

Практическая работа 1

Указатели, структуры и объединения

1 Дополнительные теоретические сведения

1.1 Сериализация данных

В вычислительной технике **сериализация** – это процесс преобразования структуры данных или состояния объекта в формат, который может быть сохранен (например, файлы на вторичных устройствах хранения, буферы данных на первичных устройствах хранения) или передан (например, потоки данных по компьютерным сетям) и реконструирован позже (возможно, на другом компьютере). Когда результирующая последовательность битов перечитывается в соответствии с форматом сериализации, ее можно использовать для создания семантически идентичного клона исходного объекта.

Противоположной операцией, извлекающей структуру данных из последовательности байтов, является **десериализация**.

1.2 Получение аргументов командной строки

Чтобы получить аргументы, переданные пользователем в командной строке при запуске программы, необходимо объявить функцию `main` следующим образом:

```
int main(int argc, char * argv[])
```

При таком объявлении в аргумент **argc** будет записано количество переданных аргументов командной строки, а в аргумент **argv** – массив строк (в стиле Си, т.е. типа **char***), содержащий представление этих аргументов.

Обращайте внимание, что **первым (нулевым) аргументом всегда передаётся имя исполняемого файла.**

1.3 Побитовые операции

Побитовые операции – операции, применяющиеся отдельно к каждому биту двоичного представления значения.

В таблице 1.1 представлены побитовые операции, которые могут понадобиться для выполнения данной работы, и их запись на языке программирования C.

Таблица 1.1. Запись побитовых операций.

Название операции	Запись на C
Логическое И	&
Логическое ИЛИ	
Исключающее ИЛИ	^
Логическое НЕ	~

Например, чтобы в переменную **y** сохранить значение переменной **x**, к которой применена операция логического ИЛИ с числом (в двоичном представлении) **10101010**, нужно написать следующее:

y = x | 0b10101010;

2 Задание

Для выполнения работы необходимо:

1) Повторно ознакомиться с конспектами семинара 1 и практического занятия 1, материалами лекций, а также рассмотреть теоретическую часть из данного файла (раздел 1).

Внимание! Все настройки, действия и манипуляции, проводимые для выполнения заданий из пункта 2, необходимо фиксировать в отчёте в виде снимков экрана или листингов вводимых команд.

2) С помощью языка программирования Си (не C++) необходимо написать две программы, одна из которых выполняет сериализацию и простое зашифрование введенных пользователем данных, а также запись результата в файл; другая – расшифрование данных из файла, десериализацию этих данных и вывод пользователю в консоль. При этом:

а) Программа должна содержать, как минимум, одну структуру (**struct**) и одно объединение (**union**).

б) Структура должна содержать поля, хранящие данные, вводимые пользователем (по варианту).

в) Объединение должно включать в качестве элементов объект структуры из подпункта б данного списка и массив байт, по размеру равный размеру объекта этой структуры.

г) Преобразование данных, полученных от пользователя, в байты должно производиться с помощью структуры и объединения.

д) Программа должна содержать, как минимум, две функции, одна из которых принимает на вход байт, модифицирует его и возвращает значение модифицированного байта, а вторая принимает на вход указатели на два буфера (массива байт) – входной и выходной –, их размер, а также функцию обратного вызова и записывать результат применения функции к элементам входного буфера на соответствующие позиции выходного буфера.

е) Простое шифрование данных, полученных от пользователя, должно производиться с помощью двух функций, описанных в подпункте д этого списка.

ж) Программа должна сохранять значение зашифрованных данных в массив байт, выделенный в динамической памяти (на куче).

3) Для выполнения задания может использоваться любая среда разработки. При выполнении необходимо учитывать следующие требования:

а) Задача считается решённой, если необходимое по варианту действие выполняется для всех допустимых входных данных.

б) Для работы с файлами и консолью должны использоваться только специфичные для языка Си функции из заголовочного файла `stdio.h`. Использование аналогов из языка C++ не допускается. Например, вывод в консоль следует делать с помощью функции **`printf`**.

в) Программа должна корректно завершаться, не вызывая аварийный останов.

г) Программа должна брать название файла из аргумента, переданного при запуске в консоли. В случае, если количество переданных аргументов не равно ожидаемому, программа должна вывести подсказку для пользователя, поясняющую правила её (программы) использования.

д) Программа должна запрашивать данные для сериализации у пользователя после запуска. Для этого должно быть выведено приглашение ко вводу, описывающее, какие данные пользователю необходимо ввести в данный момент.

е) Возвращаемые значения **всех** вызываемых функций должны проверяться на предмет возникновения ошибок. В случае возникновения ошибки необходимо вывести сообщение, оповещающее пользователя о произошедшем, содержащее в обязательном порядке код ошибки и её текстовое описание. В случае, если в результате

возникшей ошибки программа должна быть завершена, перед завершением необходимо освободить все занятые ресурсы (очистить выделенную память, закрыть открытые файлы).

ж) Допускается ограничить размер буфера входных данных, вводимых пользователем с консоли. В таком случае приглашение ко вводу должно содержать информацию об этом ограничении.

4) Оформить отчёт в соответствии с требованиями из файла «Требования к оформлению отчёта по практической работе». Не забыть прикрепить к отчёту **исходные коды** обеих программ **в виде приложения**. В отчёте необходимо отразить, как минимум:

а) Результаты работы программы в виде снимков экрана, демонстрирующих работу программы для 2-3 вариантов верных и для каждого типа неверных входных данных (чтобы продемонстрировать все предусмотренные сообщения об ошибках).

