

# Основы программирования

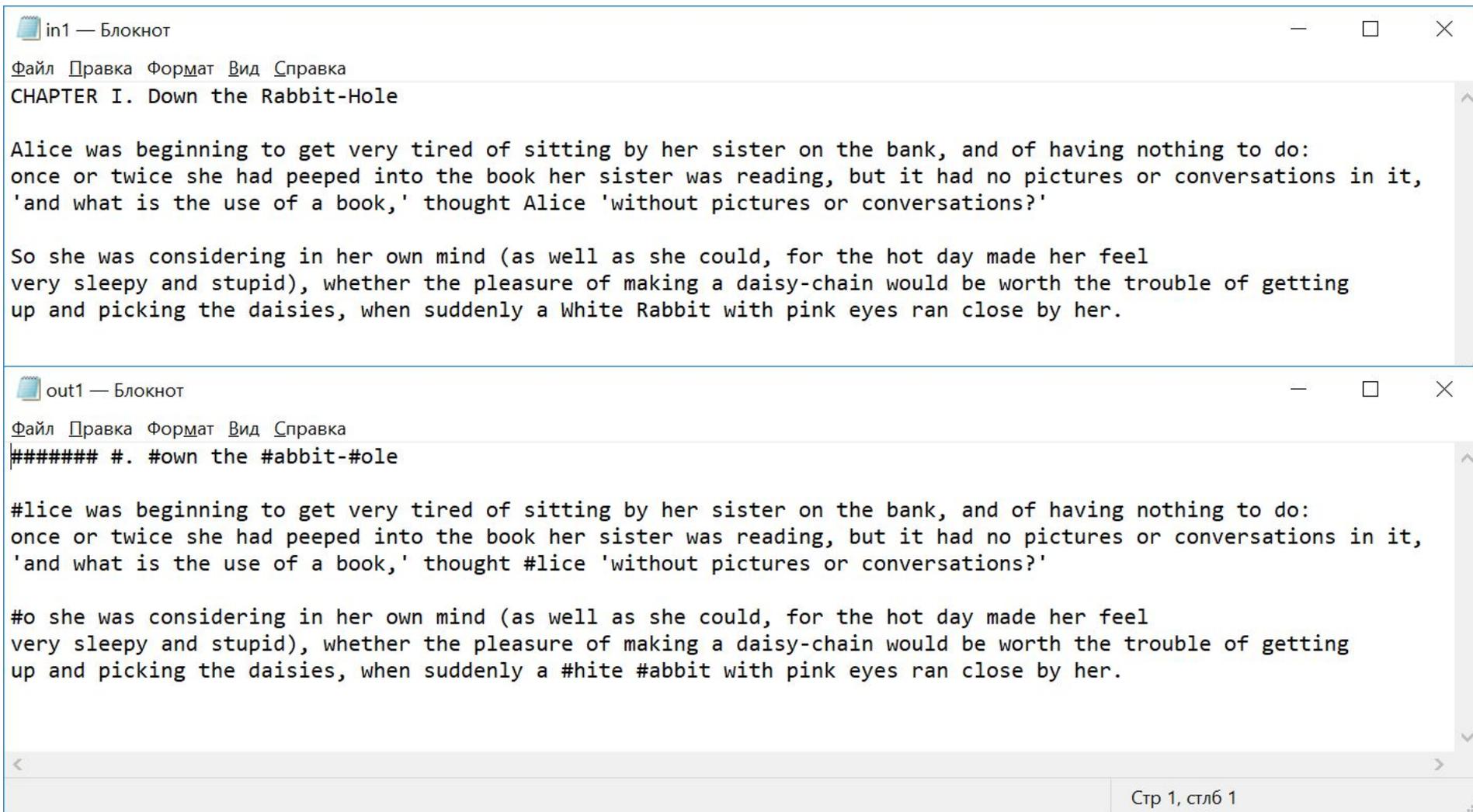
## Лабораторная работа №11

Работа с текстом.  
Многомодульные проекты.  
Простейшие способы шифрования текста.

Власенко О.Ф.

# Госимвольная обработка текстового файла

Переписать текст in1.txt в файл out1.txt заменив все большие латинские буквы на '#'.



The image shows two Notepad windows. The top window, titled 'in1 — Блокнот', contains the following text:

```
CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do:
once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it,
'and what is the use of a book,' thought Alice 'without pictures or conversations?'

So she was considering in her own mind (as well as she could, for the hot day made her feel
very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting
up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.
```

The bottom window, titled 'out1 — Блокнот', shows the result of the transformation where all uppercase letters are replaced by '#':

```
##### #. #own the #abbit-#ole

#lice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do:
once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it,
'and what is the use of a book,' thought #lice 'without pictures or conversations?'

#o she was considering in her own mind (as well as she could, for the hot day made her feel
very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting
up and picking the daisies, when suddenly a #hite #abbit with pink eyes ran close by her.
```

At the bottom right of the second window, the status bar indicates 'Стр 1, стлб 1'.

