

Derleme komutları

1. gcc Q1.c
2. ./a.out<input1.txt>myoutput1.txt
3. diff output1.txt myoutput1.txt ""output1.txt dokümanı ile myoutput1.txt dokümanını karşılaştırır. ""
4. diff --ignore-all-space output1.txt myoutput1.txt ""output1.txt dokümanı ile myoutput1.txt dokümanını boşlukları dikkate almadan karşılaştırır. ""

Lütfen yukarıdaki komutlar ile size verilen output dosyalarını kendi ürettiğiniz output dosyalarınız ile çalıştırınız!

SORU

Kullanıcıdan veri girişi sonlandırılıncaya kadar (enter tuşuna basılıncaya kadar) okuma yapan ve aşağıdaki kuralları sağlayacak şekilde alınan veriyi kontrol eden bir program yazmanız beklenmektedir. Kullanıcı veri girişi yaptığında programınız bu veriyi okuyacak ve aşağıda tanımlanan fonksiyonlar ile elde edilen sonucu üretecektir. Kullanıcı istediği uzunlukta veri girişi yapabilir. Bunu sınırlandırmak ve programın güvenliğini sağlamak için en fazla 20 elemanlık bir dizin oluşturacak şekilde programınızı tasarlayınız. 20’den fazla boyuta sahip olan bir input girilmeyeceğini varsayabilirsiniz.

void get_input (int size, char my_array[]): Fonksiyonu size uzunlukta elemanı my_array dizinine alacaktır.

void erase_non_digit (int size, int my_new_size, char my_array[], int my_new_array[]): Fonksiyonu kullanıcıdan alınan my_array dizini içerisinde sadece sayı karakterlerin (rakamların) oluşturduğu my_new_array dizinini oluşturacaktır. Örneğin kullanıcı “12 a. e 4]b” şeklinde bir input girmişse bu fonksiyon my_array dizinini 3 elemanlı 124 durumuna indirgeyecektir. Bu yapı my_new_array üzerinde saklanacak ve uzunluğu my_new_size ile tanımlanacaktır. (yeni dizinin tipine dikkat!) Programınızın oluşturacağı beklenen output için aşağıda bazı kurallar tanımlanmıştır. Her bir kuralı gerçekleştirecek 3 ayrı fonksiyon yazmanız gerekmektedir.

Kural 1: Beklenen yapıda **tam 11 rakam** olmalıdır.

int rule_1 (int size, int A[]): Bu fonksiyon argüman olarak alınan A dizinin uzunluğunun 11 olup olmadığını kontrol eder. Kuraldan geçerse 1, geçemezse 0 şeklinde sonuç döndürür.

Kural 2: 1,3,5,7 ve 9. basamaklarının (array indeksi değil birden başlayan basamak) toplamının 7 katından 2,4,6 ve 8. basamakların toplamı çıkarıldığında elde edilen sonucun 10’a bölümünden kalan sayı 10. basamaktaki sayıya eşit olmalıdır.

int rule_2(int size, int A[]): Bu fonksiyon size elemanlı A dizinin yukarıdaki kuralı sağlayıp sağlamadığını kontrol eder. Kuraldan geçerse 1, geçemezse 0 şeklinde sonuç döndürür.

Kural 3: İlk 10 hanenin toplamından elde edilen sonucun 10’a bölümünden kalan sayı 11. basamaktaki sayıya eşit olmalıdır.

int rule_3(int size, int A[]): Bu fonksiyon size elemanlı A dizinin yukarıdaki kuralı sağlayıp sağlamadığını kontrol eder. Kuraldan geçerse 1, geçemezse 0 şeklinde sonuç döndürür.

Programınız çalıştığında ilk olarak;

```
void get_input ( int size, char my_array[ ] )
```

```
void erase_non_digit ( int size, int my_new_size, int, my_array[ ], int my_new_array[ ] )
```

fonksiyonlarını çağrılacaktır. Ardından 3 kuralın sağlanıp sağlanmadığı test edilecektir. rule_1 fonksiyonu olumsuz sonuç döndürdüğünde kullanıcıdan yeni bir veri girişi yapması istenecek ve yukarıdaki ilk iki fonksiyon(**get_input** ve **erase_non_digit**) yeniden çağrılacaktır. Kullanıcının kural 1'i geçmek için **toplamda 3 veri girişi** yapma hakkı olacaktır. Bu aşamadan geçilirse diğer rule_2 ve rule_3 test edilecektir. Eğer rule_2 aşamasından geçemez ise rule_3 kontrol edilmeyecektir. Kural 2 veya 3'den geçilemez ise, **kullanıcıdan tekrar veri girdisi alınmayacaktır** ve program sonlanacaktır. Alınan veri tüm testleri geçmiş ise ekrana elde edilen input sonucu bastırılacaktır. Eğer kurallardan birini geçemezse hangi kural ihlalinin dolayısı testin geçilemediği ekrana bastırılarak program sonlanacaktır.

Sizlere verilen input-output dosyalarını inceleyiniz. Çıktılarınız output dosyalarındaki ile **birebir** aynı olmak zorundadır.

Örnek

input1:

```
(67a?14!8[2.9-34245  
(67.?14!8e2.9-34246  
(67?14!8 2.9-34.04
```

output1:

```
Invalid input: must contain exactly 11 digits  
Invalid input: must contain exactly 11 digits  
Valid: 67148293404
```

Kural 1

Aşağıdaki input1 örneğindeki 1. satırdan rakam olmayan tüm karakterler silindikten sonra:

```
(67a?14!8[2.9-34245 => 671482934245 => 12 basamaklı => Kural 1'den geçemez
```

Bu sayı 11 basamaklı olmadığından Kural 1'den geçemeyecek. Kural 1 sağlanmadığından, yeniden input istenecektir.

2. satırdaki dizide:

```
(67.?14!8e2.9-34246 => 671482934246 => 12 basamaklı => Kural 1'den geçemez
```

Yine bu sayı da 11 basamaklı olmadığından Kural 1'den geçemeyecek. Bu sefer 3. Kez input istenecektir.

3. satırda ise:

```
(67?14!8 2.9-34.04 => 67148293404 => 11 basamaklı => Kural 1'den geçer
```

Kural 2

67148293404 =>

Basamak	1	2	3	4	5	6	7	8	9	10	11
Sayı	6	7	1	4	8	2	9	3	4	0	4

$$6 + 1 + 8 + 9 + 4 = 28$$

$$7 + 4 + 2 + 3 = 16$$

$$(28 * 7) - 16 = 180$$

$$180 \% 10 = 0$$

10. hanedeki sayı == 0 => Kural 2'den geçer

Kural 3

67148293404 =>

Basamak	1	2	3	4	5	6	7	8	9	10	11
Sayı	6	7	1	4	8	2	9	3	4	0	4

67148293404 =>

$$6 + 7 + 1 + 4 + 8 + 2 + 9 + 3 + 4 + 0 = 44$$

$$44 \% 10 = 4$$

11. hanedeki sayı == 4 => Kural 3'den geçer.