

## MINI PROJE #2

DUE : Ekim 29, 2021, 23:59

Teslim : uzak.etu.edu.tr

Notlar:

- Programınızı hatalı durumları dikkate alarak düzenleyiniz.
- Çalışmanız bireyseldir,
- Zamanında teslim ediniz. Geç teslim kabul edilmeyecektir.

1) Bu soruda Caesar şifrelemesini kıran bir C kodu yazınız. Dosyanızın ismini "decrypt.c" yapınız. Programınız 3 tane argüman almalıdır. '-r', referans metin belgesinin ismini içerir. '-c', şifrelenmiş metin belgesinin ismini içerir. '-p', şifresi kırılan metnin yazılacağı belgenin ismini belirler. Metin belgelerini sadece İngilizce karakterler içerir ve değişken uzunlukta olabilirler. Referans metin belgesi, İngilizce yazılmış ve şifrelenmemiş çok uzun bir metin içerir. Bu belgeyi inceleyerek İngilizce dilinde hangi harfin hangi sıklıkta kullanıldığını analiz etmeniz gerekmektedir. Şifrelenmiş metin belgesi, Caesar şifrelenmesi ile şifrelenmiş bir metni içerir ve kayırma miktarı belli değildir. Bu metni inceleyerek hangi harflerin hangi sıklıkta kullanıldığını analiz etmeniz gerekmektedir ve referans metninin analiziyle karşılaştırıp, kayırma miktarını bulmanız gerekmektedir. Kayırma miktarını bulduktan sonra şifrelenmiş metni orijinaline çevirip çıktı metin belgesine yazmanız gerekmektedir.

- Argümanların sırası değişebilir.
- Referans metnindeki boşlukları, sayıları ve noktalama işaretlerini ihmal etmelisiniz.
- Şifrelenmiş metinde boşluk ya da noktalama işareti bulunmayacaktır.
- Referans ve şifrelenmiş metin dosyaları bulunamazsa, program hata verip çıkmalıdır.
- Çıktı dosyası önceden yazılıysa, yeniden yazılmalıdır.
- Frekans analizi yaparken, en sık kullanılan ve en seyrek kullanılan harfleri dikkate alınız. İkisinden bulduğunuz kayırma miktarları eşit ise çözümlenmiş metni "<-p>" ile belirtilen belgeye yazınız. İkisinden bulduğunuz kayırma miktarları eşit değil ise en sık kullanılan harfin verdiği çözümlenmiş metni "<-p>\_0.txt" dosyasına, en seyrek kullanılan harfin verdiği çözümlenmiş metni "<-p>\_1.txt" dosyasına yazınız.
- Programınızın işleyişi aşağıdaki örnekler şeklinde olmalıdır.
- Örnek referans ve şifrelenmiş metinleri resource altında bulabilirsiniz.
- Dosya işlemleri için [https://www.tutorialspoint.com/cprogramming/c\\_file\\_io.htm](https://www.tutorialspoint.com/cprogramming/c_file_io.htm) sayfasını inceleyebilirsiniz.
- Pointer kullanımı yasaktır (FILE pointer hariç).

Örnek:

```
~$./decrypt -r refa.txt -c ciphertext.txt -p plaintext.txt
```

refa.txt dosyası bulunamadı.

```
~$./decrypt -r ref.txt -c cipahertext.txt -p plaintext.txt
```

cipahertext.txt dosyası bulunamadı.

```
~$./decrypt -r refa.txt -c cipahertext.txt -p plaintext.txt
```

refa.txt dosyası bulunamadı.

ciphertext.txt dosyası bulunamadı.

```
~$./decrypt -r ref.txt -c ciphertext1.txt -p plaintext1.txt
```

Kayıma miktarı 5 olarak tespit edilmiştir.

Çözümlemiş metin plaintext1.txt dosyasına yazılmıştır.

```
~$./decrypt -r ref.txt -c ciphertext2.txt -p plaintext2.txt
```

Kayıma miktarı 3 (en sık harf) ve 5 (en seyrek harf) olarak tespit edilmiştir.

Çözümlemiş metin plaintext2\_0.txt ve plaintext2\_1.txt dosyalarına yazılmıştır.

(ref.txt ve ciphertext.txt Labratory Resources altında eklenmiştir)

## 2) Connect 4

Bu soruda 2 oyunculu bir connect 4 oyunu C dilinde kodlamanız isteniyor.

Programınızın adı connect4.c olmalıdır.

Proje verilış tarihine kadar derste görülen metodlar haricindeki metodların/kütüphanelerin kullanımı, piazzada metod bazında belirtilmediği takdirde, yasaktır.

Her zaman 1. Oyuncunun R olacağı varsayılabılır.

Bu program aşağıdaki örneklerdeki gibi çalışmalıdır (Sarı ile belirtilen alanlar kullanıcıdan alınmaktadır):

```
~$./connect4
```

**The game has started!**

X	0	1	2	3	4	5	6	X
0	-	-	-	-	-	-	-	0
1	-	-	-	-	-	-	-	1
2	-	-	-	-	-	-	-	2
3	-	-	-	-	-	-	-	3
4	-	-	-	-	-	-	-	4
5	-	-	-	-	-	-	-	5
X	0	1	2	3	4	5	6	X

**Player 1's Turn (R):**

**Enter a move: 3**

X	0	1	2	3	4	5	6	X
0	-	-	-	-	-	-	-	0
1	-	-	-	-	-	-	-	1
2	-	-	-	-	-	-	-	2
3	-	-	-	-	-	-	-	3
4	-	-	-	-	-	-	-	4
5	-	-	-	R	-	-	-	5
X	0	1	2	3	4	5	6	X

