

Problem-1.9. Verilen Bir Pozitif Tamsayının Mükemmel Sayı Olup Olmadığını Bulma

Verilen bir pozitif tamsayının mükemmel sayı olup olmadığını bulan programı yazalım. Kendisi dışındaki bütün pozitif bölenleri (çarpanları) toplamı kendisine eşit olan sayılara *mükemmel sayılar* denir. Örnek olarak, 6 sayısının kendisi hariç pozitif tamsayı bölenleri 1, 2, 3 dir ve $6=1+2+3$ olduğu için 6 bir mükemmel sayıdır.

Mükemmel sayıların başlıca özellikleri aşağıdaki gibidir.

1. Sonsuz sayıda mükemmel sayı vardır.
2. Herhangi bir $k>1$ için 2^k-1 asal ise $2^{k-1}(2^k-1)$ bir mükemmel sayıdır ve mükemmel sayıların hepsini üreten bir formüldür.
3. Bütün mükemmel sayılar 6 ve 8 ile biterler.
4. Günümüze kadar 44 adet mükemmel sayı bulunmuştur. 44. mükemmel sayının 19 milyondan fazla basamağı vardır.
5. 8. mükemmel sayıyı bulan Euler, kendinden önceki matematikçilerden farklı olarak, tek mükemmel sayıların da olabileceğini ileri sürdü. Günümüze kadar bu konuda yapılmış olan çalışmalar, ne bu iddianın doğruluğunu ne de yanlışlığını ispatlamaya yetmiştir.

Verilen bir tamsayının mükemmel sayı olup olmadığını bulan algoritma adımları ve program aşağıdaki gibidir.

A1. n tamsayısını GİR/OKU

A2. $t=0, s=1$ al

A3. $s=s+1$ al

A4. EĞER $n=INT(n/s)*s$ İSE

$a=INT(n/s)$

$t=t+a$ al

A5. A3'e GİT

A6. EĞER $t=n$ İSE

n , "mükemmel sayıdır" YAZ

DEĞİLSE

n , "mükemmel sayı değildir" YAZ

DUR

INPUT "sayı=", n

t = 0

s = 1

A3:

s = s + 1

IF n = INT(n / s) * s THEN

a = INT(n / s)

t = t + a

END IF

```
IF s<n THEN GOTO A3
IF t = n THEN
    PRINT n; "mükemmel sayıdır"
ELSE
    PRINT n; "mükemmel sayı değildir"
END IF
```

Problem-1.10. Verilen Bir Sayının Asal Olup Olmadığını Bulma

Verilen bir tamsayının asal sayı olup olmadığını bulan programı yazalım. Sadece kendisine ve 1 sayısına bölünebilen sayılara *asal sayılar* denir. Örnek olarak, 11 sayısının kendisi ve 1 sayısı hariç pozitif tamsayı böleni olmadığından bu sayı bir asal sayıdır. Verilen bir tamsayının asal sayı olup olmadığını bulan algoritma adımları ve program aşağıdaki gibidir.

```
A1. n tam sayısını GİR/OKU
A2. t=0, i=1 al
A3. EĞER i > n İSE A6'ya GİT
A4. EĞER INT(n/i)= n/i İSE t=t+1 al
A5. i=i+1 al. A3'e GİT
A6. EĞER t>2 İSE
    n, "asal sayı değildir" YAZ
    DEĞİLSE
    n, "asal sayıdır" YAZ
    DUR
```

```
INPUT "sayı=", n
t = 0
i=1
A3:
IF i>n THEN GOTO A6
IF INT(n / i)=n / i THEN t = t + 1
i=i+1
GOTO A3
A6:
IF t >2 THEN
    PRINT n; "asal sayı değildir."
ELSE
    PRINT n; "asal sayıdır."
END IF
```

Problem-1.11. Verilen Bir Sayının Faktöriyelini Hesaplama

Verilen bir sayının faktöriyelini hesaplayacak algoritmayı geliştirelim ve programı yazalım. Verilen n – tamsayısı için

$n! = n \times (n-1) \times (n-2) \times \dots \times 3 \times 2 \times 1$ biçiminde hesaplanır.

- A1.** Faktöriyeli alınacak sayıyı (n 'i)OKU
A2. Eğer $sayı=0$ ise $sonuc=1$ YAZ ve DUR
A3. $faktoriyel=1$
A4. $say=1$
A5. $faktoriyel=faktoriyel*say$
A6. $say=say+1$
A7. EGER $say<n$ İSE A5' E GİT
A8. $faktoriyel$ YAZ ve DUR

```
INPUT "n=", n
IF n = 0 THEN
    PRINT "sonuç=", 1
    END
END IF
F = 1
say = 1
A5:
F = F * say
say = say + 1
IF say < n THEN GOTO A5
PRINT "sonuç=", F
```

A4'deki gibi $say=1$ 'den n 'e kadar Algoritma adımlarının tekrar etmesi halinde en çok kullanılan döngü FOR-NEXT döngüsüdür. FOR-NEXT döngüsünün en genel kullanım şekli aşağıdaki gibidir.