Федеральное государственное бюджетное образовательное учреждение высшего образования "Национальный исследовательский университет "МЭИ"

ОТЧЕТ

По лабораторной работе № 4

# «Представление данных в памяти»

ФИО студента Стрюкова Татьяна Александровна

Номер группы А-02-18

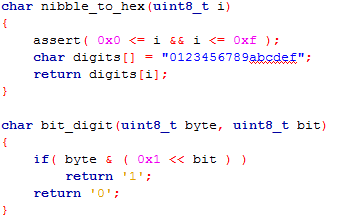
Имя преподавателя Мохов Андрей Сергеевич

Козлюк Дмитрий Александрович

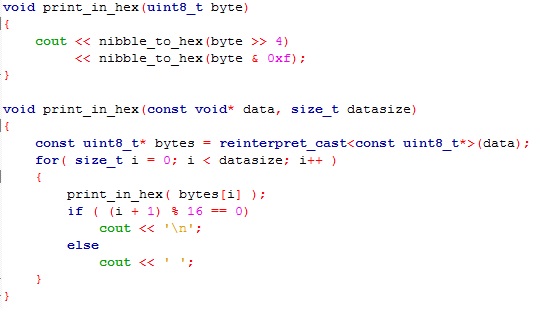
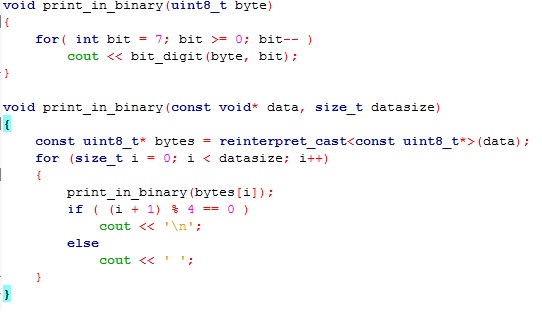
Москва, «НИУ «МЭИ», 2019

Ход работы.

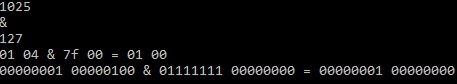
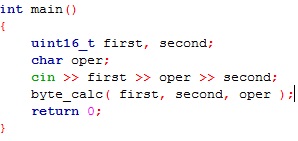
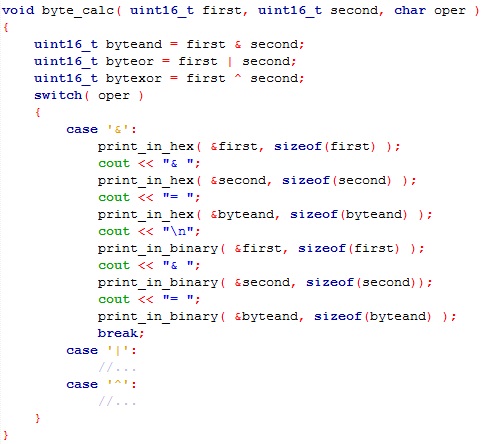
1. Для того, чтобы написать функции для печати отдельных байт и блока данных в шестнадцатеричном и в двоичном представлении, нам потребуются функции печати байтов в двоичном и шестнадцатеричном виде:



1. На основе этих функций составим требуемые:

