

ALGORİTMA TASARIMI ve ANALİZİ

2020 – 2021 Güz Dönemi

Ara Sınav 1 (Kısım – 1)

Ödev Duyurulma Tarihi : 1 Aralık 2020 Salı,

Ödev Teslim Tarihi : 11 Aralık 2020 Cuma, Saat 23:59

1. TANIMLAMA

Bu ödevde Şanlıurfa / Haliliye ilçesinde bulunan mahallelerde ikamet eden kişilerin birbirleri ile olan ilişkilerine göre Covid-19 yayılımını analiz eden C++ kodunu yazmanız beklenmektedir. Kodu yazarken graf, arama ve sıralama algoritmalarını kullanmanız gerekmektedir. İhtiyacınız dâhilinde ilgili veri yapılarını kullanabilirsiniz.

Haliliye ilçesinde bazı kişilerin Covid-19 testlerinin pozitif olduğu tespit edilmiştir. Ancak 10 gün öncesine kadar Haliliye ilçesinde herhangi bir vaka görülmemiştir. Covid-19 pozitif hastaların temasta bulunduğu kişiler filyasyon ekibi tarafından gözlem listesine alınmış ve izlenmeye başlanmıştır. Kişilerin önceki temas durumlarına göre ve il dışında bulunma durumlarına göre virüsün kimler tarafından ilçeye taşındığının tespit edilmesi amaçlanmaktadır. Bununla birlikte filyasyon ekibi ilçede yaşayan kişilerin etkileşim ve kişisel bilgilerine göre risk analizini yapacak, ilişki ve etkileşim tablolarına göre bulaşma risk haritasını oluşturacaktır.

- Tablo-1’de Haliliye ilçesinde yaşayan kişilerin bilgileri verilmiştir.
- Tablo-2’de Haliliye ilçesinde yaşayan kişilerin birbirleriyle olan arkadaşlık ve aynı evde yaşama durumları verilmiştir.
- Haliliye ilçesinde yaşayan kişilerin birbirleri ile olan 4 günlük etkileşim bilgileri etkilesim1.txt, etkilesim2.txt, etkilesim3.txt, etkilesim4.txt dosyalarında verilmiştir.
- Kişilerin birbirleriyle olan ilişki durumlarının doğru bir şekilde graf yapısına aktarılması gerekir (Bakınız Şekil-2).

