

## RLE

<code>void print_decoded()</code>	операция за декодиране.
<code>char const&amp; operator[](int);</code>	върща пореден символ
<code>rle operator+(rle) const;</code>	
<code>rle operator+=(rle);</code>	конкатениране на 2 списъка
<code>rle&amp; operator++();</code>	
<code>rle operator++(int);</code>	добавя още един символ
<code>rle&amp; operator--();</code>	
<code>rle operator--(int);</code>	премахва последния символ
<code>rle operator()(int, int);</code>	намира подсписък от дадено начало и дължина
<code>rle operator()(int, rle);</code>	вмъква подсписък в дадена позиция
<code>rle delete_f(int, int);</code>	изтрива подсписък по дадено начало и дължина
предефинирани оператори за сравнение	
<code>rle operator*();</code>	намира броя на символите в списъка
<code>friend bool findLetter(char, rle);</code>	проверява дали дадена буква я има в даден
списък	

## base64

<code>void addLetter(char);</code>	добаця буква в масива
<code>void encode();</code>	кодира масива
<code>void decode();</code>	декодира масива
<code>int findPos(char);</code>	намира мястото на буква в таблицата
<code>char operator[](int);</code>	намира буква по дадена позиция
<code>const base64 operator+(base64);</code>	
<code>base64 operator+=(base64);</code>	конкатениране на два масива
<code>base64 operator()(int, int);</code>	намиране на подмасив по дадено начало и
дължина	
<code>base64 operator()(int, base64);</code>	вмъква масив по дадено начало
<code>base64 delete_f(int, int);</code>	изтрива последователност от битове
<code>base64 cmp(int, int);</code>	намира подмасив (кодиран)