

CSE 102 Programlama Ödevi 2

VADE

27 Ekim 2024, 23:55

Açıklama

- Bu bireysel bir ödevdir. Lütfen işbirliği yapmayın.
- Bu belgenin ödevi açıkça tanımlamadığını düşünüyorsanız, çok geç olmadan sorular sorun.

Herhangi bir hatayı düzeltmek için size bir şans verilmeyecektir.

- Aşağıdaki maddelere göre bir C kodu yazınız:
 - `input.txt` dosyasını okuyun
 - `input.txt`, boşluk (space, tab, newline vs...) ile ayrılmış tamsayılar içerir. Tamsayıların sayısı bilinmemektedir. 3 veya 300000 veya daha fazla olabilir. Dosyadaki tamsayı sayısı hakkında herhangi bir varsayımda bulunmayın. (Dosya sonuna kadar okuyun. Tüm sayıları bellekte saklamaya çalışmayın. Tüm sayıları saklamanıza gerek yok).
 - Bir segmentin boyutu için bir makro tanımlayın. Bu makro `X` olsun. `X`, 1 ile 10 arasında bir tamsayı değeridir.
 - Programınız girdi dosyasındaki **farklı** segmentlerin tekrarlarını sayacaktır. En son görülen `Y` farklı diziye sayın. `Y`, programın başında tanımlanan bir makrodur ve 1 ile 20 arasında değerler alabilir.
 - En son görülen `Y` farklı dizinin tekrarlarını `output.txt` dosyasına yazın.

Örnek

- makrolar:

```
#define X 2 /* segment boyutu 2'dir. Bir segment 2 ardışık tamsayı ile tanımlanır. */
#define Y 3 /* Bu, 3 farklı segment sayacağınız anlamına gelir */
```

- `input.txt`:
- (`int1`, `int2` vb. etiketler yerine gerçek sayılar olacaktır)

```
int1 int2 int3 int2
int2 int1 int2
```

- Bu dizide 2 boyutlu segmentleri sayacağız. Çünkü `X 2` olarak tanımlanmıştır
- 2 sayı okuyun: `int1 int2`. Bu segmentin daha önce görülüp görülmediğini kontrol edin. Daha önce görülmüşse, sayısını artırın. Eğer daha önce hiç görülmemiş bir segment ise, saymaya başlayın.
- diziden başka bir sayı okuyun. Segment şimdi `int2 int3`. (temelde `X` boyutunda bir pencereyi her seferinde bir adım kaydırıyoruz). Bu segmenti sayın ve pencereyi tekrar kaydırın. Dosya sonuna ulaşana kadar bu işlemi tekrarlayın.
- En fazla `Y` sayıda farklı segmenti ve bunların sayılarını hatırlayabilirsiniz. Yeni bir segmentle karşılaşırsanız, saymakta olduğunuz en eski segmenti bırakmanız ve onun yerine yeni gördüğünüz segmenti saymaya başlamanız gerekir. (Hatırlanan segmentleri ve sayılarını ilk gördükleri sırada tutmanız gerekir).
- Segment1 segment2 ve segment3'ü saydığınızı varsayalım. Yeni bir segment (segment4) görürseniz, segment1'i saymayı bırakırsınız. Programınız segment2 segment3 ve segment4'ü hatırlayabilir. Eğer segment1 yeniden görünürse, bu yeni segmenttir. Bu durumda, segment2'yi bırakmalı ve segment3, segment4 ve segment1'i saymaya başlamalısınız.
- Dosyanın sonuna ulaştığınızda, sayılan tüm segmentleri (hatırladığınız son segmentler) ve sayılarını bir çıktı dosyasına kaydedersiniz. Dosyanın adı `output.txt`'dir
- Verilerin biçimine ve sırasına dikkat edin.

- segment: <segment> := belirli bir segmentte boşluk bırakılarak ayrılmış tam sayılar

- çıktı dosyasının biçimi. (her segment yeni bir satıra yazdırılır.)

<segment> : <sayı>

<segment> : <sayı>

...

