```
(* Wybieramy p i q *)
p = 61;
q = 53;

(*Obliczamy n i funkcję Eulera phi(n)*)
n = p * q;
phi = EulerPhi[n];
      funkcja Eulera

(* Losujemy e spełniające warunki *)
eKandydaci = Select[Range[2, phi - 1], CoprimeQ[#, phi] &];
             wybier… zakres          względnie pierwsze?
e = RandomChoice[eKandydaci];
    losowy wybór

(*Obliczamy wykładnik prywatny d*)
d = ModularInverse[e, phi];
    odwrotność modularna

(*Definiujemy funkcje szyfrowania i deszyfrowania*)
szyfrowanie[m_Integer] := PowerMod[m, e, n];
              liczba całkowita  potęga modulo
deszyfrowanie[c_Integer] := PowerMod[c, d, n];
                liczba całkowita  potęga modulo

(* Przykład użycia *)
wiadomosc = 65;
szyfr = szyfrowanie[wiadomosc];
rozszyfrowane = deszyfrowanie[szyfr];
```

In[13]:=

```
Print["p = ", p];
drukuj
Print["q = ", q];
drukuj
Print["n = ", n];
drukuj
Print["phi(n) = ", phi];
drukuj
Print["e (losowe) = ", e];
drukuj
Print["d = ", d];
drukuj
Print["Zaszyfrowana wiadomość = ", szyfr];
drukuj
Print["Odszyfrowana wiadomość = ", rozszyfrowane];
drukuj
```

```
p = 61
q = 53
n = 3233
phi(n) = 3120
e (losowe) = 1441
d = 721
Zaszyfrowana wiadomość = 2383
Odszyfrowana wiadomość = 65
```