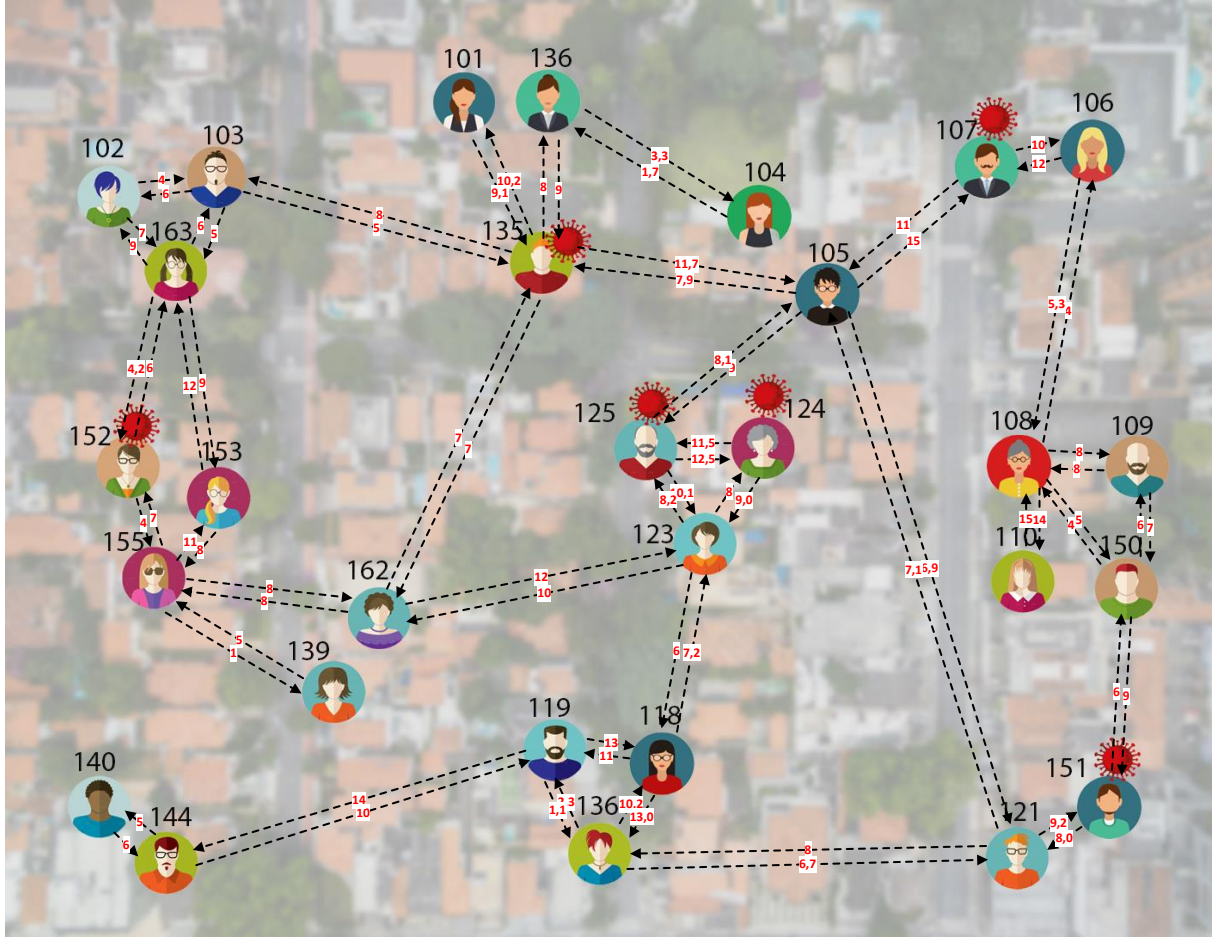


Şekil 1 – Haliliye İlçesi

2. PROBLEM

Problem adımları;

1. Şanlıurfa Haliliye ilçesinde yaşayan kişilere ait gerekli bilgiler **kisiler.txt** dosyasında verilmiştir (*bakınız Tablo-1*). Bu dosyada kişilerin id numarası, ad, cinsiyet, mahalle, yaş, Covid-19, kronik rahatsızlık durumu ve son 10 günde il dışında bulunma durum bilgileri verilmiştir.
2. Haliliye ilçesinde her bir kişi ile diğer kişiler arasında bulunan ilişki durumu **iliski.txt** dosyasında verilmiştir (*bakınız Tablo-2*). Haliliye ilçesinde bulunan bir kişi ile diğer kişiler arasındaki ilişki durumu 4 farklı türde belirtilmiştir. Bunlar; **0**:Arkadaş Değil, **1**:Arkadaş, **2**:İş arkadaşı, **3**: Aynı evde yaşayan kişiler olarak belirtilmiştir.
3. Kişilerin 4 gün içerisinde etkileşimde bulunduğu kişilerin bilgileri 4 ayrı **etkilesim.txt** dosyalarında verilmiştir (*bakınız Tablo-3, Tablo-4, Tablo-5, Tablo-6*). Kişiler arasındaki ilişki durumuna göre etkileşim puanının hesaplanması **etkilesimPuanHesapla** fonksiyonunda gösterilmiştir (*bakınız formül-1*).
4. İlçede yaşayan kişilerin 4 günlük maske kullanma durumları **maske.txt** dosyasında verilmiştir(*bakınız Tablo-7*).
5. Verilen bilgiler kullanılarak **riskPuaniHesapla** fonksiyonu ile ilçede bulunan tüm kişiler için bulaşma risk puanı hesaplanacaktır (risk puanı hesaplama **formül-2** verilmiştir).
6. Covid-19 hastalığının kişilerde görüldüğü ilk günden itibaren salgını önleme amacı ile sokağa çıkma yasağı getirilmiştir. Sokağa çıkma yasağı getirildikten sonra kişiler arasında herhangi bir temas olmadığı kabul edilecektir. Covid-19 testi pozitif olan kişilerin 4 günlük etkileşim kayıtlarında da pozitif olduğu varsayılmıştır. **temasliKisileriBul** fonksiyonu ile bir kişinin etkileşim bilgileri kullanılarak bu kişinin temas ettiği kişiler (birinci derecede temaslı olduğu) bulunup listelenecektir.
7. Kişilerin 4 günlük Etkileşim bilgileri kullanılarak etkileşim grafi oluşturulacaktır. Etkileşim grafi oluşturulduktan sonra, filyasyon ekibi belirtilen id'li kişilerden başlayarak etkileşim bilgisine göre grafi dolaşacaktır. Filyasyon ekibi uğradığı bir kişiye bir daha uğramayacaktır. Etkileşim ve ilişki tabloları kullanılarak ilçeye covid-19 hastalığını bulaştıran kişilerin bulunması için **ilkVakaTespit** fonksiyonu yazılacaktır. Bir düğümde(kişide) birden fazla temas var ise filyasyon ekibi dolaşma sırası için öncelikli olarak id numarası küçük olanı ziyaret edecektir.



