

Ödev 2

1. Aşağıdaki problemi çözebilecek algoritmanın sözde kodunu (pseudocode) ve C/C++ kodunu yazınız. Sizden ikisi de aynı zamanda sıfıra eşit olmayan 2 negatif olmayan tam sayının ortak bölenlerinin en büyüğü (OBEB) Eucild algoritmasını modifiye ederek bulmanız istenmektedir. Bunun için aşağıdaki özellikleri kullanmanız gerekmektedir.

- a) $OBEB(2a, 2b) = 2 \cdot OBEB(a, b)$
- b) b tek ise $OBEB(2a, b) = OBEB(a, b)$
- c) $OBEB(a, 0) = a$
- d) $OBEB(a, b) = OBEB(b, a - b)$, eğer $a \geq b$ ise
- e) $OBEB(a, b) = OBEB(b, a)$

Kodunuzda bölme işlemi olarak sadece 2 ile bölmeye izin verilmektedir. 2 dışında başka bir sayıyla bölme işlemi yapılarak oluşturulan kodlar değerlendirme dışı tutulacaktır. Algoritmanız aşağıdaki örnekte olduğu gibi çalışmalıdır.

Örnek. $OBEB(60, 50) = 2$. $OBEB(30, 25) = 2$. $OBEB(15, 25) = 2$. $OBEB(15, 10) = 2$. $OBEB(15, 5) = 2$. $OBEB(5, 10) = 2$. $OBEB(5, 5) = 2$. $OBEB(5, 0) = 10$

Kodunuzun giriş değişkeni kullanıcı tarafından girilecek olan ikisi de aynı zamanda sıfıra eşit olmayan iki negatif olmayan tam sayılardır. Kullanıcı koşula uymayan sayılar girdiğinde uygun sayıları girene kadar kodunuz tarafından uyarı almalıdır. Çıkış değişkeniniz girilen sayıların OBEB i olmalıdır.

2. Aşağıdaki problemi çözebilen algoritmanın C/C++ kodunu yazınız.

Problem: Verilen n elemanlı tam sayılar dizisinin ardışık terimlerinden oluşan ve azalan altdizilerini alt alta yazdırınız. Örneğin, A dizisi 9,7,5,17,13, 12, 19, 22, 20, 18, 17, 21, 24 ise sonuç

9,7,5
17, 13,12
19
20, 18, 17
21
24
Olmalıdır.

3. Aşağıdaki problemi çözebilen algoritmanın sözde kodunu(pseudocode) ve C/C++ kodunu yazınız.

Kullanıcıdan pozitif tam n sayısı girmesini isteyiniz. Pozitif tam olmayan bir giriş yapmak isterse uyarınız ve tekrar giriş yapmasını isteyiniz. Kullanıcı n sayısını girdikten sonra tekrardan n tane tam sayı girmesini isteyiniz ve daha az veya daha çok sayı girmesine izin vermeyiniz. Kullanıcının girdiği sayılar $A[1], \dots, A[n]$ ise ilk satıra $A[2] \dots A[n]$ çarpımını, 2. satıra $A[1].A[3] \dots A[n]$, 3. satıra $A[1].A[2].A[4] \dots A[n]$, ... n . satıra $A[1].A[2] \dots A[n-1]$ çarpımını yazınız.

Problemin çözümünde bölme işlemi kullanılmayacaktır ve bölme işlemi yapılan kodlar değerlendirilmeyecektir.

- a) Problemi $\Theta(n^2)$ işlem zamanında yapabilen kod yazınız
b) Problemi $\Theta(n)$ işlem zamanında yapabilen kod yazınız

.