# Символьная обработка текстового файла (2)

Переписать текст in1.txt в файл out1.txt заменив все большие латинские буквы на '#'.  
#

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

void main() {

    char filename1[] = "d:\\temp\\in1.txt";
    char filename2[] = "d:\\temp\\out1.txt";

    FILE * fin = fopen(filename1, "rt");
    FILE * fout = fopen(filename2, "wt");
    char ch;

    while ((ch = getc(fin)) != EOF)
    {
        char ch_new = ch;

        if (ch >= 'A' && ch <= 'Z')
            ch_new = '#';

        fprintf(fout, "%c", ch_new);
        //printf(" <%c / %c>", ch_new, ch);
        printf("%c", ch_new);
    }
    fclose(fin);
    fclose(fout);

    {
        int x;
        scanf("%d", &x);
    }
}
```

# Задача 1. Сделать модуль BigLettersGo

Нужно создать модуль, в который поместить функцию преобразования символа согласно предыдущего задания – все большие латинские буквы превратить в #, остальные символы оставить неизменными.

# Задача 1. Сделать модуль BigLettersGo.

Шаг 1. Создать отдельную функцию для обработки одного символа. И вызвать её из main()

## Шаг 1

```
#define _CRT_SECURE_NO_WARNINGS
```

```
#include <stdio.h>
```

```
char bigLettersGo(char c) {
```

```
    char ch_new = c;
```

```
    if (c >= 'A' && c <= 'Z')  
        ch_new = '#';
```

```
    return ch_new;
```

```
}
```

```
void main() {
```

```
    char filename1[] = "d:\\temp\\in1.txt";
```

```
    char filename2[] = "d:\\temp\\out1.txt";
```

```
    FILE * fin = fopen(filename1, "rt");
```

```
    FILE * fout = fopen(filename2, "wt");
```

```
    char ch;
```

```
    while ((ch = getc(fin)) != EOF)
```

```
    {
```

```
        char ch_new = bigLettersGo(ch);
```

# Задача 1. Сделать модуль BigLettersGo.

Шаг 2. Создать прототип функции для обработки одного символа.

## Шаг 2

```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

char bigLettersGo(char c);

char bigLettersGo(char c) {

    char ch_new = c;

    if (c >= 'A' && c <= 'Z')
        ch_new = '#';

    return ch_new;
}

void main() {

    char filename1[] = "d:\\temp\\in1.txt";
    char filename2[] = "d:\\temp\\out1.txt";

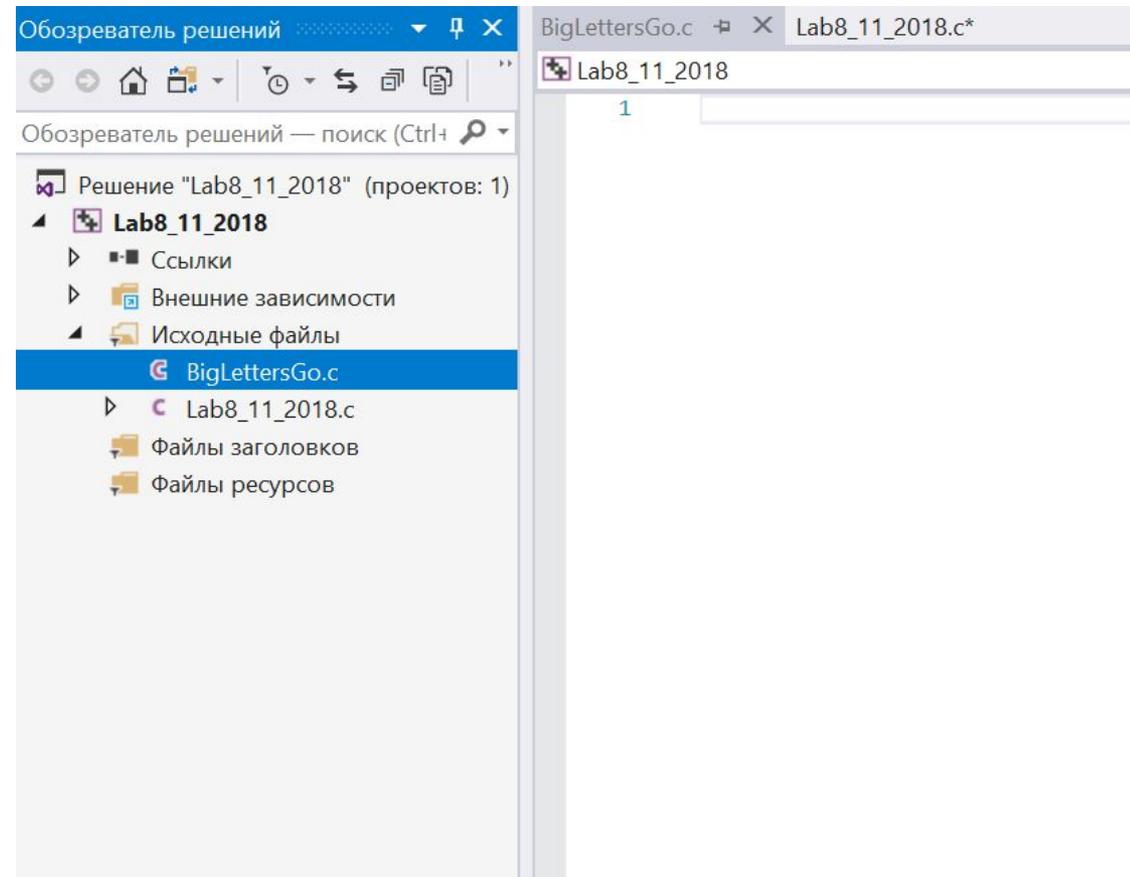
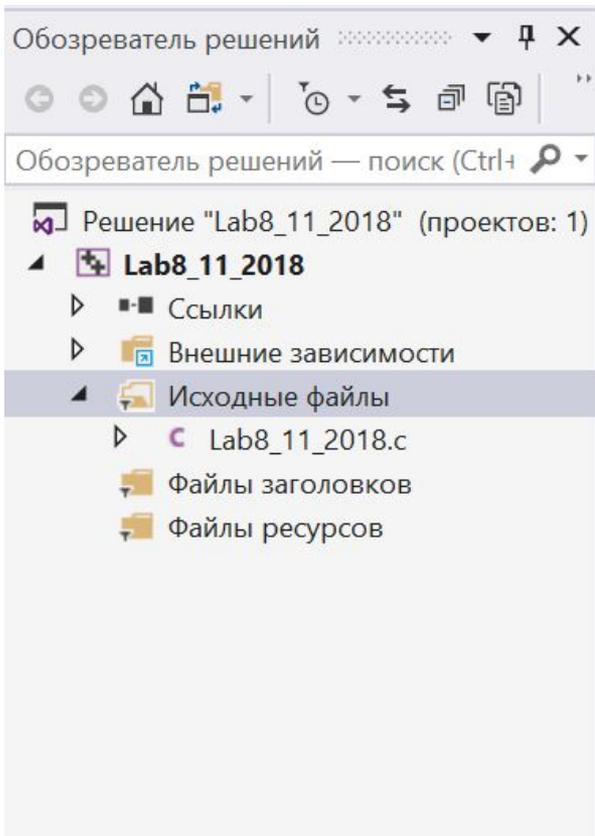
    FILE * fin = fopen(filename1, "rt");
    FILE * fout = fopen(filename2, "wt");
    char ch;

