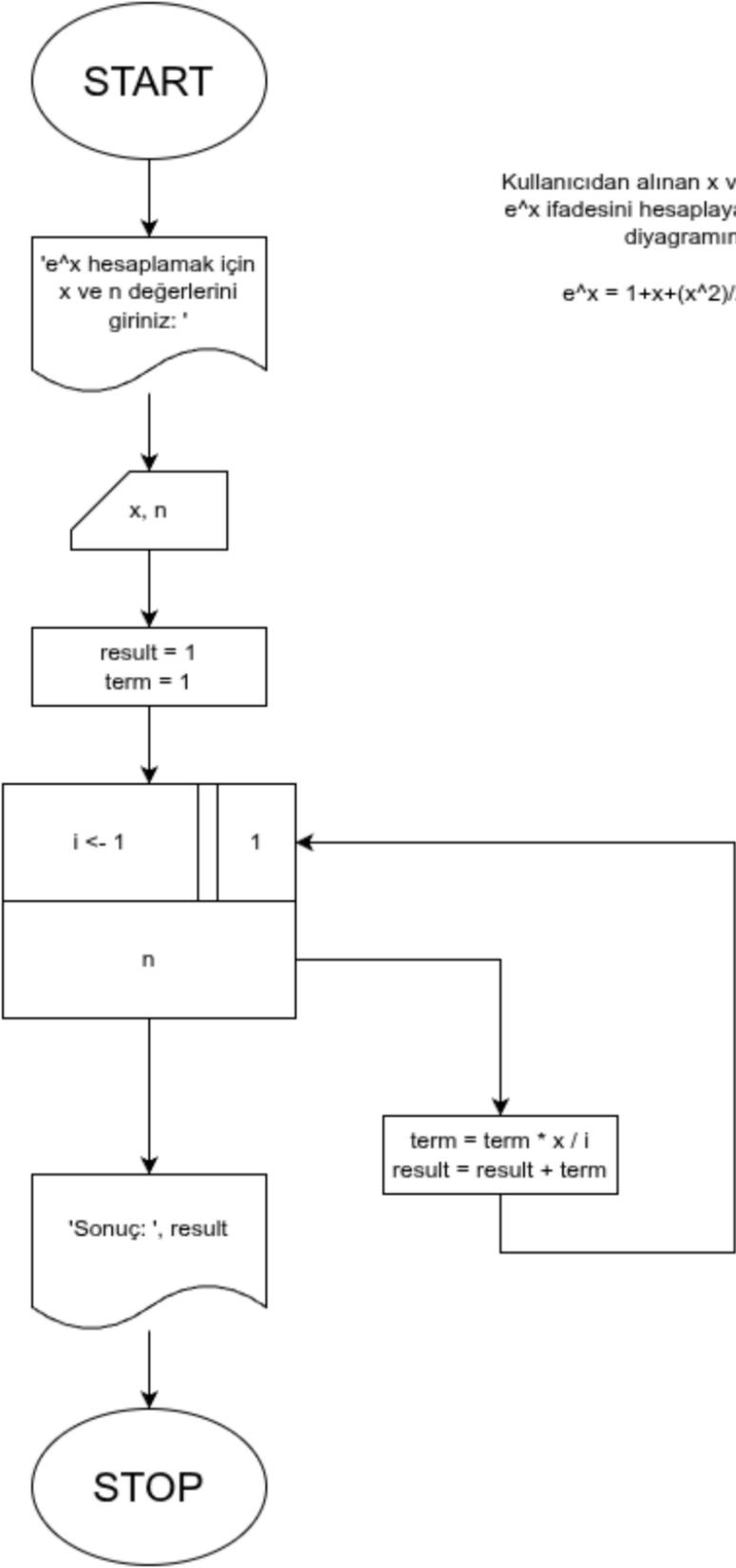


DİKKAT: Çalışma Notu Hakkında

Aşağıda yer alan sorular ve çözümleri tamamen bireysel çalışma ve pratik yapma amacıyla oluşturulmuştur. Bu doküman;

- Yazım hataları, mantıksal yanlışlıklar içerebilir.
- Resmi sınav müfredatıyla %100 uyum garantisini vermez.
- Cevap anahtarı "tek doğru" yolu temsil etmiyor olabilir.

Bu dokümanı kullanırken ana ders kitaplarınızı ve ders notlarınızı teyit mekanizması olarak kullanmanız önemle rica olunur. Oluşabilecek hatalardan hazırlayan kişi sorumlu tutulamaz.



