А-08-19, Балашов.

ЛР 7

1. Подготовить открытый текст для шифрования – строку, содержащую фамилию, имя и отчество.

In[21]:=
```
plaintext = "балашовсавваарсеньевич"
```

Out[21]=
```
балашовсавваарсеньевич
```

2. Перевести открытый текст в последовательность кодов ToCharacterCode[“string”,”encod - ing”] со спецификацией “ISOLatinCyrillic”.

In[22]:=
```
plainTextCode = ToCharacterCode[plaintext, "ISOLatinCyrillic"]
```

Out[22]=
```
{209, 208, 219, 208, 232, 222, 210, 225, 208, 210,
 210, 208, 208, 224, 225, 213, 221, 236, 213, 210, 216, 231}
```

3. Провести обратное преобразование (FromCharacterCode[,]) кодов в символы с той же спецификацией.

In[23]:=
```
FromCharacterCode[plainTextCode, "ISOLatinCyrillic"]
```

Out[23]=
```
балашовсавваарсеньевич
```

4. Провести поразрядное сложение BitXor[,]списка кодов из п.2 со случайным числом из диапазона целых [100,200].

In[24]:=
```
key = RandomInteger[{100, 200}]
cipherTextCode =
 Table[BitXor[plainTextCode[[i]], key], {i, 1, Length[plainTextCode]}]
```

Out[24]=
```
136
```

Out[25]=
```
{89, 88, 83, 88, 96, 86, 90, 105, 88, 90,
 90, 88, 88, 104, 105, 93, 85, 100, 93, 90, 80, 111}
```

5. Преобразовать коды в символы и зафиксировать результат.

In[26]:=
```
cipherText = FromCharacterCode[cipherTextCode, "ISOLatinCyrillic"]
```

Out[26]=
```
YXSX`VZiXZZXXhi]Ud]ZPo
```

6. Провести повторное поразрядное сложение шифртекста п.4 с тем же самым случайным числом и восстановить открытый текст.

In[27]:=
```
plainTextCodeCipher =
 Table[BitXor[cipherTextCode[[i]], key], {i, 1, Length[cipherTextCode]}]
FromCharacterCode[plainTextCodeCipher, "ISOLatinCyrillic"]
```

Out[27]=
```
{209, 208, 219, 208, 232, 222, 210, 225, 208, 210,
 210, 208, 208, 224, 225, 213, 221, 236, 213, 210, 216, 231}
```

Out[28]=
```
балашовсавваарсеньевич
```

7. Подготовить два массива (Array) s и k длиной в 256 элементов и со смещением (origin) равным 0.

In[29]:=
```
Clear[s];
Clear[k];
arrayS = Array[s, 256, 0]
arrayK = Array[k, 256, 0]
```

Out[31]=
```
{s[0], s[1], s[2], s[3], s[4], s[5], s[6], s[7], s[8], s[9], s[10], s[11], s[12],
 s[13], s[14], s[15], s[16], s[17], s[18], s[19], s[20], s[21], s[22], s[23],
 s[24], s[25], s[26], s[27], s[28], s[29], s[30], s[31], s[32], s[33], s[34],
 s[35], s[36], s[37], s[38], s[39], s[40], s[41], s[42], s[43], s[44], s[45],
 s[46], s[47], s[48], s[49], s[50], s[51], s[52], s[53], s[54], s[55], s[56],
 s[57], s[58], s[59], s[60], s[61], s[62], s[63], s[64], s[65], s[66], s[67],
 s[68], s[69], s[70], s[71], s[72], s[73], s[74], s[75], s[76], s[77], s[78],
 s[79], s[80], s[81], s[82], s[83], s[84], s[85], s[86], s[87], s[88], s[89],
 s[90], s[91], s[92], s[93], s[94], s[95], s[96], s[97], s[98], s[99], s[100],
 s[101], s[102], s[103], s[104], s[105], s[106], s[107], s[108], s[109], s[110],
 s[111], s[112], s[113], s[114], s[115], s[116], s[117], s[118], s[119], s[120],
 s[121], s[122], s[123], s[124], s[125], s[126], s[127], s[128], s[129],
 s[130], s[131], s[132], s[133], s[134], s[135], s[136], s[137], s[138],
 s[139], s[140], s[141], s[142], s[143], s[144], s[145], s[146], s[147],
 s[148], s[149], s[150], s[151], s[152], s[153], s[154], s[155], s[156],
 s[157], s[158], s[159], s[160], s[161], s[162], s[163], s[164], s[165],
 s[166], s[167], s[168], s[169], s[170], s[171], s[172], s[173], s[174],
 s[175], s[176], s[177], s[178], s[179], s[180], s[181], s[182], s[183],
 s[184], s[185], s[186], s[187], s[188], s[189], s[190], s[191], s[192],
 s[193], s[194], s[195], s[196], s[197], s[198], s[199], s[200], s[201],
 s[202], s[203], s[204], s[205], s[206], s[207], s[208], s[209], s[210],
 s[211], s[212], s[213], s[214], s[215], s[216], s[217], s[218], s[219],
 s[220], s[221], s[222], s[223], s[224], s[225], s[226], s[227], s[228],
 s[229], s[230], s[231], s[232], s[233], s[234], s[235], s[236], s[237],
 s[238], s[239], s[240], s[241], s[242], s[243], s[244], s[245], s[246],
 s[247], s[248], s[249], s[250], s[251], s[252], s[253], s[254], s[255]}
```

