işlem yapılması söz konusu olduğunda da WHILE/DO döngüsü kullanılmaktadır.

**ÖRNEK 1:** 100 adet girilecek sayının açıklığını bulan ve görüntüleyen algoritmayı tasarlayınız.

```
PRINT 'Bir sayı giriniz:'
READ x
Min←x
Max←x
FOR I←2 TO 100 DO
    PRINT 'Bir sayı giriniz:'
    READ x
    IF Min>x THEN Min←x ENDIF
    IF Max<x THEN Max←x ENDIF
REPEAT
PRINT 'Girilen sayıların açıklığı=',Max-Min
END
```

## 2.YOL

```
FOR I←1 TO 100 DO
    PRINT 'Bir sayı giriniz:'
    READ x
    IF I=1 THEN
        Min←x
        Max←x
    ELSE
        IF Min>x THEN Min←x ENDIF
        IF Max<x THEN Max←x ENDIF
    ENDIF
REPEAT
PRINT 'Girilen sayıların açıklığı=',Max-Min
END
```

## WHILE-DO ile

```
S←1
WHILE S<101 DO
    PRINT 'Bir sayı giriniz'
    READ x
    IF S=1 THEN
        Min←x
        Max←x
    ELSE
        IF Min>x THEN Min←x ENDIF
    ENDIF
    S←S+1
END
```

```

        IF Max<x THEN Max←x ENDIF
    ENDIF
    S←S+1
END
PRINT 'Girilen sayıların açıklığı=',Max-Min
END

```

**ÖRNEK 2:** Sıfır veya negatif sayı girilinceye kadar klavyeden veri girişi yapılmaktadır. Girilen sayılardan,

- tek olanların sayısını ve ortalamasını
- çift olanların sayısını ve harmonik ortalamasını
- girilen en büyük sayının faktöriyelini
- toplamda kaç adet pozitif sayı girildiğini

bulan ve görüntüleyen algoritmayı tasarlayınız.

```

TT←0; HT←0; TS←0; CS←0; N←0; a←1
WHILE a=1 DO
    PRINT 'Bir sayı giriniz:'
    READ x
    IF x<=0 THEN
        IF N=0 THEN
            PRINT 'İlk seferde sıfır ya da negatif sayı girilmiştir. Hesaplama
                yapılamaz.'
            a←0
        ELSE
            GOTO 1
        ENDIF
    ENDIF
    IF x>=2 THEN
        S←x
        WHILE S>=2 DO
            S←S-2
        END
        IF S=0 THEN
            CS←CS+1
            HT←HT+1/x
            N←N+1
        ELSE
            TS←TS+1
            TT←TT+x
            N←N+1
        ENDIF
    ELSE
        TS←TS+1
        TT←TT+x
        N←N+1
    ENDIF
    IF N=1 THEN
        Max←x
    ELSE
        IF Max<x THEN Max←x ENDIF
    ENDIF
END
1: PRINT 'Tek sayıların ortalaması=',TT/TS

```

```

PRINT 'Çift sayıların harmonik ortalaması=',CS/HT
F←1
FOR I←2 TO Max DO
    F←F*I
REPEAT
PRINT 'Girilen en büyük sayının faktöryeli=',F
PRINT 'Toplamda',N, 'adet pozitif sayı girilmiştir.'
END

```

**ÖRNEK 3:** Bir ilkokulda 1. sınıfın 3 şubesi bulunmaktadır. Her sınıf için öğrencilerin ortalama vücut ağırlıkları belirlenmek istenmektedir. Sınıflarda okuyan öğrenci sayısı bilinmemektedir. Bu nedenle, her sınıftaki öğrencilerin teker teker ağırlıkları girilmekte ve o sınıf için veri girişinin devam edip etmeyeceği sorusuna hayır cevabı verildiğinde diğer sınıf için veri girişi başlamaktadır. Buna göre sınıflara göre öğrencilerin ortalama vücut ağırlıklarını hesaplayan ve görüntüleyen algoritmayı tasarlayınız.

```

FOR I←1 TO 3 DO
    S←0;T←0;K←1;
    WHILE K=1 DO
        PRINT 'Öğrencinin vücut ağırlığını giriniz'
        READ kg
        S←S+1;T←T+kg
        PRINT 'Başka bilgi girmek istiyor musunuz?E/H'
        READ Cevap
        IF Cevap= 'H' THEN K←0 ENDIF
    END
    PRINT I, '. Şubenin vücut ağırlığı ortalaması=',T/S
REPEAT

```