

Безопасная передача информации является важной задачей в современном цифровом мире. Одним из методов защиты информации является скрытие факта передачи сообщения. Так, произвольный текст может быть спрятан внутри публично опубликованного изображения, пользуясь [избыточностью](#) самого изображения.

Стеганография (от греч. *στεγανός* — скрытый + *γράφω* — пишу; буквально «тайнопись») — способ передачи или хранения информации с учётом сохранения в тайне самого факта такой передачи (хранения). Этот термин ввел в 1499 году Иоганн Тритемий в своем трактате «Стеганография» (Steganographia), зашифрованном под магическую книгу.

[Википедия](#)

Постановка задачи

Необходимо реализовать программу, которая размещает исходный текст в цифровом изображении либо получает текст из изображения, хранящего текст. Изображения рекомендуется использовать в формате хранения без потерь (напр., bmp). Один байт текста будет кодироваться в одном пикселе изображения. Так, один байт текста **10 101 010** будет расположен в пикселе изображения следующим образом:

R: 111100**10**

G: 00001**101**

B: 11001**010**

Алгоритм кодирования информации:

1. Загрузить изображение в память программы
2. Определить количество доступных пикселей изображения
3. Определить максимальную длину текста
4. Загружаем текст в память программы из консоли или файла
5. Если текст превышает максимально допустимую длину, сообщить об этом пользователю и завершить программу
6. Организуем цикл по байтам текста и пикселям изображения, кодируем байт текста по схеме 2-3-3 битов на каждый из каналов пикселя
7. Сохранить изображение на диск в исходном формате

Алгоритм декодирования разрабатывается похожим образом. Для чтения изображения в формате bmp используется бинарное чтение файлов, заголовков которого описан следующей структурой для 32-битной архитектуры:

```
#pragma pack(push, 1)
typedef struct
{
    unsigned char  b1,b2;
    unsigned long  bfSize;
    unsigned short bfReserved1;
    unsigned short bfReserved2;
    unsigned long  bfOffBits;
} BITMAPFILEHEADER;

typedef struct
{
    unsigned int  biSize;
    int          biWidth;
    int          biHeight;
    unsigned short biPlanes;
    unsigned short biBitCount;
    unsigned int  biCompression;
    unsigned int  biSizeImage;
    int          biXPelsPerMeter;
    int          biYPelsPerMeter;
    unsigned int  biClrUsed;
    unsigned int  biClrImportant;
} BITMAPINFOHEADER;
#pragma pack(pop)
```

Требования к программе

Базово программа должна обеспечивать возможность кодирования текста из таблицы ASCII в изображении формата BMP без компрессии на стандартной палитре. В качестве усложнения программы могут быть реализованы:

- Кодирование любого текста из заранее известных кодировок (UTF-16, CP-1251, UTF-8)
- Использование других форматов изображения без потерь
- Шифрование текста перед кодированием в изображение для большей защищённости передачи информации
- Использование других схем записи байта текста в пиксель изображения, например, 1 бит текста в 1 пиксель, таким образом, кодировка является менее заметной, но существенно сокращается размер доступного для кодирования текста

Требования к отчёту

Отчёт выполняет в формате **docx** или **odt** вместе с копией на **pdf**. В отчёте должны присутствовать следующие части:

1. Постановка задачи
2. Теоретическое описание проблем стеганографии

3. Выбранные методы кодирования информации в выполненной работе с подробным описанием алгоритма кодирования и декодирования
4. Части программ, демонстрирующие основные моменты работы программы
5. Описание программы в части работы пользователем: вызов, аргументы, скриншоты и прочее
6. Заключение о проделанной работе

Вопросы для самопроверки

- Зачем используется `#pragma pack(push, 1)`?
 - Как осуществляется чтение и запись бинарных файлов?
 - Что такое формат хранения без потерь?
 - Что из себя представляет формат **bmp**?
 - Что такое бит/байт текста?
 - Какие каналы используются для хранения информации о цвете пикселя?
 - Как в программу передаются аргументы при вызове?
 - Каким могут быть варианты хранения скрытого текста, кроме как в изображениях?
 - Какая информация может быть скрыта?
-

Вопросы, которые стоит гуглить

- `c++ read binary file to struct`
- `why i need pragma push c++`
- `bmp format structure`
- `steganography to bmp`