Out[32]= {k[0], k[1], k[2], k[3], k[4], k[5], k[6], k[7], k[8], k[9], k[10], k[11], k[12],
k[13], k[14], k[15], k[16], k[17], k[18], k[19], k[20], k[21], k[22], k[23],
k[24], k[25], k[26], k[27], k[28], k[29], k[30], k[31], k[32], k[33], k[34],
k[35], k[36], k[37], k[38], k[39], k[40], k[41], k[42], k[43], k[44], k[45],
k[46], k[47], k[48], k[49], k[50], k[51], k[52], k[53], k[54], k[55], k[56],
k[57], k[58], k[59], k[60], k[61], k[62], k[63], k[64], k[65], k[66], k[67],
k[68], k[69], k[70], k[71], k[72], k[73], k[74], k[75], k[76], k[77], k[78],
k[79], k[80], k[81], k[82], k[83], k[84], k[85], k[86], k[87], k[88], k[89],
k[90], k[91], k[92], k[93], k[94], k[95], k[96], k[97], k[98], k[99], k[100],
k[101], k[102], k[103], k[104], k[105], k[106], k[107], k[108], k[109], k[110],
k[111], k[112], k[113], k[114], k[115], k[116], k[117], k[118], k[119], k[120],
k[121], k[122], k[123], k[124], k[125], k[126], k[127], k[128], k[129],
k[130], k[131], k[132], k[133], k[134], k[135], k[136], k[137], k[138],
k[139], k[140], k[141], k[142], k[143], k[144], k[145], k[146], k[147],
k[148], k[149], k[150], k[151], k[152], k[153], k[154], k[155], k[156],
k[157], k[158], k[159], k[160], k[161], k[162], k[163], k[164], k[165],
k[166], k[167], k[168], k[169], k[170], k[171], k[172], k[173], k[174],
k[175], k[176], k[177], k[178], k[179], k[180], k[181], k[182], k[183],
k[184], k[185], k[186], k[187], k[188], k[189], k[190], k[191], k[192],
k[193], k[194], k[195], k[196], k[197], k[198], k[199], k[200], k[201],
k[202], k[203], k[204], k[205], k[206], k[207], k[208], k[209], k[210],
k[211], k[212], k[213], k[214], k[215], k[216], k[217], k[218], k[219],
k[220], k[221], k[222], k[223], k[224], k[225], k[226], k[227], k[228],
k[229], k[230], k[231], k[232], k[233], k[234], k[235], k[236], k[237],
k[238], k[239], k[240], k[241], k[242], k[243], k[244], k[245], k[246],
k[247], k[248], k[249], k[250], k[251], k[252], k[253], k[254], k[255]}

8. Инициализировать массив s линейно (Range) целыми числами от 0 до 255.

In[33]:=
```
Do[s[i] = i, {i, 0, 255}]
arrayS
```

Out[34]= {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23,
24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83,
84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102,
103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118,
119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134,
135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150,
151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165,
166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180,
181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195,
196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225,
226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240,
241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255}

9. Установить генератор случайных чисел в начальное состояние с параметром N –

номером по списку в группе и инициализировать массив k случайными целыми числами из диапазона 0⟩255.