    while ((ch = getc(fin)) != EOF)
    {
        char ch_new = bigLettersGo(ch);
```

# Задача 1. Сделать модуль BigLettersGo.

## Шаг 3

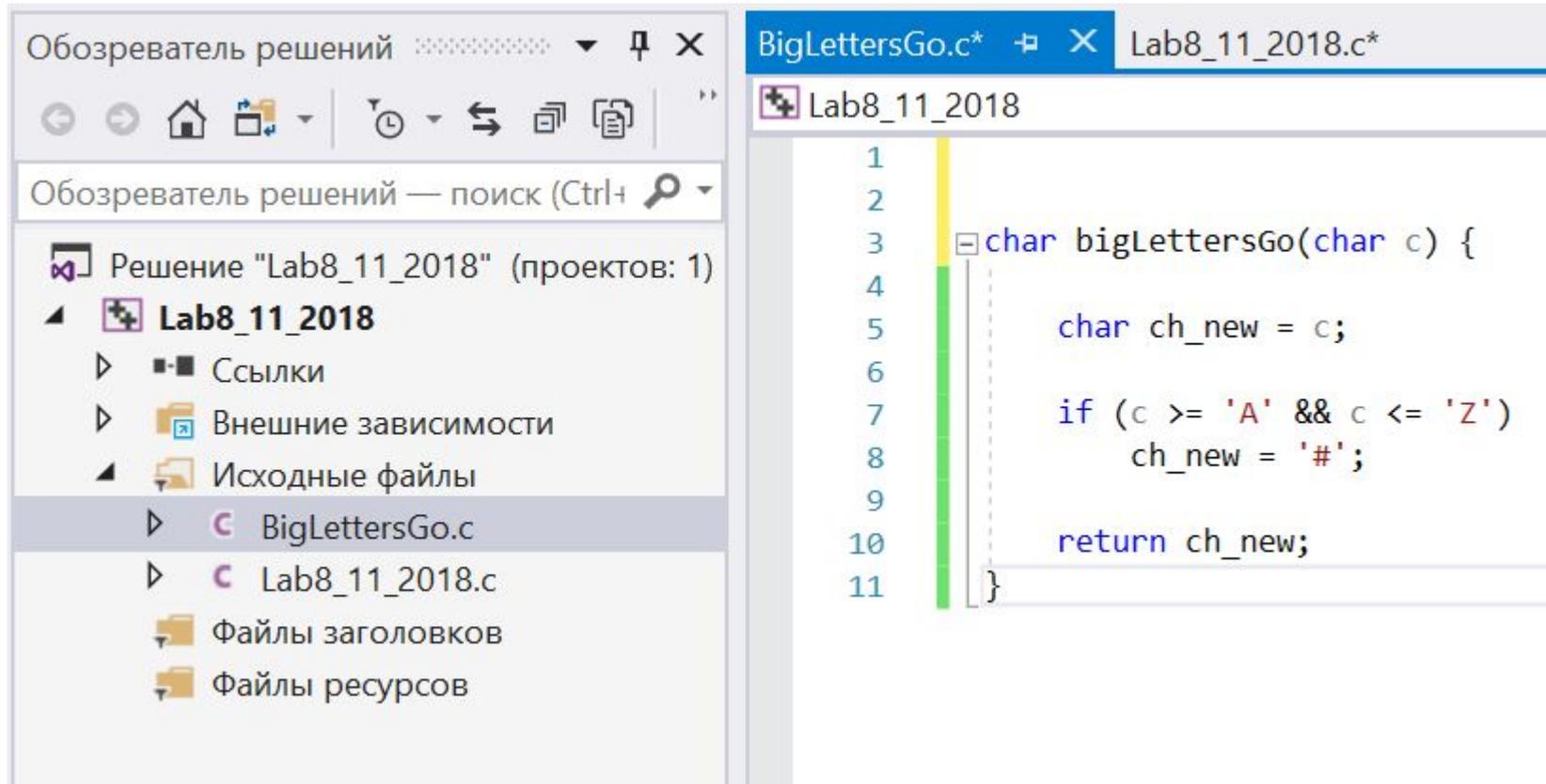
Шаг 3. Создать еще один файл с исходным кодом.



# Задача 1. Сделать модуль BigLettersGo.

## Шаг 4

Шаг 4. Перенести определение функции в новый файл исходного кода.



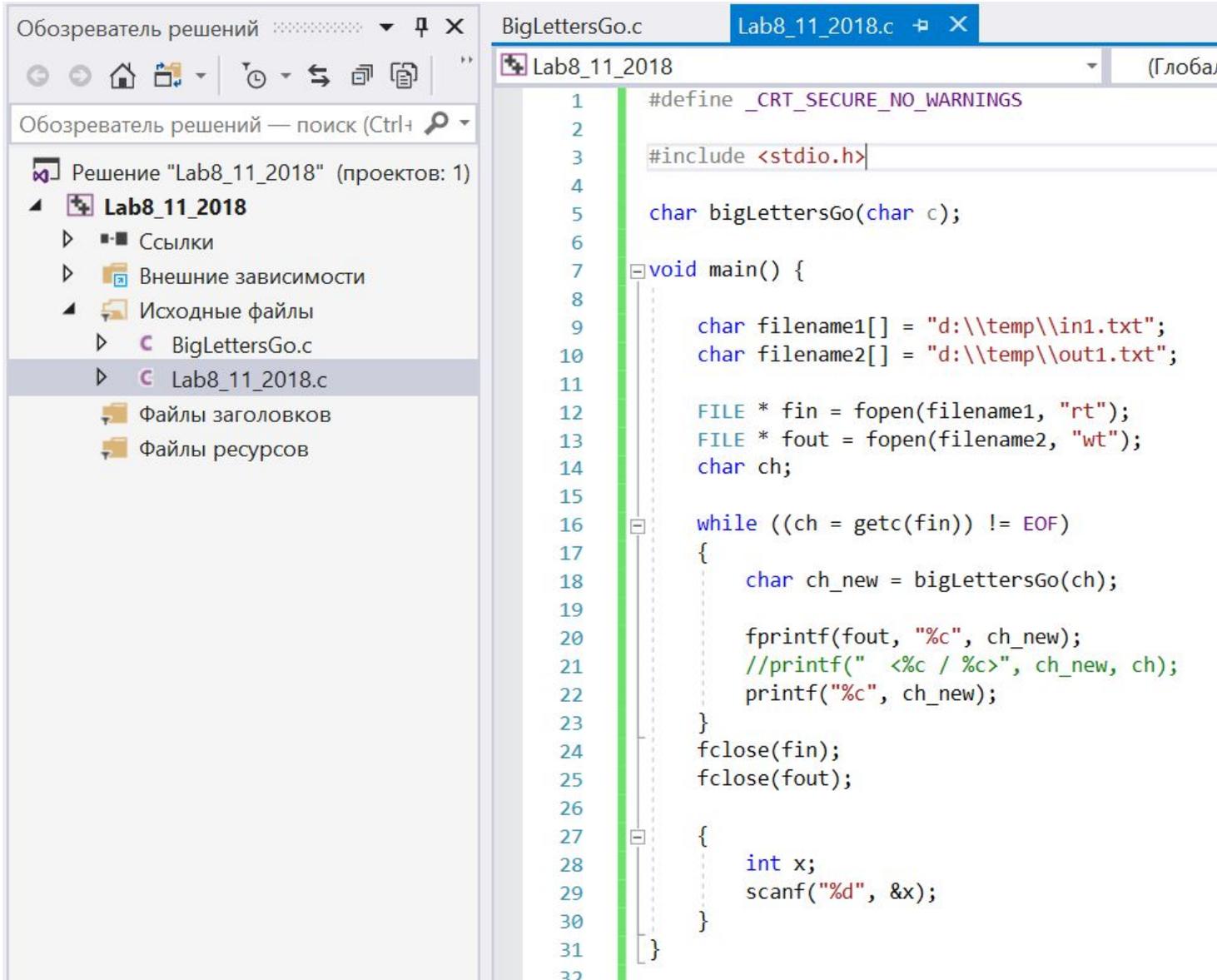
The screenshot displays the Visual Studio IDE. On the left, the Solution Explorer shows the project structure for 'Lab8\_11\_2018', including 'BigLettersGo.c' and 'Lab8\_11\_2018.c'. The main editor window shows the code for 'Lab8\_11\_2018.c' with the following function definition:

```
1
2
3 char bigLettersGo(char c) {
4
5     char ch_new = c;
6
7     if (c >= 'A' && c <= 'Z')
8         ch_new = '#';
9
10    return ch_new;
11 }
```

# Задача 1. Сделать модуль BigLettersGo.

## Шаг 4

Шаг 4. Перенести определение функции в новый файл исходного



```
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  char bigLettersGo(char c);
6
7  void main() {
8
9      char filename1[] = "d:\\temp\\in1.txt";
10     char filename2[] = "d:\\temp\\out1.txt";
11
12     FILE * fin = fopen(filename1, "rt");
13     FILE * fout = fopen(filename2, "wt");
14     char ch;
15
16     while ((ch = getc(fin)) != EOF)
17     {
18         char ch_new = bigLettersGo(ch);
19
20         fprintf(fout, "%c", ch_new);
21         //printf(" <%c / %c>", ch_new, ch);
22         printf("%c", ch_new);
23     }
24     fclose(fin);
25     fclose(fout);
26
27     {
28         int x;
29         scanf("%d", &x);
30     }
31 }
32
```

# Задача 1. Сделать модуль BigLettersGo.

## Шаг 5

Шаг 5. Создать новый заголовочный файл.

The screenshot shows an IDE interface with a solution browser on the left and a code editor on the right. The solution browser displays the project structure for "Lab8\_11\_2018", including source files and header files. The code editor shows the content of the newly created header file "BigLettersGo.h", which contains a single line of code: `#pragma once`.

Обозреватель решений

BigLettersGo.h BigLettersGo.c Lab8\_11\_2018.c

Lab8\_11\_2018 (Глобаль