- yürütmeden sonra `output.txt` (`int2` veya `int1` vb. elbette gerçek tam sayılarla değiştirilecektir):

```
int2 int2 :
1 int2 int1
: 1 int1
int2 : 1
```

Açıklamalar

- Yalnızca `C*Y*X` veya daha küçük boyutlarda diziler oluşturabilirsiniz. (C burada küçük bir sabittir.) Daha büyük diziler oluşturursanız `0.0` elde edersiniz. Verimli bir şekilde kodlayın. Gereksiz diziler oluşturursanız puan kaybedebilirsiniz.
- Dosyaların adlarına dikkat edin (`input.txt` ve `output.txt`). Bu konuda bir hata yaparsanız size ikinci bir şans verilmeyecektir. Programınız başarısız olacak ve notunuz `0.0` olacaktır.
- Derste ele alınmayan herhangi bir eleman kullanmayın. (Dizileri kullanabilirsiniz)
- Kodunuzu birkaç farklı senaryo ile test etmeden göndermeyin. Çok büyük girdi dosyaları ile test edin.
- Kodunuza yorumlar yazın.
- Dosya okuma/yazma işlemleri için `ftell()`, `fseek()` ve diğer faydalı fonksiyonları kullanabilirsiniz.
- Negatif ve pozitif tamsayılar olabilir. Dosyada 0'lar olabilir.
- Dosyanın hatasız olduğunu varsayabilirsiniz. (yani dosyada sadece tam sayılar vardır.)
- Dosya sonu ve başarılı okuma/yazma işlemlerini uygun şekilde kontrol edin.

Teslim ol:

- Tam bir C programının kaynak kodu. Dosyanın adı şu formatta olmalıdır: `<full_name>_PA2.c`.
- Örnek: `david_hilbert_PA2.c`. Lütfen herhangi bir Türkçe özel karakter kullanmayın.
- Bu ödev için bir IDE kullanmanıza gerek yoktur. Kodunuz bir komut penceresinde derlenecek ve çalıştırılacaktır.
- Kodunuz bir Linux makinesinde (Ubuntu) derlenecek ve test edilecektir. GCC kullanılacaktır.
- Programınızın belirli kodlamalar/işaretler/satır sonu karakterleri gerektirmediğinden emin olun. Linux ortamında oluşturulmuş bir dosya ile çalıştığından emin olun.
- Şu komutu verdiğinizde derleme hatası almadığınızdan emin olun: `gcc <full_name>_PA2.c`.
- Sonuçlarınızın doğruluğunu kontrol etmek için bir komut dosyası kullanılacaktır. Bu nedenle, beklenen çıktı biçimini ihlal etmemeye dikkat edin.
- Kısmi kredi ile ilgilenmediğiniz sürece yorumlarınızı belirtin. (Eğer tasarımı kolayca anlayamazsam, puan kaybedebilirsiniz).
- Uygulamanız bu belgedeki ifadelerle çelişiyorsa tam kredi alamayabilirsiniz.

Geç Teslim

- Kabul edilmedi.

Notlandırma (Geçici)

- Maksimum Derece: **100**.
- Çoklu testler (en az 5) yapılacaktır.

Aşağıdakilerin tümü Maksimum Nottan olası kesintilerdir.

- sabit kodlanmış değerler yerine makroları kullanın, aksi takdirde kaybedebilirsiniz: **-10**.
- Gönderim yok: **-100**.
- Derleme hataları: **-100**.
- Alakasız kod: **-100**.
- Büyük parçalar eksik: **-100**.
- Gereksiz uzun kod: **-30**.

- verimsiz uygulama: -20.

- İzin verilmeyen dil öğelerinin ve kütüphanelerin kullanılması: -100.
- Yapıyı ve verimliliği önemsememek: -30. (sabit kodlanmış değerler kullanmaktan kaçının, takip edilmesi zor ifadelerden kaçının, kod tekrarından kaçının, gereksiz döngülerden kaçının).
- Önemli sayıda derleyici uyarısı: -10.
- Yeterince yorum yapılmamış: -5. (Yorumlar İngilizcedir).
- Kaynak kod kodlaması UTF-8 değil ve karakterler düzgün görüntülenmiyor: -5. ('Visual Studio Code', 'Sublime Text', 'Atom' vb. kullanabilirsiniz. Metin düzenleyicinizin karakter kodlamasını kontrol edin ve UTF-8 olarak ayarlayın).
- Eksik veya yanlış çıktı değerleri: Testten başarısız olur.
- Çıktı biçimi yanlış: -30.
- Sonsuz döngü: Testte başarısız olur.
- Segmentasyon hatası: Testte başarısız olur.
- 5 veya daha fazla rastgele testi geçemezse: -100.
- Testi geçemezse: 20'ye kadar kesinti.
- Fazladan bir şey yazdırır: -30.
- Dosyanın sonunda boşluk/yeni satır gerektirir: -20.
- Özel satırsonu işareti (CR/LF) gerektirir: -20.
- Çıktıda istenmeyen karakterler ve boşluklar: -30.
- Gönderim, beklenenin dışında dosyalar içerir: -10.
- Gönderim dosya adlandırma kuralına uymuyor: -10.
- Kodun paylaşılması veya miras alınması: -200.