

ÖDEV-2

Dersin Adı: Veri Yapıları ve Algoritmalar

Veriliş tarihi: 17.11.2016

Teslim Tarihi: 30.11.2016 Çarşamba, Saat 17:00'e kadar (Sadece kaynak kod)

Kontrol Tarihi: 02.12.2016 Cuma, Saat 14:00 (Rapor ile birlikte)

ACIKLAMALAR:

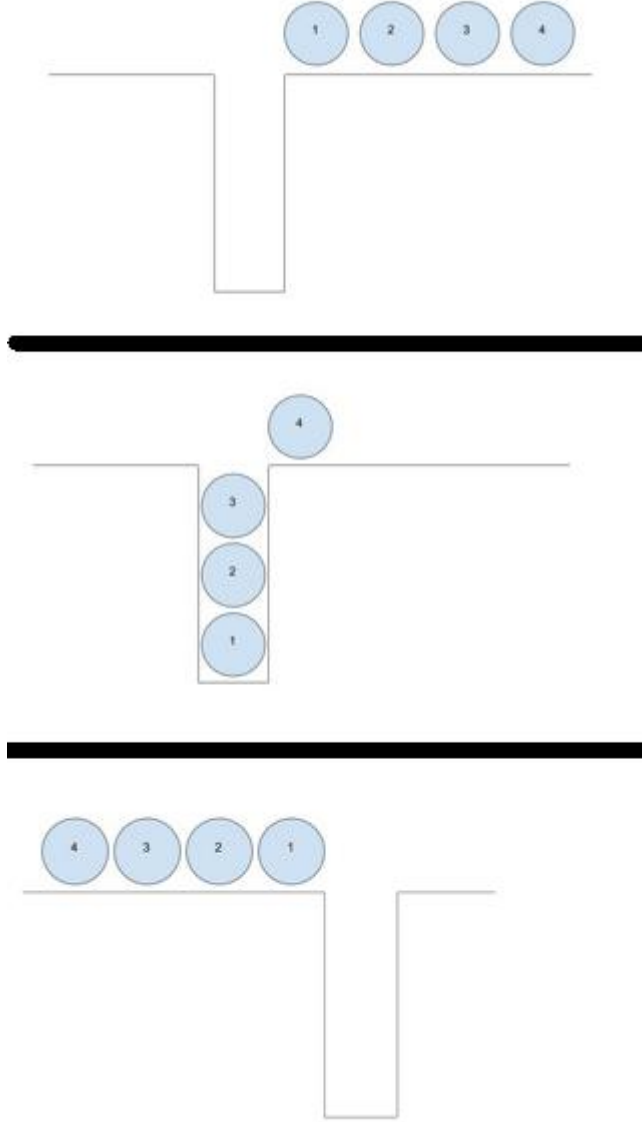
1. Ödev tek bir kod parçası halinde olmalıdır. Kodun ilk satırında açıklama olarak öğrenci ismi ve numarası verilmelidir.
2. Kodun içerisinde gerekli görülen yerlerde açıklama yapılmalıdır.
3. **goto** komutu kullanan programlar değerlendirmeye alınmaz.
4. Kaynak kod dosyası (.c uzantılı), dosya adı öğrenci numarası olacak şekilde (örneğin 131601001.c), elektronik posta ile ensararif@gmail.com adresine dosya eki (attachment) olarak belirtilen zamana kadar gönderilmelidir. Mesajın konusu **BSM2017-ÖDEV2** olmalıdır. Alıcı e-posta adresini doğru yazdığınızdan ve ödev dosyasını eklediğinizden emin olunuz.
5. Rapor, çıktı olarak kontrol tarihinde teslim edilmelidir.
6. Ödevin değerlendirmesinde; programın doğru ve eksiksiz çalışmasının yanında uygun veri tipleri ve veri yapılarının kullanılması, etkinlik, yapısal ve modüler programlama ilkelerine uygunluk dikkate alınacaktır.
7. Kopya çekildiği tespit edildiğinde, o kişiler ödevden sıfır alacaktır.
8. Kodu göndermiş olsa da kontrole gelmeyen veya raporunu getirmeyen öğrencinin ödevi değerlendirmeye alınmaz.

İstenilen program:

C programlama dili kullanarak bir program yazılması isteniyor. Program ilk çalıştırıldığında ekranda 3 seçenekten oluşan bir menü görüntülenmelidir. 1. ve 2. seçenekler istenilen program parçalarını çalıştırmak için, 3. seçenek ise çıkış içindir. Her bir işlem fonksiyon şeklinde tanımlanarak çağrılmalı ve her bir işlemten sonra program ana menüye geri dönmelidir. Yığıt ve kuyruk işlemleri için dizi ve pointer kullanılmalıdır.

- 1- **Sıralama:** Aşağıdaki şekle göre zor bir arazide 1,2,3,4 diye numaralandırılmış 4 kişi 3 derinliğine sahip bir çukurdan geçiyorlar. Geçişin sağlanması için çukurun derinliği kadar ön tarafta bulunan kişinin çukura inmesi ile diğer kişilerin güvenli bir şekilde geçmesi sağlanıyor. Daha sonra en son geçen kişi çukurdakileri sırayla çekip çıkartıyor. Geçiş sonucunda oluşan sıralama 4,3,2,1 şeklinde oluyor.