```
1 #pragma once
2
```

Обозреватель решений — поиск (Ctrl+)

Решение "Lab8\_11\_2018" (проектов: 1)

- Lab8\_11\_2018
  - Ссылки
  - Внешние зависимости
  - Исходные файлы
    - BigLettersGo.c
    - Lab8\_11\_2018.c
  - Файлы заголовков
    - BigLettersGo.h
  - Файлы ресурсов

# Задача 1. Сделать модуль BigLettersGo.

## Шаг 6

Шаг 6. Перенести объявление функции (прототип) в новый заголовочный файл.

The screenshot displays the Visual Studio IDE. On the left, the Solution Explorer shows a project named 'Lab8\_11\_2018' with a tree view containing 'Ссылки', 'Внешние зависимости', 'Исходные файлы' (with sub-items 'BigLettersGo.c' and 'Lab8\_11\_2018.c'), 'Файлы заголовков' (with 'BigLettersGo.h' selected), and 'Файлы ресурсов'. The main editor window has three tabs: 'BigLettersGo.h', 'BigLettersGo.c', and 'Lab8\_11\_2018.c\*'. The active tab is 'Lab8\_11\_2018.c', showing the following code:

```
1 #pragma once
2
3 char bigLettersGo(char c);
```

# Задача 1. Сделать модуль BigLettersGo.

## Шаг 7

Шаг 7. Подключить новый заголовочный файл и файлам с исходным кодом – к новому файлу и к главному файлу.

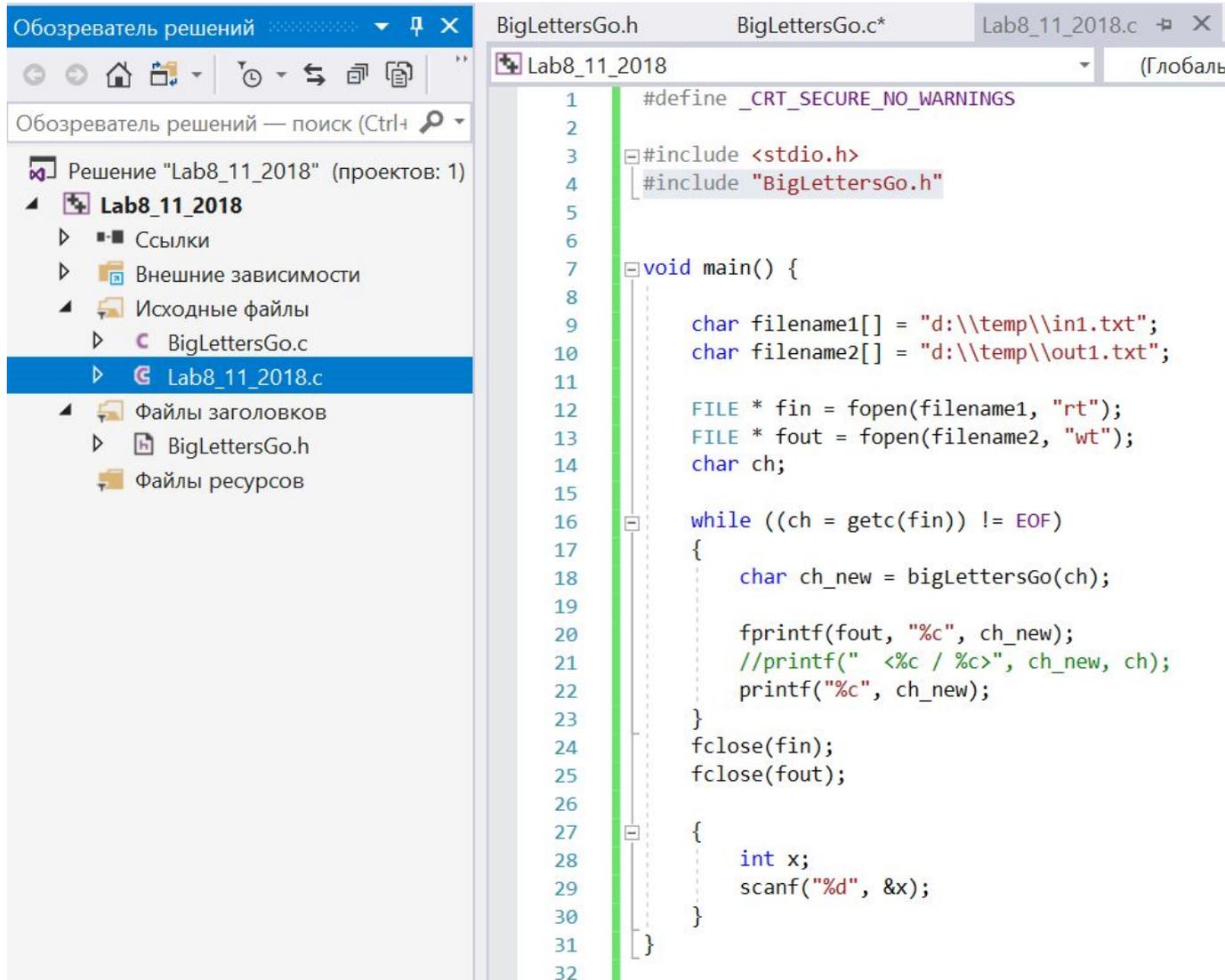
The screenshot shows the Visual Studio IDE. On the left is the 'Обозреватель решений' (Solution Explorer) window, displaying the project structure for 'Lab8\_11\_2018'. The 'Исходные файлы' (Source Files) folder is expanded, showing 'BigLettersGo.c' selected. On the right is the 'BigLettersGo.c\*' code editor window, showing the following C++ code:

```
1 #include "BigLettersGo.h"
2
3 char bigLettersGo(char c) {
4
5     char ch_new = c;
6
7     if (c >= 'A' && c <= 'Z')
8         ch_new = '#';
9
10    return ch_new;
11 }
12
13
```

# Задача 1. Сделать модуль BigLettersGo.

## Шаг 7

Шаг 7. Подключить новый заголовочный файл и файлам с исходным кодом – к новому файлу и к главному файлу.



The screenshot shows the Visual Studio IDE. On the left is the Solution Explorer with the project 'Lab8\_11\_2018' expanded to show files 'BigLettersGo.c' and 'Lab8\_11\_2018.c'. The main editor window displays the code for 'Lab8\_11\_2018.c'. The code includes a preprocessor directive to suppress warnings, includes 'stdio.h' and 'BigLettersGo.h', and contains a 'main' function that reads from 'in1.txt' and writes to 'out1.txt' using the 'bigLettersGo' function.

```
1 #define _CRT_SECURE_NO_WARNINGS
2
3 #include <stdio.h>
4 #include "BigLettersGo.h"
5
6
7 void main() {
8
9     char filename1[] = "d:\\temp\\in1.txt";
10    char filename2[] = "d:\\temp\\out1.txt";
11
12    FILE * fin = fopen(filename1, "rt");
13    FILE * fout = fopen(filename2, "wt");
14    char ch;
15
16    while ((ch = getc(fin)) != EOF)
17    {
18        char ch_new = bigLettersGo(ch);
19
20        fprintf(fout, "%c", ch_new);
21        //printf(" <%c / %c>", ch_new, ch);
22        printf("%c", ch_new);
23    }
24    fclose(fin);
25    fclose(fout);
26
27    {
28        int x;
29        scanf("%d", &x);
30    }
31 }
32
```

# Задача 1. Сделать модуль BigLettersGo.

## ИТОГО

Поздравляем Вас!

Вы только что создали многомодульный проект!

# ROT13 – реализация шифрования

На основе шифрования Цезаря создадим шифрование ROT13

```
in1 — Блокнот
Файл Правка Формат Вид Справка
CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do:
once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it,
'and what is the use of a book,' thought Alice 'without pictures or conversations?'
```

```
out2 — Блокнот
Файл Правка Формат Вид Справка
PUNCGRE V. Qbjg gur Enoovg-Ubyr

Nyvpr jnf ortvaavat gb trg irel gverq bs fvggvat ol ure fvfgrg ba gur onax, naq bs univat abguvat gb qb:
bapr be gjvpr fur unq crrcrq vagb gur obbx ure fvfgrg jnf ernqvaf, ohg vg unq ab cvpgherf be pbairfngvbaf va vg,
'naq jung vf gur hfr bs n obbx,' gubhtug Nyvpr 'jvgubhg cvpgherf be pbairfngvbaf?'
```

```
out2_2 — Блокнот
Файл Правка Формат Вид Справка
CHAPTER I. Down the Rabbit-Hole

Alice was beginning to get very tired of sitting by her sister on the bank, and of having nothing to do:
once or twice she had peeped into the book her sister was reading, but it had no pictures or conversations in it,
'and what is the use of a book,' thought Alice 'without pictures or conversations?'

So she was considering in her own mind (as well as she could, for the hot day made her feel
very sleepy and stupid), whether the pleasure of making a daisy-chain would be worth the trouble of getting
up and picking the daisies, when suddenly a White Rabbit with pink eyes ran close by her.
```

Стр 1, стлб 1

# ROT13 – реализация шифрования

На основе шифрования  
Цезаря создадим  
шифрование ROT13

```
BigLettersGo.h  BigLettersGo.c  Lab8_11_2018.c  [X]
Lab8_11_2018  (Глобаль)
1  #define _CRT_SECURE_NO_WARNINGS
2
3  #include <stdio.h>
4
5  int encodeCaesar(int ch, int key);
6  int encoderROT13(int ch);
7
8  void main() {
9      char filename1[] = "d:\\temp\\in1.txt";
10     char filename2[] = "d:\\temp\\out2.txt";
11     char filename3[] = "d:\\temp\\out2_2.txt";
12     FILE * fin = fopen(filename1, "rt");
13     FILE * fout = fopen(filename2, "wt");
14     char ch;
15
16     while ((ch = getc(fin)) != EOF)
17     {
18         char ch_new = encoderROT13(ch);
19
20         fprintf(fout, "%c", ch_new);
21         //printf(" <%c / %c>", ch_new, ch);
22         printf("%c", ch_new);
23     }
24     fclose(fin);
25     fclose(fout);
26
27     {
28         int x;
29         scanf("%d", &x);
30     }
31 }
```

# ROT13 – реализация шифрования

На основе шифрования  
Цезаря создадим  
шифрование ROT13

```
int encodeCaesar(int ch, int key) {
    // char smallLetters[] = "abcdefghijklmnopqrstuvwxyz";
    /// char bigLetters[] = "ABCDEFGHIJKLMNPOQRSTUVWXYZ";

    int newCh = ch;

    if (ch >= 'A' && ch <= 'Z') {
        newCh = ch + key;
        if (newCh > 'Z')
            newCh = 'A' + (newCh - 'Z') - 1;
    }