б) Процесс сборки и запуска программы.

в) Принципы работы алгоритмов, реализованных в коде и саму реализацию этих алгоритмов в виде листингов кода.

г) Изменения ключевых значений (в переменных) в ходе работы программы.

5) Защитить работу на занятии. Во время защиты необходимо:

а) Продемонстрировать работу программы.

б) При необходимости изменить входные данные и продемонстрировать, что программа отработывает верно на новых данных.

в) Ответить на вопросы по логике работы программы (почему и зачем была написана та или иная строка, каково её назначение и вклад в решение данной задачи, что будет, если её убрать).

г) Ответить на вопросы по теоретической части (**в том числе по лекциям и семинарам**).

6) Дополнительное задание 1: Изучить стандарт хранения двоичных данных в текстовом виде Base64. Реализовать сериализацию данных в файл и десериализацию их из файла согласно этому стандарту (вместо двоичной записи). Даёт два дополнительных балла. **Не обязательно к выполнению!**

7) Дополнительное задание 2: Реализовать сериализацию данных в файл и десериализацию их из файла в виде последовательности цифр 1 и 0, соответствующих значениям битов сериализованных данных. Даёт один дополнительный балл. **Не обязательно к выполнению!**

Внимание! В случае, если сумма набранных баллов превышает 10, в качестве результирующей оценки выставляется 10 баллов. Дополнительные бонусы за превышение максимального количества баллов не предусмотрены.

3 Варианты заданий

№	Вводимые данные	Шифрование и расшифровывание
1	Наименование товара и три отдельных дробных числовых параметра – его длина, ширина и высота.	Шифрование: от численного представления каждого байта отнимается 1. Расшифровывание: к численному представлению каждого байта прибавляется 1.
2	Наименование товара и три отдельных целых числовых параметра – его длина, ширина и высота.	Шифрование: от численного представления каждого байта отнимается 10. Расшифровывание: к численному представлению каждого байта прибавляется 10.
3	Имя человека и его адрес, состоящий из: названия улицы, номера дома и номера квартиры.	Шифрование: к каждому байту применяется операцию побитового исключающего ИЛИ с двоичным числом 10101010. Расшифровывание: аналогично шифрованию.
4	Фамилия, имя и отчество человека в виде отдельных параметров и его целочисленных идентификатор в некоторой системе.	Шифрование: к каждому байту применяется операцию побитового исключающего ИЛИ с двоичным числом

		01010101. Расшифровывание: аналогично шифрованию.
5	Название торгового центра и список из 20 номеров свободных парковочных мест.	Шифрование: инверсия всех бит байта. Расшифровывание: аналогично шифрованию.
6	Название населённого пункта и два отдельных дробных параметра – его координаты на карте.	Шифрование: инверсия старшей половины бит байта. Расшифровывание: аналогично шифрованию.
7	URL-адрес сайта и список из 4 чисел-компонент его IP-адреса.	Шифрование: инверсия младшей половины бит байта. Расшифровывание: аналогично шифрованию.
8	Наименование товара и список дробных чисел – его средняя цена за каждый из последних 20 лет.	Шифрование: к численному представлению каждого байта прибавляется 2. Расшифровывание: от численного представления каждого байта отнимается 2.
9	Название населённого пункта и список из 10 целочисленных номеров маршрутов общественного транспорта, которые в нём проходят.	Шифрование: к численному представлению каждого байта прибавляется 12. Расшифровывание: от численного представления каждого байта отнимается 12.
10	Наименование товара и список логических значений – его наличие в каждом из 15 магазинов города.	Шифрование: к каждому байту применяется операцию побитового исключающего ИЛИ с двоичным числом

		10101010. Расшифровывание: аналогично шифрованию.
11	Название аэропорта и список из трёх символьных значений – его обозначения на латинице (например, SVO или DME).	Шифрование: к каждому байту применяется операцию побитового исключающего ИЛИ с двоичным числом 01010101. Расшифровывание: аналогично шифрованию.
12	Название месяца, номер дня в месяце и численное отклонение средней температуры в этот день от среднего за предыдущий год.	Шифрование: инверсия всех бит байта. Расшифровывание: аналогично шифрованию.
13	Название месяца, номер дня в месяце и название дня недели, выпадающего на этот день.	Шифрование: инверсия старшей половины бит байта. Расшифровывание: аналогично шифрованию.
14	Название учебной дисциплины и список 16 логических значений, показывающих, преподаётся ли эта дисциплина в каждом из модулей.	Шифрование: инверсия младшей половины бит байта. Расшифровывание: аналогично шифрованию.
15	Название учебной дисциплины и список 30 целочисленных оценок.	Шифрование: инверсия младшего бита байта. Расшифровывание: аналогично шифрованию.

4 Список полезных источников

1. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Глава 6 – Массивы.
2. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Глава 7 – Указатели (в частности, раздел 7.11 – Указатели на функции).
3. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Разделы 10.1 – 10.4 – Структуры.
4. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Раздел 10.8 – Объединения.
5. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Раздел 10.9 – Побитовые операции.
6. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Глава 11 – Файлы.
7. Х. Дейтел, П. Дейтел – «Как программировать на Си». 4 издание.
Главы 8 – 9 – Строки и консольный ввод/вывод.