In[35]:=
```
SeedRandom[4]
Do[k[i] = RandomInteger[{0, 255}], {i, 0, Length[arrayK] - 1}]
arrayK
```

Out[35]=
```
RandomGeneratorState[ Method: ExtendedCA
                      State hash: 4128846954141538471 ]
```

Out[37]=
```
{79, 182, 88, 90, 45, 81, 89, 14, 90, 53, 242, 115, 255, 160, 141, 64, 176, 14,
 241, 66, 17, 170, 240, 198, 167, 233, 202, 168, 167, 231, 243, 160, 234, 120,
 117, 47, 138, 11, 92, 27, 86, 36, 82, 243, 3, 85, 201, 248, 130, 49, 164, 140,
 223, 60, 39, 75, 0, 208, 36, 181, 27, 1, 20, 34, 221, 202, 3, 82, 116, 23, 169,
 243, 112, 211, 209, 14, 23, 254, 229, 213, 12, 162, 88, 188, 98, 253, 144, 2,
 53, 3, 207, 170, 254, 128, 194, 35, 92, 197, 239, 63, 156, 92, 94, 103, 55,
 147, 125, 7, 196, 212, 69, 17, 211, 87, 141, 12, 167, 146, 246, 154, 208, 60,
 153, 21, 235, 246, 89, 111, 134, 203, 133, 81, 216, 178, 56, 74, 236, 21, 155,
 196, 102, 32, 24, 206, 239, 109, 32, 13, 92, 157, 116, 48, 140, 64, 169, 104,
 192, 29, 153, 12, 234, 82, 171, 220, 107, 190, 138, 40, 39, 202, 17, 234, 45,
 114, 96, 245, 205, 163, 225, 241, 65, 136, 168, 253, 32, 100, 152, 163, 241,
 200, 239, 1, 116, 156, 241, 252, 73, 215, 250, 41, 228, 96, 220, 113, 128,
 68, 10, 26, 186, 52, 169, 135, 161, 1, 212, 162, 240, 130, 219, 89, 148, 101,
 245, 206, 18, 250, 32, 109, 88, 67, 25, 233, 191, 27, 120, 202, 35, 150, 230,
 85, 193, 195, 158, 197, 223, 36, 77, 93, 218, 6, 228, 220, 83, 195, 207, 142}
```

10. Сформировать s – блок, выполнив следующие операции:
Установим значение индекса j равным 0.
Затем:
Для i от 0 до 255
j = (j + Si + Ki) mod 256
Поменяйте местами Si и Sj.

In[38]:=
```
j = 0;
Do[
 j = Mod[j + s[i] + k[i], 256];
 {s[i], s[j]} = {s[j], s[i]},
 {i, 0, 255}]
arrayS
```

Out[40]=
```
{247, 6, 96, 189, 165, 72, 158, 54, 21, 83, 64, 20, 227, 133, 219, 60, 206, 78, 159,
 162, 222, 62, 131, 2, 31, 102, 30, 200, 24, 56, 160, 75, 80, 73, 120, 172, 69,
 217, 51, 9, 234, 55, 16, 11, 252, 129, 117, 48, 8, 197, 103, 181, 127, 170,
 179, 109, 50, 183, 194, 97, 74, 71, 86, 77, 122, 94, 186, 155, 27, 34, 45, 147,
 228, 98, 92, 67, 248, 105, 42, 119, 132, 104, 195, 136, 236, 214, 118, 204,
 139, 212, 161, 110, 224, 3, 5, 95, 210, 76, 188, 235, 169, 138, 251, 1, 33,
 154, 0, 124, 178, 37, 19, 41, 232, 134, 13, 153, 114, 199, 151, 240, 89, 126,
 106, 99, 66, 211, 220, 57, 125, 87, 4, 244, 101, 90, 40, 201, 82, 49, 182, 150,
 243, 176, 208, 250, 255, 245, 163, 63, 123, 196, 38, 84, 237, 135, 180, 52,
 70, 43, 39, 81, 68, 164, 156, 146, 198, 115, 191, 167, 148, 53, 15, 184, 152,
 241, 85, 47, 175, 112, 22, 143, 100, 29, 144, 253, 213, 65, 177, 17, 226, 254,
 230, 233, 130, 223, 128, 140, 116, 145, 239, 107, 221, 231, 12, 205, 207, 7,
 215, 113, 193, 28, 202, 192, 185, 174, 14, 209, 121, 168, 218, 32, 166, 46,
 203, 93, 111, 137, 225, 216, 59, 44, 91, 173, 229, 171, 35, 246, 58, 61, 141,
 88, 149, 142, 36, 108, 157, 242, 23, 79, 26, 25, 187, 18, 190, 10, 238, 249}
```