Şekil 2 – 4 Günlük Etkileşim Grafi

3. OLUŞTURULACAK FONKSİYONLAR

Main.cpp

dosyaOku() : *kisiler.txt*, *iliski.txt*, *etkileşim1.txt*, *etkileşim2.txt*, *etkileşim3.txt*, *etkileşim4.txt* dosyasından bilgiler okunacaktır.

iliskisiDerecesiGoster(kaynakID, hedefID) : *iliski.txt* tablosundan kaynak ve hedef kişinin arkadaşlık bilgisi kontrol edilir. Arkadaşlık bilgisi ekrana yazdırılır. **iliskisiDerecesiGoster** fonksiyonu çalıştırılmadan önce kullanıcıdan (klavyeden) *kaynakID* ve *hedefID* bilgileri istenmeli ve daha sonra **iliskisiDerecesiGoster** fonksiyonu çalıştırılmalıdır. Arama yapılırken verimlilik esas alınacağından Binary Search kullanılmalıdır. *kaynakID*'ye sahip kişinin *hedefID*'ye sahip olan kişiye ulaşabilmesi için graf üzerinde izlenmesi gereken adımlar listelenmelidir.

etkileşimPuanıHesapla() : Kişiler arasındaki etkileşim puanı; arkadaşlık, iş arkadaşlığı, aynı evde yaşama durumu ve gün içerisindeki etkileşim bilgisine göre hesaplanacaktır (detaylar için bakınız Formül-1).

Eğer gün içerisinde görüşme(etkileşim) var ise katsayısı 1, görüşme yok ise katsayısı 0 olacaktır. Bir kişinin etkileşim puanı o kişinin 4 gün boyunca diğer kişilerle yapmış olduğu etkileşimlerin toplamıdır. Her bir kişi için tüm kişilerle olan etkileşimleri ayrı ayrı hesaplanacaktır.

$etkilesimKatsayisi =$	$\left\{ \begin{array}{cc} etkilesim & Puan \\ 0: & yok, & 0 \\ 1: & var, & 1 \end{array} \right\}$	Etkileşim bilgileri Tablo-3, Tablo-4, Tablo-5, Tablo-6 da verilmiştir.
$iliskiDurumKatsayisi =$	$\left\{ \begin{array}{cc} iliskiDurumu & Puan \\ 0 & 1 \\ 1 & 1.2 \\ 2 & 1.5 \\ 3 & 1.9 \end{array} \right\}$	İlişki durum bilgileri Tablo-2 de verilmiştir.
$maskeKatsayisi =$	$\left\{ \begin{array}{cc} maskeTakma Durumu & Puan \\ 0 & 3 \\ 1 & 1 \end{array} \right\}$	Maske kullanma bilgileri Tablo-7 de verilmiştir.
$etkileşimpuani_{i,j} = \sum_{i=1}^{kişisayısı} \sum_{j=1}^{kişisayısı} \sum_{d=1}^{günsayısı} (etkilesimKatsayisi_{i,j,d} * iliskiDurumKatsayisi_{i,j} * maskeKatsayisi_{i,d})$		

Formül-1

Bir kişi için etkileşim puanı formülde gösterildiği gibi hesaplanır. Kişinin kendisiyle olan etkileşim puanı 0 olarak alınacaktır. **etkilesimPuani_{i,j}** ve **etkilesimPuani_{j,i}** değerleri bir birinden farklı olabilir.

