

Ankara Üniversitesi
Bilgisayar Mühendisliği
BLM2067 LAB 1

	A	B	C	D	E	F	G	H	I	J
A	0	40	8	27	-1	-1	-1	-1	-1	-1
B	-1	0	-1	-1	6	-1	10	-1	-1	-1
C	-1	7	0	12	-1	2	-1	-1	-1	-1
D	-1	-1	-1	0	-1	9	-1	-1	-1	-1
E	-1	-1	10	-1	0	12	11	-1	-1	-1
F	-1	-1	-1	-1	-1	0	-1	13	14	-1
G	-1	-1	-1	-1	-1	-1	0	20	-1	15
H	-1	-1	-1	-1	17	-1	-1	0	-1	18
I	-1	-1	-1	19	-1	-1	-1	21	0	20
J	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

Bu lab kapsamında C programlama dilinde şehirler arasında en kısa yolu bulmanız beklenmektedir. Öncelikle 10 x 10 boyutunda iki boyutlu bir tamsayı dizisi oluşturmalı ve bu dizinin değerlerini kullanıcıdan almalısınız. Bu matris şehirler arasındaki uzaklığı gösterecektir. “-1” değeri iki şehir arasında yol bulunmadığını ifade etmektedir. Daha sonra en kısa yolun bulunacağı iki şehri kullanıcıdan almalı ve bu şehirler ile iki boyutlu diziyi aşağıda prototipi verilen fonksiyona göndermelisiniz. Fonksiyon prototipini bozmanız labınızın değerlendirilmemesiyle sonuçlanacaktır.

int* functionFindMin(int* array, int row, int cols, char source, char destination);

Fonksiyonunuz en kısa yolu bulduğunda, iki şehir arasındaki şehirleri ve en kısa mesafeyi main fonksiyonuna göndermelidir. Gönderilen değerler ekrana “-” ile yazdırılacaktır. İki şehir arasında bağlantı yoksa ekrana “iki şehir arası gidilemez” uyarısı yazdırılmalıdır.

1- Örneğin kullanıcının girdiği şehirler A ve D olsun. A dan D’ ye bir çok alternatif yol vardır.

A-D 27

A-C-D 20

A-C-F-I-D 43

A-B-E-C-D 68

En kısa yol A-C-D dir ve çıktınız A-C-D 20 şeklinde olacaktır.

1- Örneğin kullanıcının girdiği şehirler F ve J olsun. F den J’ ye bir çok alternatif yol vardır.

F-H-J 31

F-I-J 34

F-H-E-G-J 56

F-I -H-J 53

En kısa yol F-H-J dir ve çıktınız F-H-J 31 şeklinde olacaktır.

Yukarıdaki örneklerde verilenler dışında bir çok alternatif yol daha olabilir. Sadece doğru sonuçla beraber birkaç tanesi verilmiştir.

Çalışmanızı OgrenciNumarasi.c olarak isimlendirerek sisteme yükleyiniz. Programınızın Ubuntu ortamında çalıştığından emin olunuz. Doğru çıktı formatı için size verilen örnek girdi ve çıktı dosyalarını dikkatle inceleyiniz. Programınızın doğruluğunu kontrol etmek için aşağıdaki işlemleri gerçekleştirebilirsiniz.

1. Programınızı gcc komutunu kullanarak derleyin.
2. `./a.out < input1.txt >myOutput1.txt` komutunu kullanarak programınızı input1.txt dosyası ile çalıştırın ve çıktınızı myOutput1.txt dosyasına kaydedin.
3. `diff myOutput1.txt output1.txt` komutunu kullanarak kendi çıktınız ile olması gereken çıktıyı otomatik olarak karşılaştırın. Bu komutu girdikten sonra aşağıdaki gibi ekranda bir uyarı çıkmıyorsa bu, bu değerler için programınız doğru çalışıyor demektir. Eğer komutu girdikten sonra komut sisteminde uyarı görüyorsanız bu çıktınızda problem olduğunu gösterir.

A screenshot of a terminal window with a dark background. The window title is 'pk@dellPC: ~/Desktop/BLM'. The prompt is 'pk@dellPC:~/Desktop/BLM\$'. The command 'diff myOutput1.txt output1.txt' has been entered and executed. The output is empty, indicating no differences were found between the two files. The cursor is on the line following the command.

```
pk@dellPC:~/Desktop/BLM$ diff myOutput1.txt output1.txt
pk@dellPC:~/Desktop/BLM$
```

4. 2. ve 3. maddelerindeki komutları size verilen bütün girdi dosyaları için de deneyin.
5. Kendi oluşturacağınız farklı girdiler için de programınızı test edin. Size verilen girdi dosyaları ile değerlendirme sırasında kullanılan girdi dosyalarının birbirinden farklılık gösterebileceğini unutmayın.