

WHILE-WEND döngüsü kullanımı ile program aşağıdaki şekilde yazılır.

```
INPUT "N=", N
T = 1
F = 1
i=0
WHILE i<N+1
  i=i+1
  F = F * i
  T = T + 1 / F
  PRINT USING "#.#####"; T
WEND
```

Öte yandan, FOR-NEXT döngüsü yerine bir koşul doğru iken veya doğru olana kadar tekrarlanan DO-LOOP döngüleri de kullanılabilir. DO-LOOP genel kullanımı,

DO		DO WHILE/UNTIL <i>koşul</i>
<i>deyim(ler)</i>	veya	<i>deyim(ler)</i>
LOOP WHILE/UNTIL <i>koşul</i>		LOOP

biçimindedir. “*koşul*” doğru olana kadar veya “*koşul*” doğru olduğu sürece DO-LOOP döngüsü içerisinde kalan “*deyim(ler)*” uygulanır.

```
INPUT "N=", N
T = 1
F = 1
i=0
DO
  i=i+1
  F = F * i
  T = T + 1 / F
  LOCATE 10 + i, 20
  PRINT USING "#.#####"; T
LOOP WHILE i<N+1
```

Problem-1.22. Verilen Bir x Değeri İçin Parçalı Fonksiyonun Değerini Ekrana Yazdırma

Verilen bir x değeri için,

$$f(x) = \begin{cases} 1 & , & x < 6 \\ 2 & , & 6 \leq x < 8 \\ 3 & , & 8 \leq x < 9 \\ 4 & , & x \geq 9 \end{cases}$$

biçiminde tanımlanan f parçalı fonksiyonunun değerini ekrana yazdıracak programı yazalım. Her bir aralıkta $f(x)$ fonksiyonu farklı değerler almaktadır. Bu nedenle, bu fonksiyonun değerini bulacak program için SELECT-CASE deyiminden yararlanılabilir. SELECT-CASE deyiminin kullanımı,

SELECT CASE *durum*

CASE *durumun 1. Değeri*

Deyimler

CASE *durumun 2. Değeri*

Deyimler

•
•
•

CASE ELSE

ELSE deyimleri

END SELECT

biçimindedir. Programda SELECT-CASE deyimi ile karşılaşıldığında “*durum*” olarak belirtilen test ifadesi için her bir CASE ile tanımlanan *durum değerlerine* göre ilgili deyimler işletilir. CASE ELSE deyim satırına CASE dallanmaları dışında bir değer olduğunda geçilir. Programda, <, >, <=, >=, <>, = operatörlerinin kullanımı CASE IS deyimi ile yapılır.

INPUT “x=”,x

SELECT CASE x

CASE IS<6

fonk=1

CASE IS>=6 AND x<8

fonk=2

CASE IS>=8 AND x<9

fonk=3

CASE IS>=9

fonk=4

END SELECT

PRINT “fonksiyonun değeri=”, fonk

