

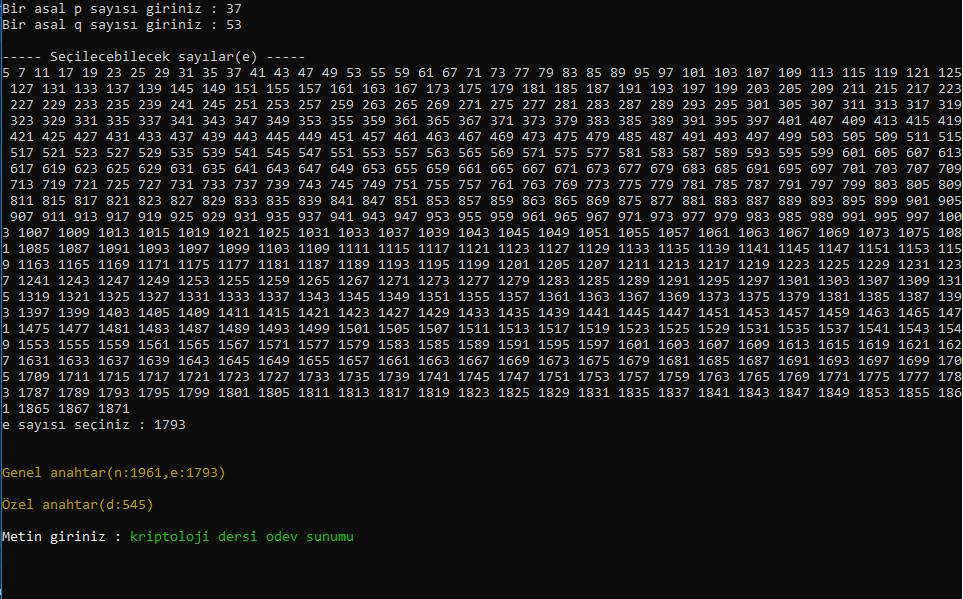
**Bilgi Güvenliği ve Kriptoloji**

Öğrenci : ***Sahil Rzayev 399973***

**RSA**

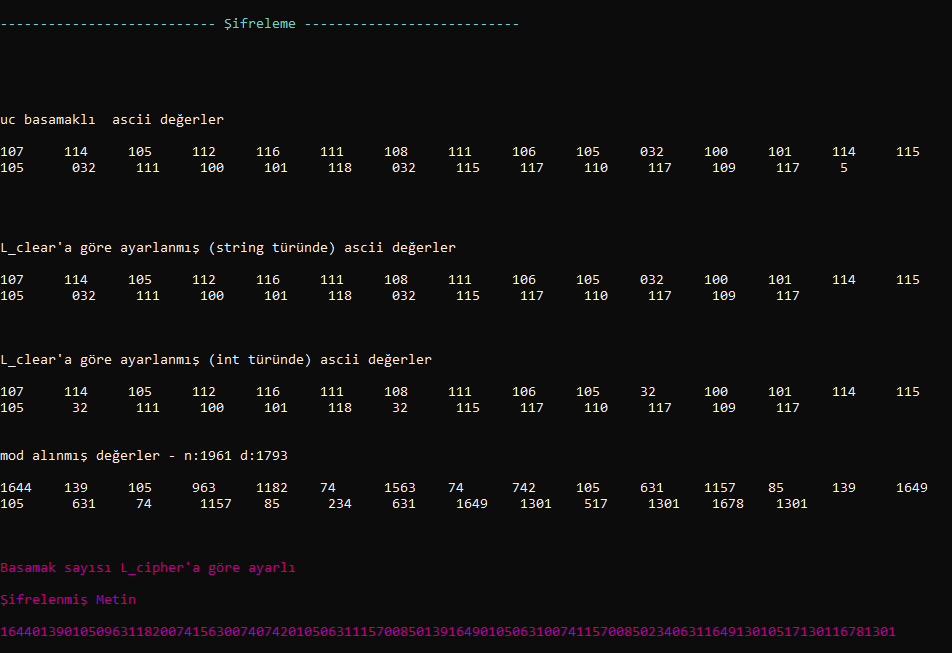
**1.Anahtar Üretimi**

* **p** ve **q** asal sayıları kullanıcı tarafından ekrana giriliyor.
* Buna göre de program tüm uygun **e** değerlerinin bir listesini kullanıcıya sunuyor ve kullanıcı seçtiği **e** değerine göre **d** değeri de otomatik program tarafından hesaplanıyor.
* Tüm bunlar sonucunda ekrana **Genel** ve **Özel** **anahtar** değerleri basılıyor.



