1. Решить уравнение ax + b =c (modp) обычным методом с использованием обратного элемента: a = 5; b = 9; c = 29; p = 31.

In[15]:=
```
a = 5;
b = 9;
p = 31;
c = 29;
Mod[(c - b) * PowerMod[a, -1, p], p]
```

Out[19]= 4

2. Решить это же уравнение ax + b =c (modp) с применением функции Solve[a*x + b == c, {x}, Modulus -> 31].

In[20]:=
```
Solve[a * x + b == c, {x}, Modulus → p]
```

Out[20]= { { x → 4 } }

3. Восстановить сообщение M в пороговой схеме (4,12) по четырем долям, индивидуальные задания по номеру в списке группы приведены в Табл. 1.

In[21]:=
```
nv = Mod[4, 5] + 1
```

Out[21]= 5

In[22]:=
```
p = 1759
```

Out[22]= 1759

In[23]:=
```
Solve[a1 * 4^3 + 4^2 * b1 + 4 * c1 + M == 617 &&
  a1 * 5^3 + 5^2 * b1 + 5 * c1 + M == 828 &&
  a1 * 6^3 + 6^2 * b1 + 6 * c1 + M == 534 &&
  a1 * 9^3 + 9^2 * b1 + 9 * c1 + M == 1296,
 {a1, b1, c1, M},
 Modulus → p]
```

Out[23]=
```
{{M → 1206, c1 → 1388, b1 → 338, a1 → 1661}}
```

4. Провести оценку числа возможных решений при наличии только трех долей.

In[24]:=
```
Solve[a1 * 4^3 + 4^2 * b1 + 4 * c1 + M == 617 &&
  a1 * 5^3 + 5^2 * b1 + 5 * c1 + M == 828 &&
  a1 * 6^3 + 6^2 * b1 + 6 * c1 + M == 534,
 {a1, b1, c1, M},
 Modulus → p]
```

••• Solve : Equations may not give solutions for all "solve" variables.

Out[24]=
```
{{M → 1639 a1, c1 → 2 (802 + 37 a1), b1 → 627 + 1744 a1}}
```

Данная система имеет число решений, равное модулю p. Однако, надежность с точки зрения защиты информации в том, что любое полученное решение потенциально может быть верным.

5. Разработать криптосистему с тремя открытыми ключами. Требования к системным параметрам такие же как и в RSA (с выполнением всех необходимых проверок) : k1*k2*k3 = 1 mod ((n)).

In[25]:=
```
RSA[P0_, Q0_] := Module[{P = P0, Q = Q0},
  N1 = P * Q;
  phi = EulerPhi[N1];
  flag = True;
  While[flag == True, k1 = RandomInteger[{1, phi}];
   k2 = RandomInteger[{1, phi}];
   If[GCD[k1 * k2, phi] == 1, flag = False]];
  k3 = PowerMod[k1 * k2, -1, phi];
  {k1, k2, k3}]
```

In[26]:=
```
n = 4;
Q = Prime[10 000 - n];
P = Prime[10 000 + n];
```

In[29]:=
```
SeedRandom[4];
{k1, k2, k3} = RSA[P, Q];
Print[k1, "\n", k2, "\n", k3]
```

```
1 081 419 965
2 195 413 397
9 455 862 961
```

In[32]:=
```
GCD[k1 * k2 * k3, phi]
```

Out[32]= 1

6. Представить свою фамилию в числовом эквиваленте, подписать на первом ключе (СК1) и выполнить проверку (восстановить текст-фамилию) с применением ключей К2 и К3.

In[33]:=
```
fam = "балашов";
fam = ToCharacterCode[fam] - 1071
c = PowerMod[fam, k1, N1]
FromCharacterCode[PowerMod[c, k2 * k3, N1] + 1071]
```

Out[34]= {2, 1, 12, 1, 25, 15, 3}

Out[35]= {7 215 198 557, 1, 10 265 415 630, 1, 10 853 269 938, 9 624 695 886, 10 347 006 833}

Out[36]= балашов

7. Подписать СК1 на втором ключе и восстановить текст-фамилию на К3.

In[37]:=
```
c2 = PowerMod[c, k2, N1]
FromCharacterCode[PowerMod[c2, k3, N1] + 1071]
```

Out[37]= {6 097 199 697, 1, 7 642 895 026, 1, 7 417 272 710, 5 046 921 833, 9 411 118 502}

Out[38]= балашов