11. Сформировать случайный байт, выполнив следующие операции:
В алгоритме применяются два счетчика i и j с нулевыми начальными значениями.
Чтобы сгенерировать случайный байт, выполните следующие операции:
i = (i + 1) mod 256;
j = (j + Si) mod 256;
Поменяйте местами Si и Sj;
t = (Si + Sj) mod 256;
K = St
Байт K используется в операции BitXor с открытым текстом для получения шифртекста или в операции BitXor с шифртекстом для получения открытого текста.

In[41]:=
```
i = 0;
j = 0;
i = Mod[i + 1, 256];
j = Mod[j + s[i], 256];
{s[i], s[j]} = {s[j], s[i]};
t = Mod[s[i] + s[j], 256];
kByte = s[t]
```

Out[47]=
```
198
```

12. Зашифровать, с применением операции BitXor[,] первый символ открытого текста.
Аналогичным образом расшифровать первый символ шифртекста.

In[48]:=
```
char = StringTake[plaintext, 1];
charNum = ToCharacterCode[char, "ISOLatinCyrillic"]
charNumCipher = BitXor[charNum, kByte]
charCipher = FromCharacterCode[charNumCipher, "ISOLatinCyrillic"]
```

Out[49]= {209}

Out[50]= {23}

Out[51]=

In[52]:=
```
charNum = BitXor[charNumCipher, kByte]
char = FromCharacterCode[charNum, "ISOLatinCyrillic"]
```

Out[52]= {209}

Out[53]= б

13. Определить длину открытого текста и провести поточное шифрование, получая для каждого символа открытого текста новый случайный байт шифрования (п. 11).

In[54]:=
```
plainTextLen = StringLength[plaintext]
Clear[s]; Clear[k];
arrayS = Array[s, 256, 0];
arrayK = Array[k, 256, 0];
SeedRandom[8]
Do[k[i] = RandomInteger[{0, 255}], {i, 0, Length[arrayK] - 1}]
Do[s[i] = i, {i, 0, 255}]
j = 0;
Do[
 j = Mod[j + s[i] + k[i], 256];
 {s[i], s[j]} = {s[j], s[i]},
 {i, 0, 255}]
keyPosl = {};
i = 0;
j = 0;
Do[
 i = Mod[i + 1, 256];
 j = Mod[j + s[i], 256];
 {s[i], s[j]} = {s[j], s[i]};
 t = Mod[s[i] + s[j], 256];
 AppendTo[keyPosl, s[t]],
 {i, 1, plainTextLen}]
keyPosl
```

Out[54]= 22

Out[58]= RandomGeneratorState[Method: ExtendedCA
State hash: 3 313 605 182 657 953 387]

Out[67]= {102, 100, 250, 203, 112, 253, 246, 64, 27,
 249, 0, 65, 22, 136, 236, 226, 95, 63, 178, 156, 157, 128}

In[68]:=
```
plainTextCode = ToCharacterCode[plaintext, "ISOLatinCyrillic"];
cipherTextCode =
 Table[BitXor[plainTextCode[[i]], keyPosl[[i]]], {i, 1, plainTextLen}]
```

Out[69]=
```
{183, 180, 33, 27, 152, 35, 36, 161, 203, 43,
 210, 145, 198, 104, 13, 55, 130, 211, 103, 78, 69, 103}
```

14. Преобразовать полученные коды в символы и сравнить с результатом п. 5.

In[70]:=
```
cipherText = FromCharacterCode[cipherTextCode, "ISOLatinCyrillic"]
```

Out[70]=
```
ЗД!  ⌗$ЁЫ+вЦh
7rgNEg
```

15. Расшифровать шифртекст, получив вновь исходную кодовую последовательность и преобразовав ее в символы.

In[71]:=
```
cipherTextCode = ToCharacterCode[cipherText, "ISOLatinCyrillic"];
plainTextCode =
  Table[BitXor[cipherTextCode[[i]], keyPosl[[i]]], {i, 1, Length[cipherTextCode]}];
plainText = FromCharacterCode[plainTextCode, "ISOLatinCyrillic"]
```

Out[73]=
```
балашовсавваарсеньевич
```