

[Описание программы](#)

[Код программы](#)

[Код с подробными комментариями](#)

[Советы по улучшению и расширению программы](#)

Описание программы

Код Цезаря представляет собой способ шифрования текстов, основанный на простом правиле: при шифровке каждая буква меняется таким образом, что номер получившейся буквы в алфавите есть номер исходной буквы в алфавите плюс некоторое для всего шифруемого текста определенное значение. При этом алфавит "закольцовывался" - считается, что за последней буквой алфавита идет первая, за первой, естественно, вторая. Например, при сдвиге 2 для русского алфавита имеем: "а" меняется на "в", "б" на "г", "в" на "д", "г" на "е",... "э" на "я", "ю" на "а", "я" на "б".

Данная программа работает с латинским алфавитом, причем заглавные буквы шифруются заглавными, строчные - строчными. Символы, не относящиеся к буквам латинского алфавита, не изменяются.

Программа реализует два режима: шифрование и дешифровка. В первом случае происходит шифрование введенной строки с заданным сдвигом. Во втором случае программа последовательно перебирает варианты возможных расшифровок, по сути шифруя введенную строку все более и более увеличивающимся значением сдвига, предлагая каждый вариант на рассмотрение пользователю. Когда пользователь сочтет очередной вариант верным, дешифровка закончится.

Основной работы программы в обоих режимах является одна и та же функция шифрования.

Код программы

```
#include <iostream>
#include <conio>

const int ABCSize = 26;

const char low_ch[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'};
const char high_ch[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'};

std::string cipher(const std::string &input_s, const int shift)
{
    bool Ok;
```

Код Цезаря

Автор: Андрей

22.07.2009 08:21 - Обновлено 04.12.2009 18:09

```
std::string output_s("");
for(unsigned i = 0; i < input_s.length(); i++)
{
    Ok = false;
    for(int j = 0; j < ABCSize; j++)
    {
        if (input_s[i] == low_ch[j])
        {
            j += shift;
            while (j >= ABCSize) j -= ABCSize;
            while (j < 0) j += ABCSize;
            output_s += low_ch[j];
            Ok = true;
            break;
        }
        else if (input_s[i] == high_ch[j])
        {
            j += shift;
            while (j >= ABCSize) j -= ABCSize;
            while (j < 0) j += ABCSize;
            output_s += high_ch[j];
            Ok = true;
            break;
        }
    }
    if (!Ok) output_s += input_s[i];
}
return output_s;
}
```

```
int main()
{
    std::string s;

    std::cout << "If you want to cipher string, press \"1\", if you want to decode,"
    " press \"2\"";

    bool Ok = false;
    int shift = 0;
    while(!Ok)
    {
        switch(getch())
        {
            case '1':
            {
                std::cout << "\nInput shift: ";
```

```
std::cin >> shift;
std::cout << "Input string to cipher: ";
while (std::cin >> s)
{
    std::cout << cipher(s, shift) << ' ';
    if (std::cin.get() == '\n') break;
}
Ok = true;
} break;
case '2':
{
    bool Done = false;
    std::cout << "\nInput string to decode: ";
    getline(std::cin, s);
    for (int i = 0; i < ABCSize && !Done; i++)
    {
        std::cout << "\nWith shift equal " << i << " we have such string:\n";
        std::cout << cipher(s, i);
        std::cout << "\nIf decoding is done, press '1'\n";
        if (getch() == '1') Done = true;
    }
    Ok = true;
} break;
default: std::cout << "Press either '1' or '2'!\n";
}
}
getch();
}
```

Код программы с комментариями

```
#include <iostream> //Подключаем библиотеку, обрабатывающую
//стандартные потоки ввода/вывода
#include <conio> //Библиотека содержит функцию getch

const int ABCSize = 26; //Размер алфавита

const char low_ch[] = {'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l',
'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z'}; //Массив
//строчных букв, которые шифруются также строчными
const char high_ch[] = {'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L',
'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z'}; //Массив
//заглавных букв, которые шифруются также заглавными
```