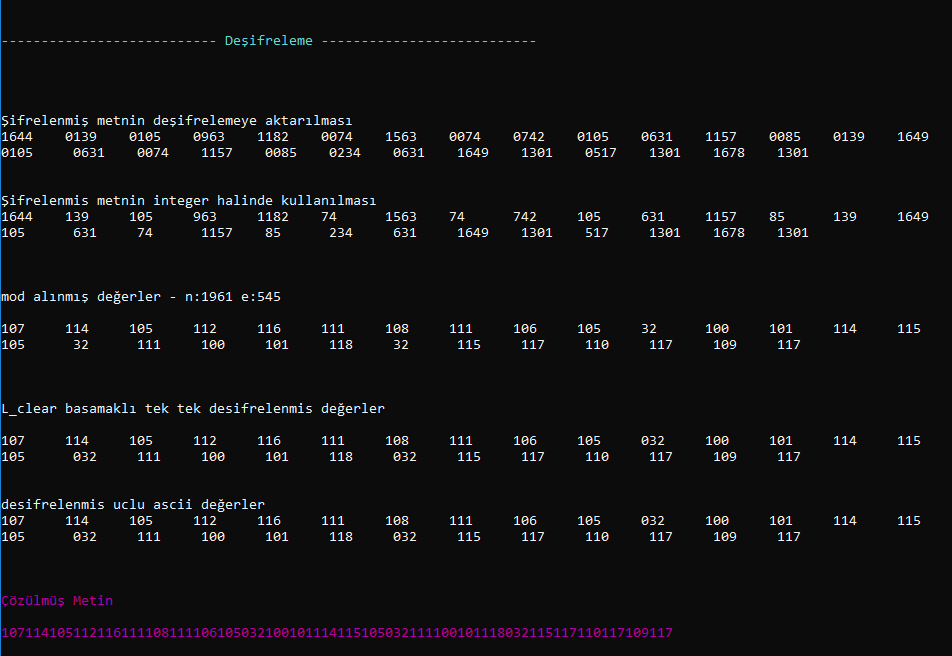
**2.Şifreleme**

* **Genel anahtarla(n,e)** şifreleme yapıldı(gönderici genel anahtarla mesajı şifreliyor).
* ASCII değerlerinin basamak sayıları **L\_clear** ve **L\_cipher**’e (şifreli metnindeki ascii değerlerinin basamak sayısı) göre ayarlandı.



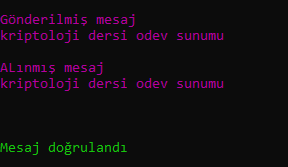
**3.Deşifreleme**

* **Özel anahtarla(d)** deşifreleme yapıldı(alıcı özel anahtarla mesajı deşifreliyor).
* ASCII değerlerinin basamak sayıları **L\_clear** ve **L\_cipher**’e göre ayarlandı



**4.Mesaj Doğrulaması**

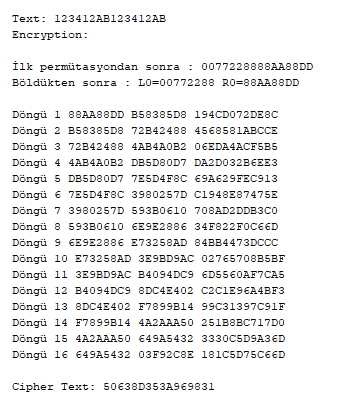
* Göndericinin gönderdiği mesajla alıcının elde ettiği mesaj aynıdırsa yeşil rengli “Mesaj doğrulandı” yazısı ekrana basılıcak.



**DES**

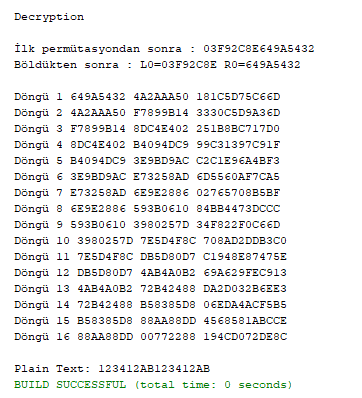
**1.Şifreleme**

* Kullanıcı metni giriyor. Girilen metin ilk olarak **başlangıç permütasyon**(**IP**) uygulanıyor. Sonra 32 bitlik iki alt bloğa bölünüyor.
* Daha sonra 16 döngü uygulanıyor. Her döngüde anahtar bitlerinden 56’dan 48’i seçiliyor.Bloğun sağ yarısı **genişletme permütasyonu**(**EP**) ile 48 bite(32 bitden 48 bite genişletme) genişletip, **XOR** işlemi uygulanarak anahtarla birleştiriliyoruz.
* Elde edilen sonuç **S-box**’lara gönderilerek 32 yeni bit üretiliyor ve yeniden tekrar **permütasyon**(**D-box**) uygulanıyor. Elde edilen sonuç bloğun sol yarısı ile birleştirilerek **XOR** yapılır ve **sağ blo**k olarak atanır.
* **Sol blok** sağ bloğun döngüye başlanğıcındaki halidir.
* 16 döngüden sonra bir sonraki 64 bitlik bloğa geçilir.En sonda başlangıçta uygulanan permütasyonun tersi(**IP1**) uygulanır.



**2.Deşifreleme**

* Şifrelemeden alınan şifrelenmiş metin çözülmesi için deşifrelemeye dahil ediliyor. Burada da şifrelemdeki adımların aynısı uygulanıyor. Tek fark 16 dögüden uygulanan anahtar sırası tersden uygulanıyor.