**Player 2's Turn (Y):**

**Enter a move: 3**

X	0	1	2	3	4	5	6	X
0	-	-	-	-	-	-	-	0
1	-	-	-	-	-	-	-	1
2	-	-	-	-	-	-	-	2
3	-	-	-	-	-	-	-	3
4	-	-	-	Y	-	-	-	4
5	-	-	-	R	-	-	-	5
X	0	1	2	3	4	5	6	X

---

Oyun tahtada yer kalmayınca ya da oyunculardan biri dikey, yatay, veya çaprazda 4 lü yaptığıında bitmelidir.

```
X 0 1 2 3 4 5 6 X
0 - - - - - - - 0
1 - - - - - - - 1
2 - - - - - - - 2
3 - - - Y - - - 3
4 - - - Y Y - - 4
5 - - - R R R - 5
X 0 1 2 3 4 5 6 X
```

Player 1's Turn (R):  
Enter a move: 6

```
X 0 1 2 3 4 5 6 X
0 - - - - - - - 0
1 - - - - - - - 1
2 - - - - - - - 2
3 - - - Y - - - 3
4 - - - Y Y - - 4
5 - - - R R R - 5
X 0 1 2 3 4 5 6 X
```

Player 1 has won!

(Exit the program here)

~\$

---

Dolu olan sütuna veya oyun tahtasında yer almayan hamleler yapılmak istenirse uygun hata mesajı basılarak girdi tekrar istenmelidir.

Geçersiz hamle örneği:

```
X 0 1 2 3 4 5 6 X
0 - - - - - - - 0
1 - - - - - - - 1
2 - - - - - - - 2
3 - - - Y - - - 3
4 - - - Y Y - - 4
5 - - - R R R - 5
X 0 1 2 3 4 5 6 X
```

Player 1's Turn (R):  
Enter a move: 7

Entered move is not valid!

```
X 0 1 2 3 4 5 6 X
0 - - - - - - - 0
1 - - - - - - - 1
2 - - - - - - - 2
3 - - - Y - - - 3
4 - - - Y Y - - 4
5 - - - R R R - 5
X 0 1 2 3 4 5 6 X
```

Player 1's Turn (R):

**Enter a move:**

Dolu sütuna hamle örneği:

```
X 0 1 2 3 4 5 6 X
0 - - - Y - - - 0
1 - - - R - - - 1
2 - - - Y - - - 2
3 - - - R Y - - - 3
4 - - - Y Y - - - 4
5 - - - R R R - - 5
X 0 1 2 3 4 5 6 X
```

**Player 1's Turn (R):**

**Enter a move: 3**

**The column is full, try again!**

```
X 0 1 2 3 4 5 6 X
0 - - - Y - - - 0
1 - - - R - - - 1
2 - - - Y - - - 2
3 - - - R Y - - - 3
4 - - - Y Y - - - 4
5 - - - R R R - - 5
X 0 1 2 3 4 5 6 X
```

**Player 1's Turn (R):**

**Enter a move:**

3) JSON Nedir? (<https://www.guru99.com/json-tutorial-example.html>  
<https://support.oneskyapp.com/hc/en-us/articles/208047697-JSON-sample-files>)

Örnek Json Dosyası (a.json) :

```
{ "shop": {
  "book": [
    { "category": "encyclopedia",
      "title": "Encyclopedia of Management",
      "authors": "Marilyn Helms",
      "price": 120.45,
      "year" : 2021
    },
    { "category": "history",
      "title": "Children of Ash and Elm: A History of the Vikings",
      "authors": "Neil Price",
      "price": 30.49,
```

```

    "year" : 2019
  },
  { "category": "biography",
    "title": "Artists: Their Lives and Works",
    "authors": "Ross King",
    "isbn": "0-993-34532-9",
    "price": 28.99,
    "year" : 2019
  },
  { "category": "biography",
    "title": " Steve Jobs: A Biography (Greenwood Biographies)",
    "authors": "J. R. R. Tolkien",
    "isbn": "9-781-61069-4",
    "price": 44.00,
    "year" : 2017
  }
],
"stuffedtoy" : [
  { "name" : "teddy bear",
    "weight": "350",
    "color": "brown",
    "price": 48.52
  }
],
"buildingblock" : [
  { "name" : "classic deep sea ship",
    "brand" : "lego" ,
    "size": "35x42x5cm",
    "price": 58.22
  },
  { "name" : "classic ninjago serpent",
    "brand": "lego" ,
    "size": "35x42x5cm",
    "price": 38.22
  }
],
kite:{price: 3}
}

```

Herhangi bir json dosyasını input olarak alıp json dosyasının yapısını mevcut tüm path bilgilerini listeleyen bash scriptini yazınız. (that gives the structure of the json file as the set of paths).

Çağırma şekli : struct a.json

\$.shop  
\$.shop.book[]  
\$.shop.book[].category  
\$.shop.book[].authors  
\$.shop.book[].title  
\$.shop.book[].price  
\$.shop.book[].year  
\$.shop.book[].isbn  
\$.shop.stuffedtoy[]  
\$.shop.stuffedtoy[].name  
\$.shop.stuffedtoy[].weight  
\$.shop.stuffedtoy[].color  
\$.shop.stuffedtoy[].price  
\$.shop.buildingblock[]  
\$.shop.buildingblock[].name  
\$.shop.buildingblock[].brand  
\$.shop.buildingblock[].size  
\$.shop.buildingblock[].price  
\$.shop.kite  
\$.shop.kite.price