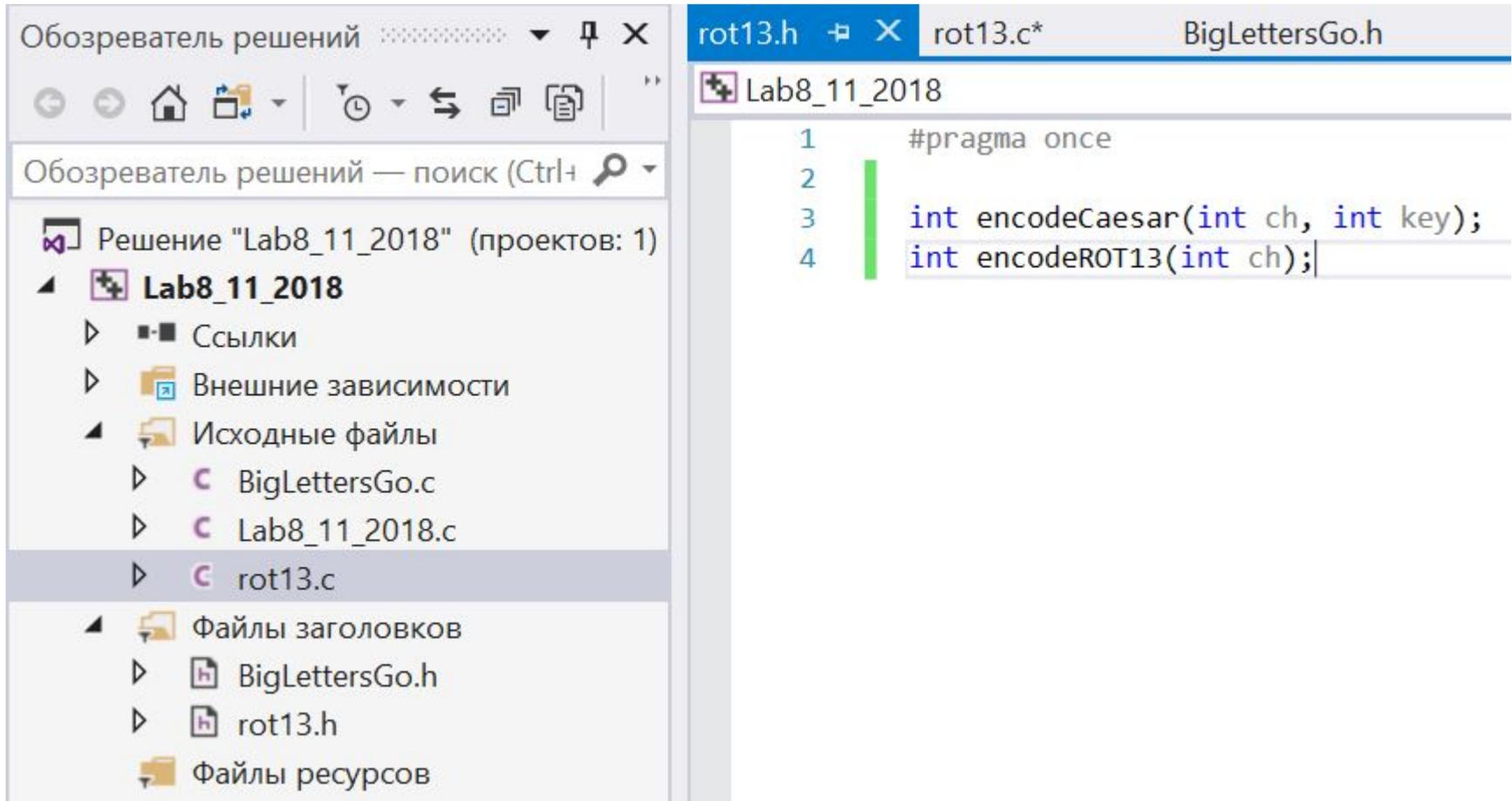
    if (ch >= 'a' && ch <= 'z') {
        newCh = ch + key;
        if (newCh > 'z')
            newCh = 'a' + (newCh - 'z') - 1;
    }

    return newCh;
}

int encodeROT13(int ch) {
    return encodeCaesar(ch, 13);
}
```

# Задача 2. Создать модуль ROT13

Сделать модуль с функциями шифрования методами Цезаря и ROT13.



The image shows a screenshot of an IDE interface. On the left is the 'Обозреватель решений' (Solution Explorer) pane, which displays a project named 'Lab8\_11\_2018'. Under this project, there are several folders: 'Ссылки' (Links), 'Внешние зависимости' (External Dependencies), 'Исходные файлы' (Source Files), 'Файлы заголовков' (Header Files), and 'Файлы ресурсов' (Resource Files). The 'Исходные файлы' folder is expanded, showing three files: 'BigLettersGo.c', 'Lab8\_11\_2018.c', and 'rot13.c'. The 'rot13.c' file is selected and highlighted. The 'Файлы заголовков' folder is also expanded, showing 'BigLettersGo.h' and 'rot13.h'. On the right is the code editor, which has three tabs: 'rot13.h', 'rot13.c\*', and 'BigLettersGo.h'. The 'rot13.c\*' tab is active, showing the following code:

```
1 #pragma once
2
3 int encodeCaesar(int ch, int key);
4 int encodeROT13(int ch);
```

# Задача 3. Создать модуль

## EncodeTransposition

Сделать модуль с функциями шифрования/дешифрования методом блочной перестановки.