Örneğin, 101 id'li kişinin 102 id'li kişi ile olan etkileşim puanı şu şekilde hesaplanmaktadır;

$$etkileşim_{(101,102,1.gün)} = etkilesimKatsayisi_{101,102,1.gün} * iliskiDurumKatsayisi_{101,102} * maskeKatsayisi_{101,1.gün}$$

$$etkileşim_{(101,102,1.gün)} = 1 \times 1 \times 1 = 1$$

$$etkileşim_{(101,102,2.gün)} = 1 \times 1 \times 3 = 3$$

$$etkileşim_{(101,102,3.gün)} = 0 \times 1 \times 1 = 0$$

$$etkileşim_{(101,102,4.gün)} = 0 \times 1 \times 1 = 0$$

$$etkileşim_{(101,102)} = 1 + 3 + 0 + 0 = 4$$

etkileşimGrafOlustur() : Kişilerin 4 günlük etkileşim bilgileri kullanılarak kişiler arasındaki toplam etkileşim durumuna göre şekil-2 de gösterildiği gibi graf bağlantıları oluşturulacaktır. (**etkilesimPuaniHesapla** fonksiyonu ile elde edilen bilgiler kullanılacaktır).

riskPuaniHesapla fonksiyonu ile ilçede bulunan tüm kişiler için bulaşma risk puanı hesaplanacaktır (aşağıda risk puanı hesaplama formülü verilmiştir bakınız **formül-2**).

$yasKatsayisi =$	$\left\{ \begin{array}{ll} yasAraligi & Puan \\ 0 < yas < 5 & 0.2 \\ 5 < yas < 20 & 0.5 \\ 20 < yas < 55 & 1.0 \\ 55 < yas < 75 & 1.7 \\ 75 < yas & 2.0 \end{array} \right\}$	Yas bilgileri Tablo-1 (kişiler.txt)'de verilmiştir.
$kronikRahatsızlıkKatsayisi =$	$\left\{ \begin{array}{ll} kronikRahatsızlık & Puan \\ 0 & 1 \\ 1 & 1.7 \end{array} \right\}$	Kronik rahatsızlık bilgileri Tablo-1 de verilmiştir.
$riskPuani_i = \sum_{i=1}^{kişisayısı} \sum_{j=1}^{kişisayısı} (yasKatsayisi_i * kronikRahatsızlıkKatsayisi_i * etkileşimPuani_{i,j})$		

Formül 2

Örneğin, 101 id'li kişinin risk puanı şu şekilde hesaplanmaktadır;

$$riskpuani_{(101,102,1.gün)} = (yasKatsayisi_{101} * kronikRahatsızlıkKatsayisi_{101} * etkileşimPuani_{101,102})$$

$$riskpuani_{(101,102)} = 0.5 \times 1.7 \times 4 = 3.4$$

$$riskpuani_{(101,103)} = 0.5 \times 1.7 \times 6 = 5.1$$

...

$$riskpuani_{(101,150)} = 0.5 \times 1.7 \times 4.2 = 3.57$$

$$riskpuani_{(101)} = 3.4 + 5.1 + \dots + 3.57$$

mahalleRiskPuanHesapla(): Mahallelere gün içerisinde uğrayan ve orada ikamet eden kişilerin pozitiflik, temas durumu ve birbirleri ile etkileşim durumlarına göre mahallelerin bulaş risk oranı hesaplanacaktır.

$$mahallePuani_m = \sum_{m=1}^{Mahalle\ sayısı} \sum_{i=1}^{Mahallenin\ kişi\ sayısı} riskPuani_{m,i}$$

Formül 3

$riskPuani_{m,i}$ m'inci mahallede bulunan i'inci kişinin daha önce hesaplanmış risk puanını ifade eder.

genelSiralama() : İlçede bulunan bütün kişiler risk puanına göre büyükten küçüğe doğru sıralanmalıdır.

mahalleBulasmaRiskiSiralama() : İlçede bulunan en çok bulaşma riski olan mahalle bilgileri büyükten küçüğe doğru sıralanmalıdır.

temasliKisileriBul () : İlçede bulunan pozitif kişilerin temas ettiği kişiler gösterilmelidir(Temaslı kişilerin temas ettiği kişiler dikkate alınmayacaktır).

ilkVakaTespit() : 4 günlük toplam etkileşim tablolarından elde edilen etkileşim grafi ve kişiler tablosu kullanılarak ilçeye Covid-19 hastalığını bulaştıran kişilerin bulunması gerekmektedir. Bunun için **max heap** algoritmasını kullanınız.(il dışından gelen kişiler dikkate alınacaktır). Kişiler risk puanlarına göre büyükten küçüğe doğru sıralanacak ve kişi bilgileri **şekil 3**'te gösterildiği gibi ekrana yazdırılacaktır.

