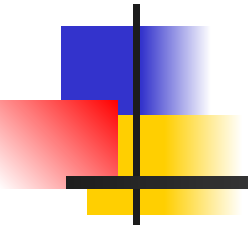
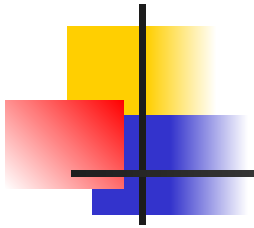


Шифры перестановки

Программирование алгоритмов



- **Шифр перестановки** использует изменение порядка следования символов.
- Криптограмма и исходный текст состоят из одних и тех же букв.

Шифр простой вертикальный перестановки

- Открытый текст пишется по горизонтали фиксированной ширины, а шифртекст считывается по вертикали.
- Для расшифрования такого текста достаточно написать шифртекст по вертикали той же самой ширины, и затем прочитать открытый текст по горизонтали.

Шифр простой вертикальной перестановки

- Фраза «*байты сохраняются в виде файлов*» размещается следующим образом

б а й т ы с о х р

а н я ю т с я в в

и д е ф а й л о в

- После считывания по вертикали получаем криптограмму: «*баи анд йяе тюф ыта ссй оял хво рвв*» (если в последнем блоке не хватает символов, добавляется буква х).

Программный код для алгоритма простой вертикальной перестановки

```
program prost_vert_perest;
var n,i,r,j,k:integer; s,s1,s2:string; a:array[1..5,1..5] of char;
begin
writeln ('vvedite shirinu bloka'); readln(n);
writeln('vvedite stroku'); readln(s);
writeln('vvedite regim: 1-shifrovanie, 2-rasshifrovanie');
readln(r); s1:='';
if r=1 then begin i:=1;
      while i<=length(s) do
          if s[i]=' '   then delete(s,i,1) else i:=i+1;
      if length(s) mod n <>0 then  for i:=1 to n-length(s)mod n do s:=s+'x';   {добавим в
      текст символы 'x', чтобы длина строки стала кратной ширине}
      i:=0;
      for k:=1 to length(s) div n  do
         for j:=1 to n do  begin i:=i+1;a[k,j]:=s[i]; end;
     for j:=1 to length(s) div n do
        for k:=1 to n do  s1:=s1+a[k,j];
              end
       else  begin i:=0;
            for j:=1 to length(s) div n do
                for k:=1 to n do begin i:=i+1; a[k,j]:=s[i];  end;
            for k:=1 to n do
               for j:=1 to length(s) div n do    s1:=s1+a[k,j];
                end;
  writeln (s1);
end.
```

Шифр двойной перестановки

- В таблицу записывается текст сообщения, а потом поочередно переставляются столбцы, затем строки.
- При расшифровке порядок перестановок обратный.

Шифр двойной перестановки

- Исходный текст «*байты сохраняются*».
- Ключом к шифру служат номера столбцов 2413 и номера строк 4123 исходной таблицы.
- Шифртекст «*оыхснряасюятйбта*».

	2	**4**	**1**	**3**			**1**	**2**	**3**	**4**			**1**	**2**	**3**	**4**
4	Б	А	Й	Т		**4**	Й	Б	Т	А		**1**	О	Ы	Х	С
1	Ы	С	О	Х		**1**	О	Ы	Х	С		**2**	Н	Р	Я	А
2	Р	А	Н	Я		**2**	Н	Р	Я	А		**3**	С	Ю	Я	Т
3	Ю	Т	С	Я		**3**	С	Ю	Я	Т		**4**	Й	Б	Т	А

Магические квадраты

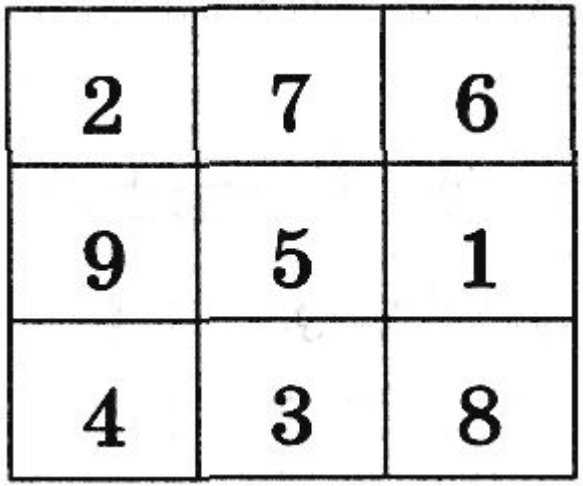
- Магический квадрат – квадратная таблица со вписанными в ее клетки последовательными натуральными числами (с 1), в которой сумма по всем строкам , столбцам и диагоналям одинакова.

Алгоритм шифрования

- Текст вписывается в таблицу в соответствии с приведенной в ней нумерацией, затем текст выписывается по строкам.
- Надежность шифра:

квадратов 3*3 – 1, 4*4 – 880, 5*5 – 250000.

- Для расшифровки записать текст в таблицу по строкам и по ключу получить открытый текст.



- Ключ:
- Текст: БИТЫ СОХРАНЯЮТСЯ В ВИДЕ ФАЙЛОВ
- Шифртекст:

ИХОАСБЫТРЯВЯИСНТЮВЕОЛ_ЙДАФВ

И	Х	О		Я	В	Я		Е	О	Л
А	С	Б		И	С	Н			Й	Д
Ы	Т	Р		Т	Ю	В		А	Ф	В

Программный код для алгоритма одиночной перестановки

```
program odin_perest;
var nmin,i,r,j,k,p:integer; s,s1,s2:string; a:array[1..5,1..5] of char;
begin
writeln('vvedite stroku'); readln(s);
writeln('vvedite kluch'); readln(s1);
writeln('vvedite regim: 1-shifrovanie, 2-rasshifrovanie');
readln(r);s2:='';
 if r=1 then begin i:=1;
    while i<=length(s) do if s[i]=' 'then delete(s,i,1) else i:=i+1;
    if length(s) mod length(s1) <>0 then
           for i:=1 to length(s1)-length(s)mod length(s1)  do s:=s+'x';
           i:=0;
           for k:=1 to length(s) div length(s1) do
                    for j:=1 to length(s1) do begin i:=i+1;a[k,j]:=s[i];end;
            for i:=1 to  length(s1) do begin
                      nmin:=1;
                      for j:=2 to length(s1) do if ord(s1[j])<ord(s1[nmin]) then nmin:=j;
                      s1[nmin]:=chr(ord('z')+10);
                     for k:=1 to length(s) div length(s1) do
                           s2:=s2+a[k,nmin];
                                        end;
                   end
         else  begin p:=0;
                  for i:=1 to  length(s1) do
                        begin nmin:=1;
                       for j:=2 to length(s1) do
                       if ord(s1[j])<ord(s1[nmin]) then nmin:=j;
                       s1[nmin]:=chr(ord('z')+10);
                       for k:=1 to length(s) div length(s1) do begin p:=p+1;
                      a[k,nmin]:=s[p];
                                                                    end;
                       end;
                   for k:=1 to length(s1) do for j:=1 to length(s) div length(s1) do
                             s2:=s2+a[k,j];
                   end;
         writeln (s2);
           end.
```