Лабораторная работа №3 Дешифровка текста из изображения

[c++](https://neroid.ru/category/coding/cpp/) / От [admin](https://neroid.ru/author/admin/" \o "Смотреть все записи от admin)

**Цель:**

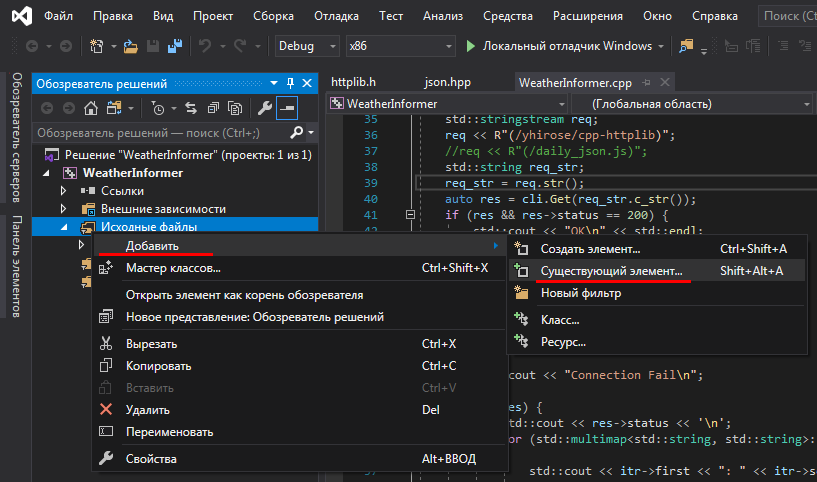
1. Закрепить навыки разработки программ использующих операторы цикла;
2. Закрепить навыки разработки программ использующих массивы;
3. Освоить методы подключения сторонних библиотек.

Постановка задачи

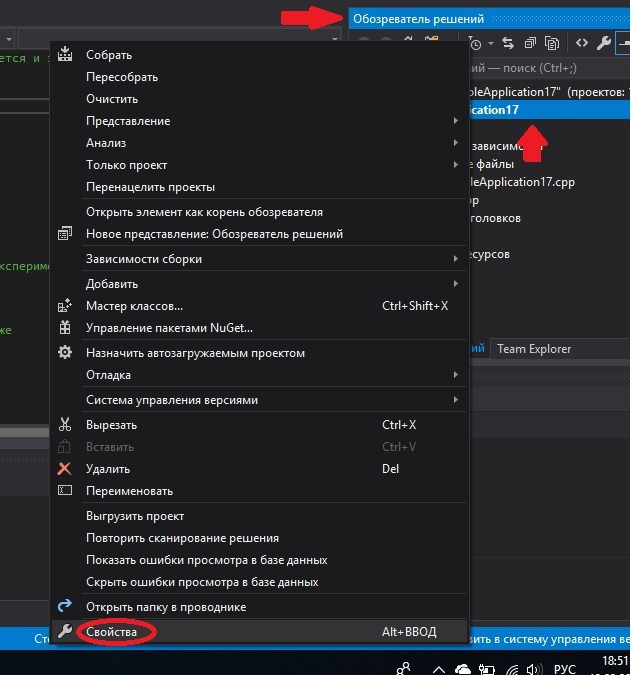
Напишите на языке С++ программу, которая по заданному изображению и ключу восстанавливает исходное текстовое сообщение (Си-строка) зашифрованное в изображении.

Ход работы

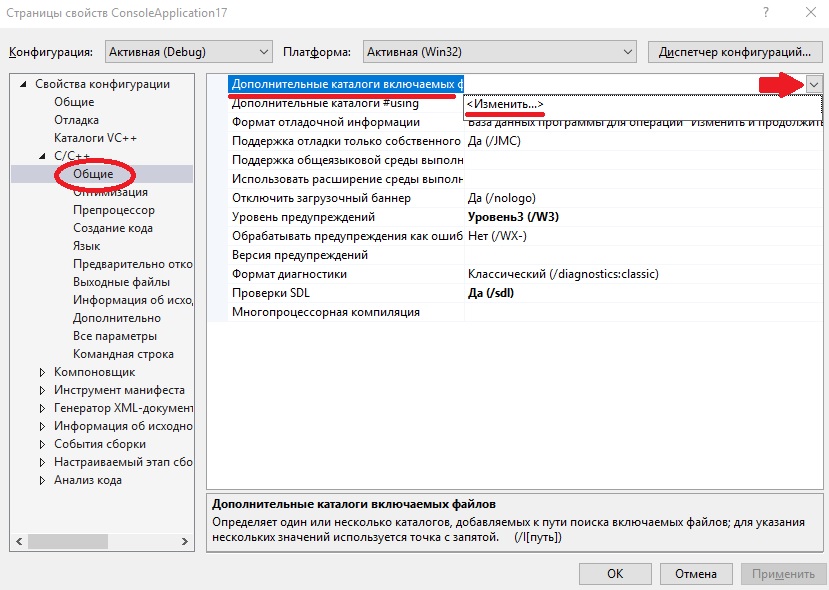
1. Для работы с изображениями в формате **bmp** скачайте библиотеку <https://github.com/marc-q/libbmp>.
2. В скаченном репозитории найдите папку **CPP**, в ней расположены файлы которые нужно подключить к своему проекту, а так же пример использования библиотеки.
3. Файл **libbmp.cpp** скопируйте в папку проекта и добавьте к проекту, как показано на Рис.1