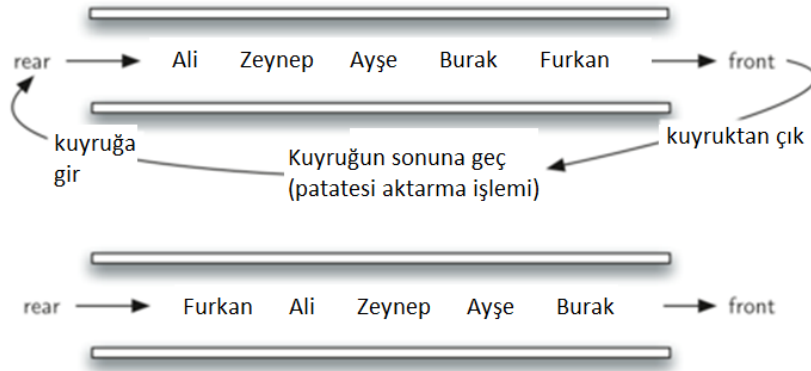
Buna göre 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20 diye numaralandırılmış kişiler sırasıyla [linkte](#) verilen derinliklere sahip 1000 adet çukurdan geçerlerse en son sıralama nasıl olur? (Her bir çukurun soyutlaması stack'dir. Stack boyutu dinamik olarak değişmektedir. Her 100. çukurdan sonraki durumu ekrana yazdırınız)



- 2- **Simülasyon: Sıcak patates (Hot potato game):** Sıcak patates bir çocuk oyunudur. Oyun için en az 3 kişi, bir sıcak patates ve müzik gerekir. Bir kişi müziği açıp kapatır ve oyunda aktif değildir. Müzik başladığında oyun başlar ve sıcak patatese sahip olan oyuncu patates el yaktığı için hemen yanındaki oyuncuya geçirir. Patatesin el değiştirmesi müzik durana dek devam eder. Müzik durduğunda patates elinde tutan oyuncu oyun dışı kalır. Bu durum 1 oyuncu kalana dek devam eder. Oyunun görsel anlatımı için kaynak: <https://www.youtube.com/watch?v=KseTjh6W9aE> <https://www.youtube.com/watch?v=qdtTjdQ0NoQ>

Bu oyunu kuyruk yapısı ile gerçekleştirmeniz gerekiyor. Sıcak patatese sahip olan oyuncu kuyruğun başındaki oyuncudur. Aşağıdaki şekilde gösterildiği gibi elindeki patatesi arkasındaki oyuncuya atarak tekrar kuyruğun sonuna geçer. Müziğin her defasında çalma süresi 10-25 sn aralığında rastgele olarak seçilir. Bir oyuncunun patatesi elinde tutma süresi ise 1-3 sn arasında rastgele seçilir. Oyuncu sayısı 15 olmalıdır. Oyunun takibi için kuyruğun durumu, müziğin çalma süresi, elenen oyuncu vb. değerler her defasında ekrana yazdırılmalıdır. Oyuncu isimlerini her defasında klavyeden girmek yerine program içinde sabit olarak

tutulması kontrol için daha uygun olacaktır. Oyun çalışmaya başladıktan sonra kazanan tek kişi belli olana kadar durmaz.



Raporda bulunması gerekenler:

- **Kapak** (Örnek kapak dersin internet sitesinden alınmalıdır)
- **Analiz** (her bir seçeneğe ait problemi kendi cümlelerinizle tanımlayıp açıklayınız)
- **Tasarım** (her bir seçeneğe ait problemin çözüm yoluna yönelik düşüncelerinizi yazınız, doğal dil ya da pseudo code ya da akış şeması ile çözüm algoritmasını veriniz)
- **Programcı Kataloğu** (programdaki prosedür ve/veya fonksiyonların ara yüzlerini verip ne yaptıklarını açıklayınız, kullanılan veri yapılarını açıklayınız, kaynak kodun çıktısını ekleyiniz)
- **Kullanıcı Kataloğu** (programın kullanım kılavuzu [çalışan programın ekran görüntüleri ile açıklama yapılabilir], programdaki kısıtlamalar vb.)

2. soruda süre tutmak (görsel olarak bekletmek) için aşağıdaki gibi bir kod örnek alınabilir:

```
#include <stdio.h>
#include<windows.h>
int main()
{
    int count = 1;
    for(;;)
    {
        // bir saniye bekle
        Sleep(1000);
        printf("%d\n", count++);
    }
    return 0;
}
```

Puanlama

Algoritma Tasarımı ve Programın Çalışması: (%70)

1. Ödev, istenilen işlerin tamamını yerine getirmelidir.
2. Gereksiz kontrollerden ve işlemlerden arınmış bir tasarım yapılmalıdır.
3. Program hatasız çalışmalıdır.
4. Programın çalışması sırasında, konuyu hiç bilmeyen kişilerin rahatlıkla anlayabilmesi için, giriş ve çıkışlarda mesajlarla bilgi verilmelidir.

Rapor Dokümantasyonu: (%30)

1. Rapor anlaşılır yazılmalı, gerekli bütün bilgileri içermeli ama gereksiz detaylarla uzatılmamalıdır. (Times New Roman ve 12 punto olarak yazılmalıdır.)
2. Ödev A4 kâğıda, yazıcıdan çıkış alınarak verilmelidir.
3. Gereksiz kod tekrarı olmamalıdır.
4. Kaynak kodun formatı düzgün (okunabilir) olmalıdır.