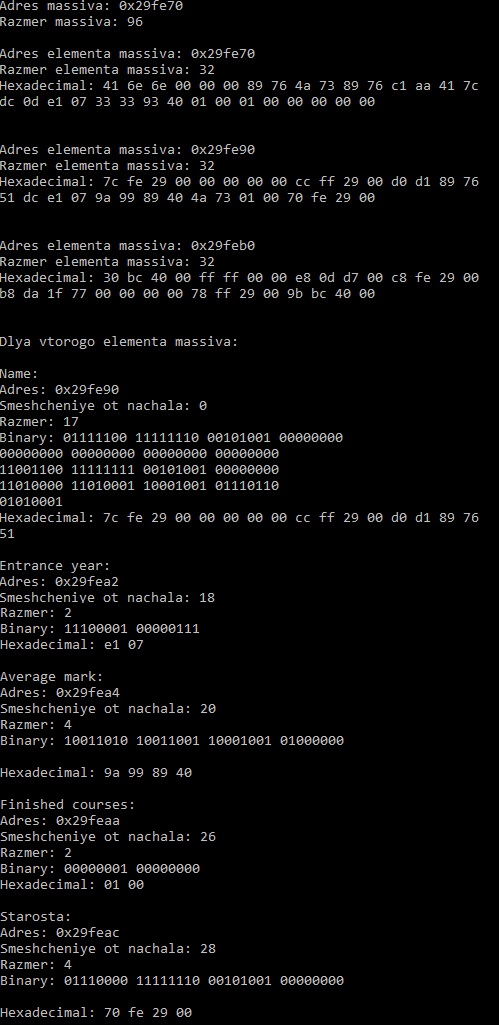


1. Напишем калькулятор побитовых операций. На вход он будет получать численные значения 2 операндов в десятичном виде, и символьное значение оператора. Вывод включает в себя двоичные и шестнадцатеричные формы операндов и результатов операций:



1. Опишем структуру **Student**, объявим и заполним массив из 2 студентов и их старосты.

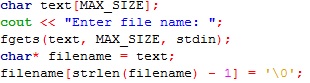
Видим, что адрес массива совпадает с адресом первого элемента, а его размер равен сумме размеров каждого элемента, который в свою очередь, предопределен структурой: 17 байт на имя (каждый символ по 2 байта), 2 байта на год (целое беззнаковое), 4 байта на среднюю оценку (с плавающей запятой), 4 байта на адрес переменной старосты. Смещение полей структуры численно равно размеру предыдущего поля. Шестнадцатеричные адреса элементов также отличаются на значение размера элемента. Элемент-структура в байтовом представлении, вообще говоря, содержит в себе байты его полей по порядку.



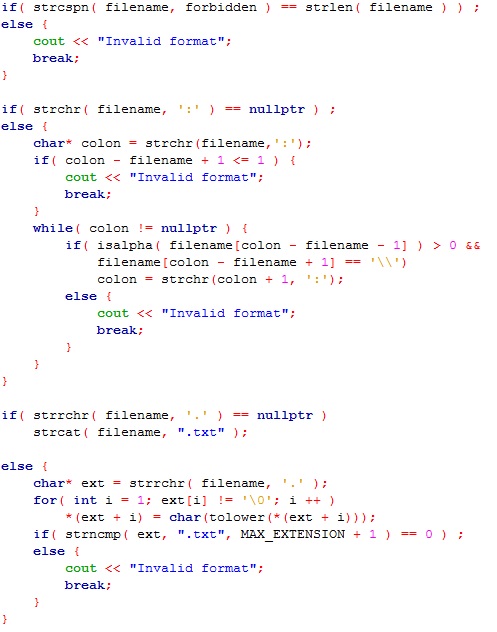
1. Определим некоторые константы: максимальную длину имени файла, запрещенные символы, ширину расширения:

C:\Users\u211-08\Desktop\lab4\1.jpg

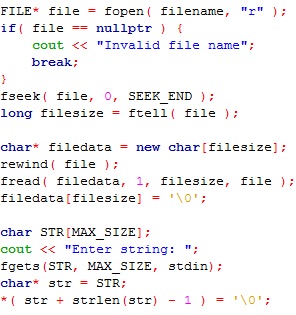
Добавим ввод строки – имени файла. Важно отметить, что необходимо последний символ строки – символ переноса заменить на символ нуль-терминатора:



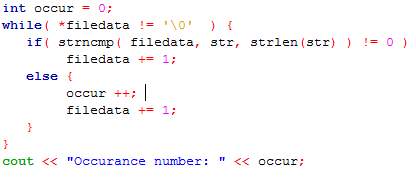
1. Остальную часть, включающую в себя проверки на корректность имени файла, загрузку в память содержимого файла, ввод строки, поиск совпадений строки с содержимым файла, оформим в виде цикла **while**. Так будут выглядеть функции проверки имени файла на корректность:



1. Загрузим в память и сохраним в динамическом массиве-строке содержимое файла, определим размер содержимого в байтах. Добавим запрос строки у пользователя, также не забыв записать в нее нуль-терминатор.



1. Подсчитаем количество вхождений введенной строки в файл, выведем его. Также обязательно нужно закрыть файл, откуда копировалось содержимое, и освободить динамически выделенную память. Допишем оператор **break,** совершающий выход из главного цикла, следовательно завершающий работу программы:



C:\Users\u211-08\Desktop\lab4\3.jpg

Ссылка на репозитарий: <https://github.com/StryukovaTA>