  
*Рис. 1. Добавление \*.сpp файла к проекту*

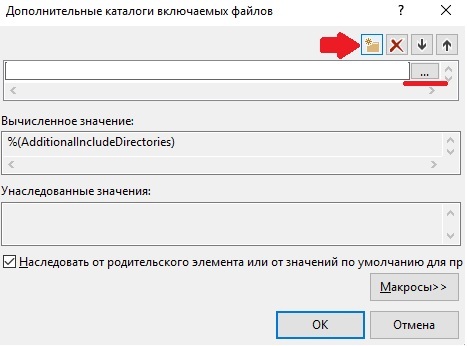
1. В папке с проектом создайте папку **include** и скопируйте туда **libbmp.h**
2. Переходим в *«Обозреватель решений» >* кликаем правой кнопкой мыши (ПКМ) по названию проекта *> «Свойства»* (Рис.2).

  
*Рис. 2. Как открыть свойства проекта*

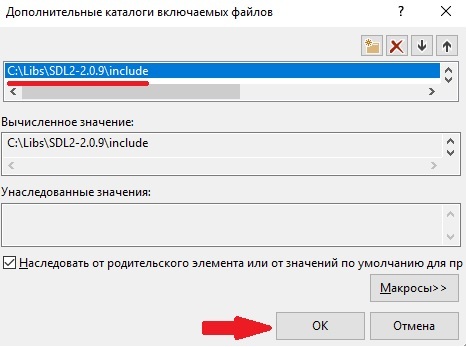
1. В *«Свойствах конфигурации»* открываем вкладку *«С/С++» > «Общие» > «Дополнительные каталоги включаемых файлов» >* нажимаем на стрелочку в конце *> «Изменить»*:

  
*Рис. 3. Свойства проекта*

1. В появившемся окне кликаем на иконку с изображением папки, а затем на появившееся троеточие:

  
*Рис. 4. Дополнительные каталоги включаемых файлов*

1. Заголовочные файлы находятся в папке **include** внутри нашего проекта, поэтому переходим в неё и нажимаем «Выбор папки», а затем «Ок». В примере фалы находятся в папке (C:\Libs\SDL2-2.0.9\include)

  
*Рис. 5. Путь к каталогу включаемых файлов*

1. Теперь можно подключить библиотеку в свой проект. Для этого добавьте в главный файл вашей программы строку (**после iostream**):



|  |  |
| --- | --- |
| 1 | #include "libbmp.h" |

1. В соответствии с вашим вариантом выберите из таблице 1 ссылку на картинку и ключ. Номер варианта соответствует номеру в журнале академической группы. **Картинка весит около 1 МБ**.
2. Декодируйте сообщение и выведите его в консоль.
3. В отчёт необходимо включить:
   * Исходное изображение и ключ;
   * Код программы для декодирования сообщения;
   * Декодированное сообщение.