kisiID	isim	riskPuanı
121	Halil	126.25
132	Merve	125.36
148	İsmail	114.00
102	Murat	112.10

Şekil 3 İlk vaka taşıma olasılığı (temsildir)

4. GİRDİ VERİ DOSYALARI

	ID	ad	cinsiyet	Mahalle	Yaş	Covid-19	Kronik Rahatsızlık	Son 10 günde İl Dışında Bulundu mu?
1	101	Tarkan	E	1	22	1	1	0
2	102	Mustafa	E	2	54	0	0	1
3	103	Seher	K	2	65	0	1	0
4	104	Ali	E	1	88	0	1	0
5	105	Ahmet	E	3	12	1	0	0
6	106	Osman	E	5	2	0	0	0
7	107	Remziye	K	3	63	0	0	1
...
50	150	Cenk	E	3	17	0	1	1

Tablo 1 – kişi bilgileri

	101	102	103	104	105	106	107	108	109	...	150
101	-	1	1	0	1	2	0	3	0	...	0
102	1	-	0	0	3	1	1	0	1	...	1
103	1	0	-	1	0	1	1	1	1	...	0
104	0	0	1	-	0	1	1	1	1	...	0
105	1	3	0	0	-	0	1	0	1	...	1
106	2	1	1	1	0	-	1	0	0	...	0
107	0	1	1	1	1	1	-	1	1	...	0
108	3	0	1	1	0	0	1	-	1	...	0
109	0	1	1	1	1	0	1	1	-	...	0
...	-	...
150	0	0	1	1	2	1	1	1	1	...	-

Tablo 2 – kişiler arasındaki ilişki bilgileri

	101	102	103	104	105	106	107	108	109	...	150
101	-	1	0	0	1	1	0	0	0	...	1
102	1	-	0	0	0	1	1	0	1	...	1
103	0	0	-	0	0	0	0	0	0	...	0
104	0	0	0	-	0	0	0	0	0	...	0
105	1	0	0	0	-	1	0	0	0	...	1
106	1	1	0	0	1	-	0	0	0	...	0
107	0	1	0	0	0	0	-	1	1	...	0
108	0	0	0	0	0	0	1	-	1	...	0
109	0	1	0	0	0	0	1	1	-	...	0
...	-	...
150	1	1	0	0	1	0	0	0	0	...	-

Tablo 3 – 1. güne ait etkileşim bilgileri

	101	102	103	104	105	106	107	108	109	...	150
101	-	1	1	1	0	1	0	1	0	...	1
102	1	-	0	0	0	0	0	0	0	...	1
103	1	0	-	0	0	0	0	0	0	...	0
104	1	0	0	-	0	1	0	0	0	...	0
105	0	0	0	0	-	0	0	0	0	...	1
106	1	0	0	1	0	-	0	1	0	...	0
107	0	0	0	0	0	0	-	0	0	...	0
108	1	0	0	0	0	1	0	-	0	...	0
109	0	0	0	0	0	0	0	0	-	...	0
...	-	...
150	1	1	0	0	1	0	0	0	0	...	-

Tablo 4 – 2. güne ait etkileşim bilgileri

	101	102	103	104	105	106	107	108	109	...	150
101	-	0	0	0	0	1	0	0	0	...	1
102	0	-	0	1	0	0	0	0	0	...	1
103	0	0	-	0	0	0	0	0	1	...	0
104	0	1	0	-	0	0	0	0	0	...	0
105	0	0	0	0	-	0	0	0	0	...	1
106	1	0	0	0	0	-	0	0	0	...	0
107	0	0	0	0	0	0	-	0	0	...	0
108	0	0	0	0	0	0	0	-	1	...	0
109	0	0	1	0	0	0	0	1	-	...	0
...	-	...
150	1	1	0	0	1	0	0	0	0	...	-