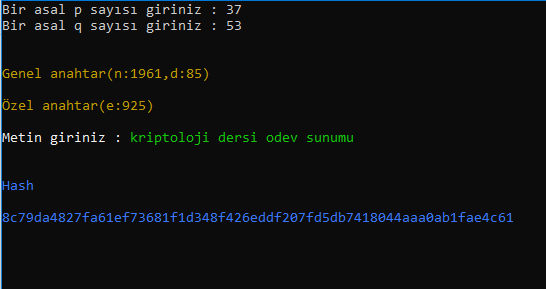
**DSA (Sayısal İmza)**

DSA Asimmetrik şifreleme için RSA algoritması kullanıldı. **DSA**’nın **RSA**’dan farkı:

* Kullanıcı **RSA**’da girdiği metnin ASCII değerlerine göre **RSA** hesaplanırken, **DSA**’da kullanıcı’nın girdiği metnin hash’lenmiş değerlerinin ASCII değerlerine göre **DSA** hesaplanıyor.
* Diğer bir farkı **RSA**’da **genel anahtarla** *şifreleme* yapıp, **özel anahtarla** *deşifreleme* yapıyoruz, ama **DSA**’da **özel anahtarla** *şifreleme* yapıp, **genel anahtarla** *deşifreliyoruz*.

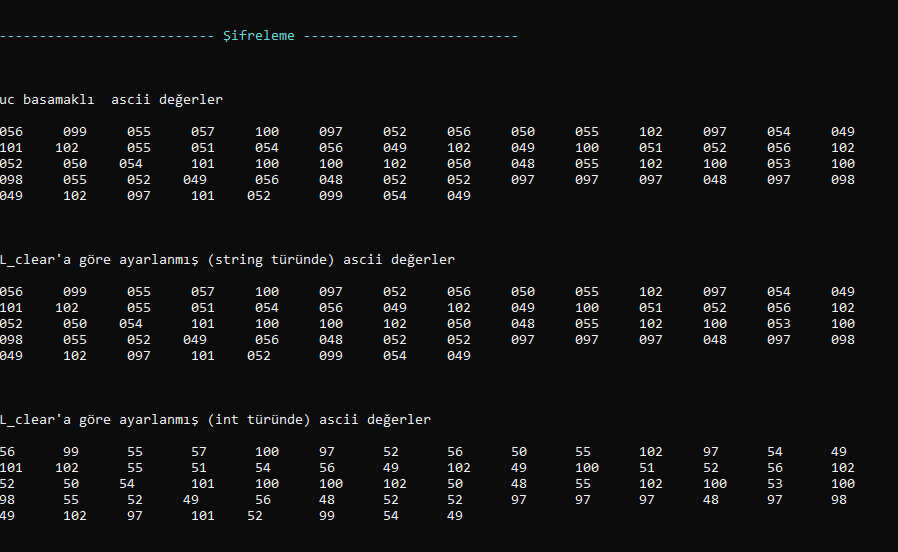
**1.Anahtar Üretimi**

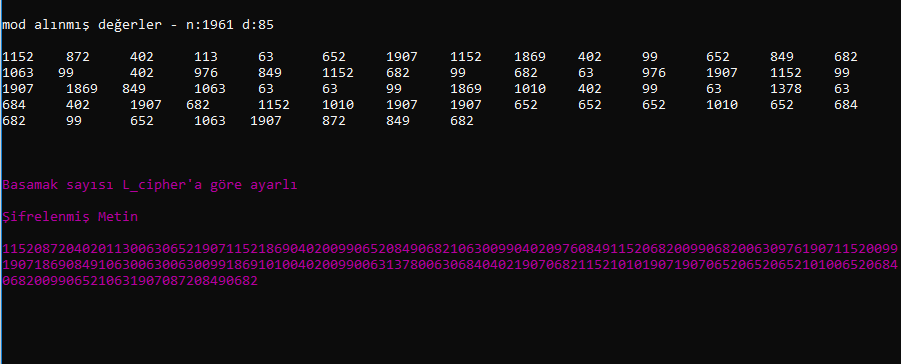
* **p** ve **q** asal sayıları kullanıcı tarafından ekrana giriliyor.
* **d** ve **e** değerleri otomatik program tarafından hesaplanıyor.
* Tüm bunlar sonucunda ekrana **Genel(n,e)** ve **Özel(d)** **anahtar** değerleri basılıyor.



**2.Şifreleme**

* **Özel anahtarla(d)** şifreleme yapıldı(gönderici özel anahtarla mesajı şifreliyor).
* ASCII değerlerinin basamak sayıları **L\_clear** ve **L\_cipher**’e (şifreli metnindeki ascii değerlerinin basamak sayısı) göre ayarlandı.





**3.Deşifreleme**

* **Genel anahtarla(n,e)** deşifreleme yapıldı(alıcı genel anahtarla mesajı deşifreliyor).
* ASCII değerlerinin basamak sayıları **L\_clear** ve **L\_cipher**’e göre ayarlandı.