**Таблица 1**

| **№** | **Ссылка на изображение** | **Ключ** |
| --- | --- | --- |
| 1 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic1.bmp) | 11r 11g 11b 01r 01g 01b 10r 10g |
| 2 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic2.bmp) | 00r 00g 00b 10r 10g 10b 01r 01g |
| 3 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic3.bmp) | 00r 00g 00b 01r 01g 01b 10r 10g |
| 4 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic4.bmp) | 11r 11g 11b 10r 10g 10b 01r 01g |
| 5 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic5.bmp) | 11r 11b 11g 01r 01b 01g 10r 10b |
| 6 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic6.bmp) | 00r 00b 00g 10r 10b 10g 01r 01b |
| 7 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic7.bmp) | 00r 00b 00g 01r 01b 01g 10r 10b |
| 8 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic8.bmp) | 11r 11b 11g 10r 10b 10g 01r 01b |
| 9 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic9.bmp) | 11b 11g 11r 01b 01g 01r 10b 10g |
| 10 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic10.bmp) | 00b 00g 00r 10b 10g 10r 01b 01g |
| 11 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic11.bmp) | 00b 00g 00r 01b 01g 01r 10b 10g |
| 12 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic12.bmp) | 11b 11g 11r 10b 10g 10r 01b 01g |
| 13 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic13.bmp) | 11g 11b 11r 01g 01b 01r 10g 10b |
| 14 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic14.bmp) | 00g 00b 00r 10g 10b 10r 01g 01b |
| 15 | [Картинка](https://neroid.ru/wp-content/uploads/2020/02/pic15.bmp) | 00g 00b 00r 01g 01b 01r 10g 10b |

Пояснение к кодированию и ключу

Текст представляет из себя обычную Си-строку, то есть последовательность байт, с завершающим нулевым символом.

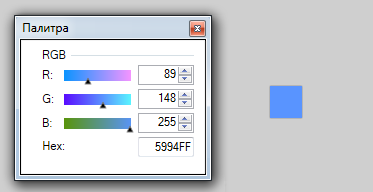
Каждый байт это набор из восьми бит в порядке слева на право от старшего к младшему, т.е. старший бит слева.

Итого, исходный текст представляется в виде последовательности бит.



|  |  |
| --- | --- |
| 1  2 | ABC  01000001 01000010 01000011 00000000 |

Изображение представляется в виде двумерной матрицы, элементом которой является пиксель. Цвет пикселя определяется тремя каналами R(red), G(green), B(Blue). Каждый из каналов задаётся числом в диапазоне [0..255].



*Рис. 6. Представление цвета пикселя в виде разложения по каналам RGB*

Координаты каждого пикселя определяются парой чисел X и Y. Начало отсчёта расположено в левом верхнем углу, X начинается с нуля и увеличивается слева на право, Y увеличивается сверху вниз.



*Рис. 7. Координатная сетка*

Ключ — последовательность из восьми триплетов. Каждая триплет ключа задаётся в формате: X-координата Y-координата Канал (r, g или b) и определяет координаты места в изображении, в которое будет помещён очередной бит текста.



|  |  |
| --- | --- |
| 1 | 11r 11g 11b 01r 01g 01b 10r 10g |

Ключ показывает, в какие координаты (X, Y, Канал) изображения попали бы 8 первых бит текста, если бы они кодировались в изображение 2Х2 пикселя.



|  |  |
| --- | --- |
| 1  2  3  4 | x| 0 | 1 |  y  ---------  0  |rgb|rgb|  1  |rgb|rgb| |

Для вышеуказанного ключа и «изображения» 2Х2 биты будут помещаться в следующем порядке:



|  |  |
| --- | --- |
| 1  2  3  4 | x| 0 | 1 |  y  ---------  0  |rgb|789|  1  |456|123| |

Для вышеуказанного ключа и «изображения» 3Х3 биты будут помещаться в следующем порядке:



|  |  |
| --- | --- |
| 1  2  3  4  5 | x|   0    |   1    |   2    |  y  ----------------------------  0  |r  g  b |22 23 24|19 20 21|  1  |16 17 18|13 14 15|10 11 12|  2  | 7  8  9| 4  5  6| 1  2  3| |

Алгоритм кодирования

Все биты из текста берутся один за одним в порядке от старшего к младшему и подмешиваются к цвету в координате (X, Y, Канал) в порядке указанном в ключе.

Подмешивание осуществляется по следующему алгоритму (оба способа дают один и тот же результат):

Способ первый:

* Заменяем у цвета канала младший бит на бит из текста.  
  Например: цвет канала 152, а значение бита 1. В побитовой форме 152 это 1001 1000 в итоге получаем 1001 1001 (в десятичном виде: 153).
* Получившееся значение сохраняем обратно в картинку.

Способ второй:

* Для выбранного цвета канала берём ближайшее меньшее чётное.  
  Например: цвет канала исходного изображения 131, значит, берём 130. Если 170, так и оставляем.
* Прибавляем к получившемуся числу значение бита из текста.
* Результат сложение сохраняется обратно в картинку.