Tablo 5 – 3. güne ait etkileşim bilgileri

	101	102	103	104	105	106	107	108	109	...	150
101	-	0	0	1	0	0	0	0	1	...	1
102	0	-	0	0	1	0	0	1	0	...	1
103	0	0	-	0	1	0	0	0	0	...	0
104	1	0	0	-	0	0	0	1	0	...	0
105	0	1	1	0	-	0	0	0	0	...	1
106	0	0	0	0	0	-	0	1	0	...	0
107	0	0	0	0	0	0	-	0	1	...	0
108	0	1	0	1	0	1	0	-	0	...	0
109	1	0	0	0	0	0	1	0	-	...	0
...
150	1	1	0	0	1	0	0	0	0	...	-

Tablo 6 – 4. güne ait etkileşim bilgileri

	ID	1.Gün	2.Gün	3.Gün	4.Gün
1	101	1	0	1	1
2	102	0	0	0	1
3	103	1	1	1	1
4	104	0	0	0	0
5	105	0	0	0	0
6	106	0	1	1	0
7	107	1	0	1	1
...
50	150	0	1	1	1

Tablo 7 – kişilerin 4 günlük maske kullanma bilgileri

5. DİKKAT EDİLMESİ GEREKENLER

1. Ödevleriniz değerlendirilirken elimizdeki farklı test dosyaları (**kişiler.txt**, **etkilesim1.txt**, **etkilesim2.txt**, **etkilesim3.txt**, **etkilesim4.txt**, **iliski.txt**, **maske.txt**,) kullanılacaktır. Bu test dosyaları sizinle paylaştığımız örnek “*.txt” dosyaları ile aynı formatta olacaktır. Bu yüzden kodunuzun farklı boyutlarda veri alıp veri yazma işlemine izin verdiğinden emin olunuz.

2. Ödev C++ programlama diliyle yazılacaktır (C++ dili, C dilini desteklediğinden kodunuzun çalışıyor olması C++’a özgü hazırlanmış anlamına gelmez. C++ dilinde hazır **STL kütüphanelerini** kullanmayınız. Kullandığınız takdirde ödev değerlendirmeye alınmayacaktır. C++ diline özgü olmasına dikkat edin, nesne kullanımı, bellekte yer ayırma gibi işlemlerde dikkatli olun, malloc yerine new, free yerine delete gibi).

3. Ödevi hazırlarken doğru veri yapısını ve doğru algoritmayı kullandığınızdan emin olun!

4. Ödevlerinizi **algoritmatasarimodevi@gmail.com** adresine aşağıdaki formatta gönderiniz!

Örneğin, ödevi gönderen kişi Şadiye Aydoğan olsun ve öğrenci numarası 200500020 olsun.

Öncelikle, ödevde main dosya ismi **200500020_SadiyeAydogan _odev1.cpp** olacak. Ödevinizin bulunduğu klasörünüzün ismi de;

200500020_SadiyeAydogan _odev1.zip

şeklinde olacaktır. Yani göndereceğiniz ödev dosyasını, kendi öğrenci numaranız ve ad soyadınıza göre örnekte gösterildiği gibi adlandırın (dosyayı isimlendirirken Türkçe karakter kullanmayınız, boşluk bırakmayınız, ayırmayı alt çizgiler kullanarak yapınız).

5. Gönderdiğiniz e-postalarınızda karışıklık olmaması adına, ödevinizi gönderirken posta başlığı (e-mail başlığı): **Algoritma Analizi ve Tasarımı Ödev-1** şeklinde olmasına dikkat ediniz.

6. Ödevlerinizi en geç teslim tarihi saatine (ilk sayfada) kadar göndermiş olduğunuzdan emin olunuz. **Ödev teslim tarih saatinden sonra gönderilen ödevler değerlendirilmeye alınmayacaktır!**

7. Bu ödevin, genel dönem puanına etkisi **20 puandır**. Kopya ödev, İnternette hazır kopyaya yapıştır ödev teslim etmeyiniz. Kopya yazılımlarında benzerlik olacağından kendiniz yazınız. Bu tür durumların tespitinde öğrencinin **disiplin kuruluna sevki** yapılacaktır. O yüzden ödevinizi kendiniz yapınız!

8. Göndereceğiniz kodun Linux konsol (terminal) ortamında çalışıyor olduğuna dikkat ediniz.

9. Ödevi Gönderirken sadece “*.cpp”, “*.h” dosyalarınızı gönderiniz. Ödevi gönderirken proje halinde göndermeyiniz. Aksi Takdirde ödev değerlendirmeye alınmayacaktır. Ayrıca ödevi gönderirken lütfen “*.exe” dosyası göndermeyiniz.