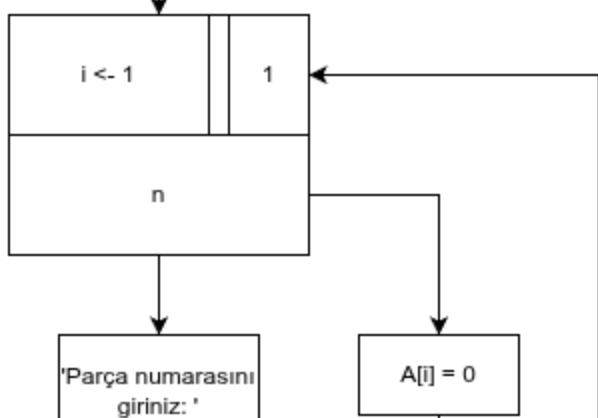
Kullanıcıdan alınan x ve n değerlerine göre e^x ifadesini hesaplayan algoritmanın akış diyagramını çiziniz.

$$e^x = 1+x+(x^2)/2!+\dots+(x^n)/n!$$



'Puzzle parça sayısını giriniz: '

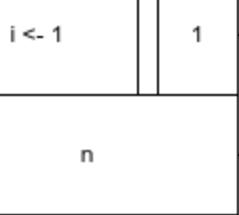
n



x

x != 0

count = 0



'Toplam ', count, ' adet parça kaybolmuştur.'

A[x] = 1

'Parça numarasını giriniz: '

x

F A[i] == 0 T

count = count + 1

i, '. parça kaybolmuştur.\n'

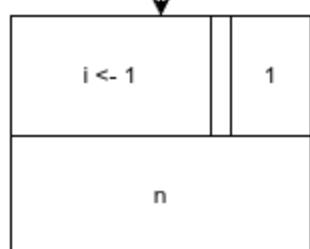
STOP

n parçadan oluşan bir yapbozun parçaları 1'den n'e kadar numaralandırılmıştır. Yapbozun bazı parçaları kaybolmuştur. Kullanıcı, öncelikle yapbozun toplam parça sayısını (n), ardından elinde bulunan parçaların numaralarını girecektir. Kullanıcı, elindeki tüm parçaları girdikten sonra 0 değerini girerek işlemi sonlandırır. Kaybolan parçaların numaralarını ve toplam kayıp parça sayısını bulan algoritmanın akış diyagramını çiziniz. (Kullanıcının tüm verileri doğru girdiği varsayılmaktır.)

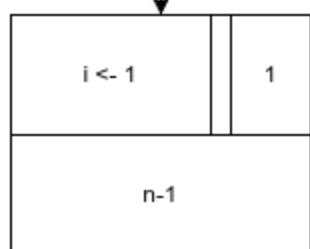


'Çokgenin kenar sayısını giriniz: '

n



total = 0



Soru: n kenarlı kapalı bir çokgenin geometrik hesaplamalarını yapan bir algoritma tasarılanacaktır. Kullanıcıdan öncelikle köşe sayısını (n) istenecek, ardından çokgenin köşelerine ait (x, y) koordinatları **ardışık sırayla (saat yönünde veya tersinde)** girilecektir.

Bu verileri kullanarak:

1. Her bir kenarın uzunluğunu ayrı ayrı hesaplayan,
2. Çokgenin toplam çevre uzunluğunu bulan,
3. Sonuçları ekrana yazdırın,

algoritmanın akış diyagramını çiziniz.

i, '. koordinatı giriniz (x, y): '

X[i], Y[i]

```
length = SQRT( (X[n]-X[1]) * (X[n]-X[1]) + (Y[n]-Y[1]) * (Y[n]-Y[1]) )
total = total + length
```

