

---> Algoritma nedir? Algoritmanın gerekliliğini ve avantajlarını açıklayınız.

> Algoritma, belirli bir problemi çözmek veya belirli bir amaca ulaşmak için çözüm yolunun adım adım tasarlanmasıdır.

Programın işlem akışı görülür, işlemler daha hızlı çalışır bu sayede program yazmak kolaylaşır.

---> Program yazılmadan önce algoritması mı hazırlanmalı yoksa akış diyagramı mı çizilmelidir?

> Her ikisi de yapılabilir, programı yazan kişiye bağlıdır.

---> Algoritma hazırlanırken dikkat edilmesi gereken hususları açıklayınız.

> - Algoritma adımları sıralı açık ve net olmalıdır kesin ifadelerden oluşmalıdır.

- Farklı programlama dillerine kolayca uyarlanabilmelidir.

- Algoritmanın bir çözümü olmalıdır.

---> Değişken nedir? Programlarda neden değişkene ihtiyaç duyulmaktadır?

> Eğer değişkenler olmasaydı programlama yaparken tüm olasılıkları tek tek yazmak zorunda kalırdık.

Değişkenler sayesinde hem zaman kazanabiliriz hem de hata olasılığı daha düşük seviyelerde olur.

---> Sayaçlar nerelerde ve niçin kullanılmaktadır?

> Sayaçları işlemlerin kaç defa yapıldığını veya bir veri kümesinde kaç tane eleman olduğunu bulmak için kullanırız.

---> Algoritmanın Sonucunu Hesaplayınız.

> T=60

---> Algoritmanın Sonucunu Hesaplayınız.

> F=70

---> Girilen üç sayıdan en büyüğünü bulan programın algoritmasını hazırlayınız.

- > 1. Başla
- 2. A,B,C değerlerini sayı olarak tanımla
- 3. A,B,C değerlerini gir
- 4. Eğer $A > B, C$ ise Git 7
- 5. Eğer $B > A, C$ ise Git 8
- 6. Eğer $C > A, B$ ise Git 9
- 7. Yaz A Git 10
- 8. Yaz B Git 10
- 9. Yaz C Git 10
- 10. Dur

---> Girilen üç sayıyı küçükten büyüğe doğru sıralayan programın algoritmasını hazırlayınız.

- > 1. Başla
- 2. A,B,C değerlerini sayı olarak tanımla
- 3. A,B,C değerlerini gir
- 4. Eğer $A < B < C$ ise Git 10
- 5. Eğer $B < A < C$ ise Git 11
- 6. Eğer $B < C < A$ ise Git 12
- 7. Eğer $C < B < A$ ise Git 13
- 8. Eğer $C < A < B$ ise Git 14
- 9. Eğer $A < C < B$ ise Git 15
- 10. Yaz $A < B < C$ Git 16
- 11. Yaz $B < A < C$ Git 16
- 12. Yaz $B < C < A$ Git 16
- 13. Yaz $C < B < A$ Git 16

14. Yaz $C < A < B$ Git 16

15. Yaz $A < C < B$ Git 16

16. Dur

---> 1-99 arasındaki tek ve çift sayıların toplamaları ile çarpımlarını ayrı ayrı hesaplayan programın algoritmasını hesaplayınız.

> 1. Başla

2. Tek sayıların toplamını tutan bir değişken tanımla ($tektop = 0$)

3. Tek sayıların çarpımını tutan bir değişken tanımla ($tekcarp = 1$)

4. Çift sayıların toplamını tutan bir değişken tanımla ($cifttop = 0$)

5. Çift sayıların çarpımını tutan bir değişken tanımla ($ciftcarp = 1$)

6. Sayacı 1 olarak ayarla ($sayac = 1$)

7. Sayacın 99'dan küçük veya eşit olduğu sürece döngü yap

8. Eğer sayacın 2'ye bölümünden kalan 0 ise ($sayac \% 2 == 0$)

9. Sayacı çift sayıların toplamına ekle ($cifttop = cifttop + sayac$)

10. Sayacı çift sayıların çarpımına çarp ($ciftcarp = ciftcarp * sayac$)

11. Eğer sayacın 2'ye bölümünden kalan 1 ise ($sayac \% 2 == 1$)

12. Sayacı tek sayıların toplamına ekle ($tektop = tektop + sayac$)

13. Sayacı tek sayıların çarpımına çarp ($tekcarp = tekcarp * sayac$)

14. Sayacı bir arttır ($sayac = sayac + 1$)

15. Döngü bittiğinde, ekrana tek ve çift sayıların toplamaları ve çarpımları yaz

16. Dur