```
void main() {  
    char filename1[] = "d:\\temp\\in1.txt";  
    char filename2[] = "d:\\temp\\out3.txt";  
    char filename3[] = "d:\\temp\\out3_2.txt";  
  
    EncodeTranspositionFile(filename1, filename2);  
    DecodeTranspositionFile(filename2, filename3);  
  
    {  
        int x;  
        scanf("%d", &x);  
    }  
}
```

# Задача 3. Создать модуль EncodeTransposition (2)

Реализация шифрования на уровне файла

```
void EncodeTranspositionFile(char * filenameIn, char * filenameOut) {
    FILE * fin = fopen(filenameIn, "rt");
    FILE * fout = fopen(filenameOut, "wt");

    char s[9]; // Входная строка - s[9] ==> 8 символов и завершающий '\0'
    char s_out[9]; // Зашифрованная строка

    while (!feof(fin)) {
        read8chars(fin, s);

        // в s ровно 8 символов - шифруем перестановкой
        EncodeTransposition(s, s_out);

        fprintf(fout, "%s", s_out);
        printf(" {%s --> %s}", s, s_out);
    }

    fclose(fin);
    fclose(fout);

    printf("\n===== \n");
}
```

# Задача 3. Создать модуль EncodeTransposition (3)

Реализация шифрования на уровне одного блока

```
// ключ шифрования - индексы, куда ставить очередной символ
int key[8] = { 3, 1, 5, 2, 6 ,7 ,4, 0 };

// шифрует один блок текста (8 символов) используя ключ key
void EncodeTransposition(char * s1, char * s2) {
    int i;
    for (i = 0; i < 8; i++) {
        s2[key[i]] = s1[i];
    }
    s2[8] = '\0';
}
```

# Задача 3. Создать модуль

## EncodeTransposition (4)

Чтение из файла одного блока текста (8 символов)

```
// прочитать из файла 8 символов
// если символов меньше чем 8, то дополнить пробелами
void read8chars(FILE * fin, char * s) {

    int i = 0;

    do {

        char ch = getc(fin);
        if (ch == EOF)
            break;
        s[i] = ch;
        i++;
    } while (i < 8);

    while (i < 8) {
        s[i] = ' ';
        i++;
    }

    s[8] = '\0';
}
```

# Задача 3. Создать модуль EncodeTransposition (5)

Шифрование одно блока и шифрование файла нужно реализовать самостоятельно.

```
void DecodeTransposition(char * s1, char * s2) {
    int i;
    for (i = 0; i < 8; i++) {

void DecodeTranspositionFile(char * filenameIn, char * filenameOut) {

    FILE * fin = fopen(filenameIn, "rt");
    FILE * fout = fopen(filenameOut, "wt");

    char s[9]; // Входная строка - s[9] ==> 8 символов и завершающий '\0'
    char s_out[9]; // Зашифрованная строка

    while (!feof(fin)) {
        read8chars(fin, s);
```

# Задача 4\*\*. Ключ задается с клавиатуры

Нужно предоставить пользователю возможность указать файл и задать ключ шифровать - через клавиатуру. Ключ должен быть вычислен на основе ключевой фразы.

# Задача 4. Ключ задается с клавиатуры (2)

Генерация ключа для перестановки на основе ключевой строки.

Нужно выполнить трассировку функции (ручную), чтобы понять как она работает.

```
void main() {
    char filename1[] = "d:\\temp\\in1.txt";
    char filename2[] = "d:\\temp\\out3.txt";
    char filename3[] = "d:\\temp\\out3_2.txt";

    KeyGeneration("I have an Apple");

    printf("\n");
    for (int i = 0; i < 8; i++)
        printf("%d, ", key[i]);
    printf("]\n");

    EncodeTranspositionFile(filename1, filename2);
    DecodeTranspositionFile(filename2, filename3);

    {
        int x;
        scanf("%d", &x);
    }
}
```

```
// Генерация ключа по ключевой фразе
// Ключевая фраза не короче 8 символов!
void KeyGeneration(char * phrase) {
    int i, j;
    for (i = 0; i < 8; i++) {
        int index = 0;
        for (j = 0; j < 8; j++)
            if (phrase[j] < phrase[i])
                index++;

        for (j = 0; j < i; j++)
            if (phrase[j] == phrase[i])
                index++;

        key[i] = index;
    }
}
```

# Задача 4. Ключ задается с клавиатуры (3)

Ввод имен файлов с клавиатуры:

```
void main() {  
    char filename1[100] = "d:\\temp\\in1.txt";  
    char filename2[100] = "d:\\temp\\out3.txt";  
    char filename3[100] = "d:\\temp\\out3_2.txt";  
    char keyPhrase[100];
```

```
    printf("Key phrase:");  
    fgets(keyPhrase, sizeof(keyPhrase), stdin);
```

```
    correctFileName(filename1);  
    correctFileName(filename2);  
    correctFileName(filename3);
```

```
    EncodeTranspositionFile(filename1, filename2);  
    DecodeTranspositionFile(filename2, filename3);
```

```
void correctFileName(char *filename) {  
  
    int len = strlen(filename);  
  
    if (filename[len - 1] == '\\n')  
        filename[len - 1] = '\\0';  
  
}
```

# Задача 5. Индивидуальная задача

1. Все маленькие латинские буквы заменить символом '#'.
2. Все гласные латинские буквы заменить символом '&'.
3. Все согласные латинские буквы заменить символом '&'.
4. Все гласные большие латинские буквы заменить символом '\$'.
5. Все гласные маленькие латинские буквы заменить символом '@'.
6. Все цифры заменить символом 'X'.
7. Все латинские буквы сделать большими.
8. Все латинские буквы сделать маленькими.
9. Все символы за исключением латинских букв, заменить символом '\_' (подчеркивание).
10. Все знаки препинания заменить символом '\_' (подчеркивание).
11. Все символы арифметических операций, заменить символом '\_' (подчеркивание).

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	DEL

Таблица ASCII

# Домашнее задание

Домашняя работа по лабораторной работе №11 включает в себя

**НИЧЕГО** если **ВСЕ** обязательные задачи были сделаны на уроке.

Если не все задачи были сделаны на уроке, то необходимо все доделать, и оформить стандартный отчет – включая распечатки и блоксхемы.