Код Цезаря

Автор: Андрей

22.07.2009 08:21 - Обновлено 04.12.2009 18:09

```
std::string cipher(const std::string &input_s, const int shift)
{ //Функция осуществляет сдвиг строки по алфавиту на указанную величину
  bool Ok; //Был ли символ определен как буква алфавита и затем зашифрован
  std::string output_s(""); //Зашифрованная строка, вначале инициализируется
//пустой строкой
  for(unsigned i = 0; i < input_s.length(); i++)
  { //Для всех символов шифруемой строки
    Ok = false; //Вначале сбрасываем значение флага
    for(int j = 0; j < ABCSize; j++)
    { //Перебираем все буквы алфавита на поиск соответствия
      if (input_s[i] == low_ch[j]) //Если символ оказался строчной буквой алфавита
      {
        j += shift; //Сдвигаем букву по алфавиту на указанное значение
        while (j >= ABCSize) j -= ABCSize; //Если значение вышло за диапазон,
        while (j < 0) j += ABCSize; //корректируем его
        output_s += low_ch[j]; //Добавляем полученный символ в конец
//зашифрованной строки
        Ok = true; //Символ был благополучно зашифрован и добавлен в строку
        break; //Перебор для данного символа можно закончить
      }
      else if (input_s[i] == high_ch[j]) //То же самое, если символ оказался
//заглавной буквой алфавита
      {
        j += shift;
        if (j >= ABCSize) j -= ABCSize;
        else if (j < 0) j += ABCSize;
        output_s += high_ch[j];
        Ok = true;
        break;
      }
    }
    if (!Ok) output_s += input_s[i]; //Если символ не является буквой алфавита,
//записываем его без изменений
  }
  return output_s; //По окончании возвращаем получившуюся строку
}

int main()
{
  std::string s; //Шифруемая/дешифруемая строка

  std::cout << "If you want to cipher string, press \'1\', if you want to decode,"
    " press \'2\'";

  bool Ok = false; //Корректна ли нажатая клавиша
  int shift = //Величина сдвига
```

```
while(!Ok) //Пока не будет нажато "1" или "2"
{
    switch(getch())
    {
        case '1': //Если нажато "1", шифруем строку
        {
            std::cout << "\nInput shift: ";
            std::cin >> shift;
            std::cout << "Input string to cipher: ";
            while (std::cin >> s) //Шифруем одним и тем же сдвигом по одному слову
            {
                std::cout << cipher(s, shift) << ' ';
                if (std::cin.get() == '\n') break; //Заканчиваем по нажатию Enter
            }
            Ok = true; //Клавиша была нажата корректно
        } break;
        case '2': //Если нажато "2", пытаемся дешифровать строку
        {
            bool Done = false; //Завершен ли процесс дешифровки
            std::cout << "\nInput string to decode: ";
            getline(std::cin, s); //Считываем всю дешифруемую строку
            for (int i = 0; i < ABCSize && !Done; i++) //Пробуем разные величины сдвига
            //до тех пор, пока не расшифруем или не проверим все возможные его значения
            {
                std::cout << "\nWith shift equal " << i << " we have such string:\n";
                std::cout << cipher(s, i);
                std::cout << "\nIf decoding is done, press \"1\"";
                if (getch() == '1') Done = true; //Строка дешифрована
            }
            Ok = true; //Клавиша была нажата корректно
        } break;
        default: std::cout << "Press either \"1\" or \"2\"!"; //Некорректно нажатая
        //клавиша
    }
}
getch();
}
```

Советы по улучшению и расширению программы

Безусловно, хотелось бы, чтобы можно было шифровать не только тексты, написанные с использованием латиницы, но и кириллические тексты. К сожалению, реализовать это непосредственно с помощью замены массивов алфавита не получится из-за различия кодировки. Попробуйте в тексте программы направить на вывод что-нибудь навроче

"Мой дядя самых честных правил" - и посмотрите, что получится.

Для решения этой проблемы можно каждый раз в начале выполнения программы просить пользователя ввести весь алфавит, который будет использоваться при кодировке (отдельно алфавиты строчных и прописных букв). Однако, вводить оба алфавита всякий раз при запуске программы может оказаться очень нудно и неудобно, поэтому при первом запуске можно, считав оба массива, сохранить их в файл и при всех последующих запусках читать их сразу из файла, не напрягая пользователя.

Впрочем, есть и другой вариант. [Здесь](#) я рассказывал, как можно перевести русский текст, вводимый в текст программы, в русский же текст, выводимый скомпилированной консольной программой. Применив обратную операцию при считывании шифруемой строки, можно подружить эту строку с массивом русских символов, а потом с помощью прямой операции вывести итоговую строку.

Также можно шифровать не только вводимые пользователем строки, но и целые текстовые файлы, запрашивая у пользователя имя исходного и имя результирующего.