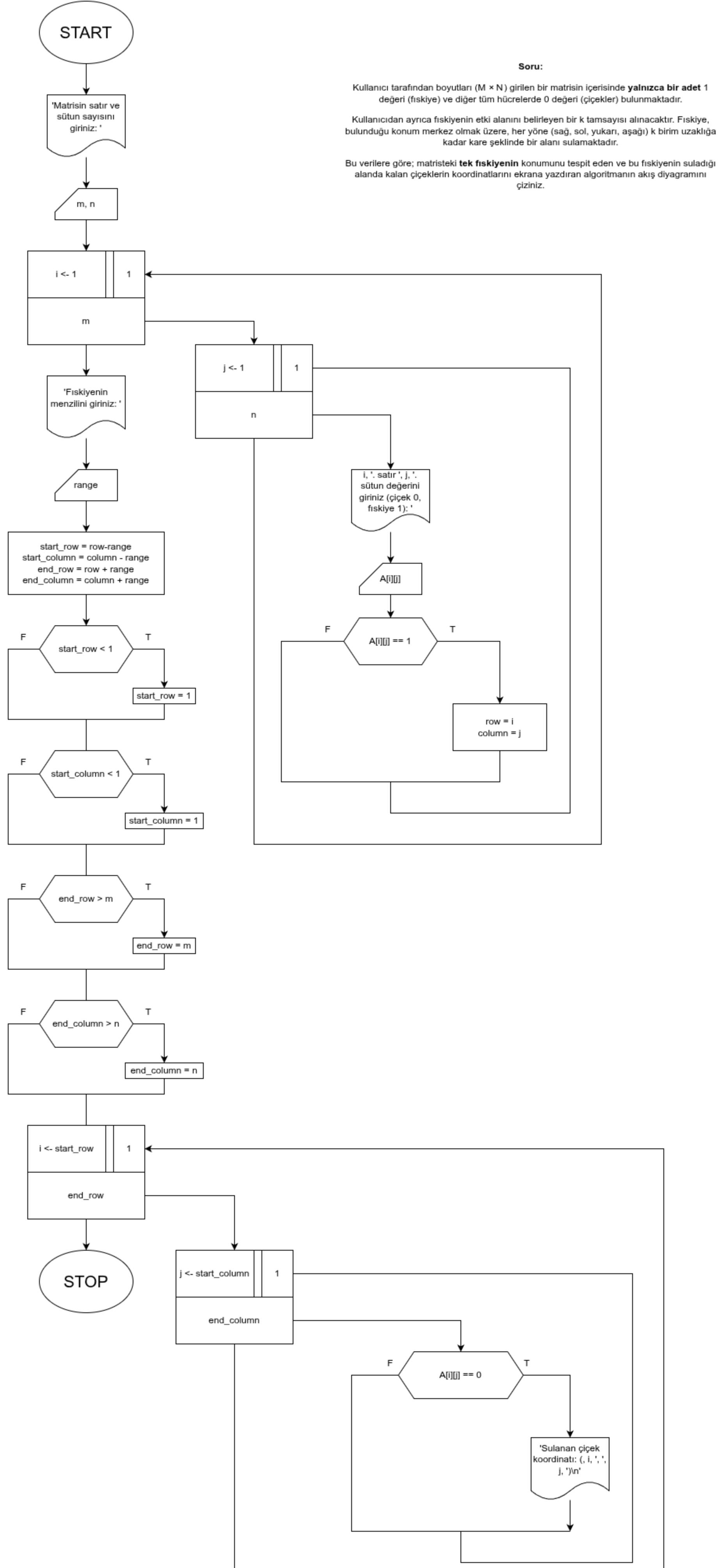
n, '. kenar uzunluğu: ', length

'Çevre uzunluğu: ', total

```
length = SQRT( (X[i+1]-X[i]) * (X[i+1]-X[i]) + (Y[i+1]-Y[i]) * (Y[i+1]-Y[i]) )
total = total + length
```

i, '. kenar uzunluğu: ', length





Soru:

Kullanıcıdan şehir sayısı (N) ve bu şehirler arasındaki uçuş bağlantılarını temsil eden bir matris (komşuluk matrisi) alınacaktır.

- Matrisin (i, j) hücreindeki **1** değeri, i . ve j . şehirler arasında uçuş olduğunu; **0** değeri ise uçuş olmadığını gösterir.
- Uçuşlar çift yönlüdür.

Bu bilgilere göre:

Kullanıcıdan alınan iki farklı şehrin birbirine tek aktarma ile ulaşıp ulaşamayacağını bildiren algoritmanın akış diyagramını (flowchart) çiziniz.

Örnek Senaryo 1:

Girdi:

Şehir Sayısı (N): 5
Uçuş Matrisi:

```
0 1 1 1 0
1 0 1 0 0
1 1 0 0 0
1 0 0 0 0
0 0 0 0 0
```

Kaynak: 2
Hedef: 3

Çıktı:

"2. şehirden 3. şehrre tek aktarma ile varılabilir."

Örnek Senaryo 2:

Girdi:

Şehir Sayısı (N): 5
Uçuş Matrisi:

```
0 1 1 1 0
1 0 1 0 0
1 1 0 0 0
1 0 0 0 0
0 0 0 0 0
```

Kaynak: 1
Hedef: 4

Çıktı:

"1. şehirden 4. şehrre tek aktarma ile varılamaz."

