

**МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ**

**КУРСОВАЯ РАБОТА
по дисциплине «Программирование»
Тема: Обработка строк**

Студента гр. 4388

Харченко Я.К.

Преподаватель

Гаврилов А.В.

Санкт-Петербург

2024

ЗАДАНИЕ НА КУРСОВУЮ РАБОТУ

Студента Харченко Я.К.

Группа 4388

Тема работы: Обработка строк

Исходные данные:

Вывод программы должен быть произведен в стандартный поток вывода: *stdout*. Ввод данных в программе в стандартный поток ввода: *stdin*. В случае использования Makefile название исполняемого файла должно быть: *cw*. После вывода информацию о варианте курсовой работе программа ожидает ввода пользователем числа – номера команды:

- 0 – вывод текста после первичной обязательной обработки (если предусмотрена заданием данного уровня сложности)
- 1 – вызов функции под номером 1 из списка задания
- 2 – вызов функции под номером 2 из списка задания
- 3 – вызов функции под номером 3 из списка задания
- 4 – вызов функции под номером 4 из списка задания
- 5 – вывод справки о функциях, которые реализует программа.

Программа не должна выводить никаких строк, пока пользователь не введет число.

Дата выдачи задания: 07.11.2024

Дата сдачи реферата: 27.11.2024

Дата защиты реферата: 06.12.2024

Студента

Харченко Я.К.

Преподаватель

Гаврилов А.В.

АННОТАЦИЯ

В данном курсовом проекте рассматривается разработка консольной программы для выполнения ряда текстовых операций над строками, с использованием языка программирования C. Основное содержание работы включает реализацию четырех функциональных методов: исключение символов одной строки из другой, преобразование символов в кодовые значения, проверку палиндрома и форматирование строк с заглавной буквой. В качестве метода разработки используется реализация функций без использования стандартных и сторонних библиотек, что позволяет углубиться в изучение работы с памятью и строковыми данными. Результатом работы станет программа, которая после ввода номера операции позволяет пользователю взаимодействовать с текстом в заданном формате, обеспечивая также обработку ошибок и защиту от выхода за границы буфера. Программа будет полезна для изучения основ работы с строками и обработки пользовательского ввода.

ЦЕЛЬ И ЗАДАЧИ

Цель:

Целью данной работы является разработка программы для обработки текстовых данных с реализацией функциональности преобразования строк на основе пользовательского ввода.

Задачи:

Для достижения поставленной цели требуется решить следующие задачи:

1. Изучить требования задания к функциональности программы и форматам ввода/вывода.
2. Реализовать функцию, исключающую из второй строки символы, присутствующие в первой строке.
3. Реализовать функцию, преобразующую строку в последовательность кодов символов.
4. Реализовать функцию проверки, является ли одна строка обратной записью другой.
5. Реализовать функцию преобразования строки, чтобы каждое слово начиналось с заглавной буквы, остальные буквы — строчные.
6. Организовать защиту от выхода за границу буфера и обработку возможных ошибок ввода.
7. Добавить функциональность вывода справки о работе программы.
8. Выполнить тестирование реализованных функций для проверки корректности их работы.
9. Оформить работу в строгом соответствии с заданным форматом вывода.

СОДЕРЖАНИЕ

| | | |
|------|---|----|
| | Введение | 8 |
| 1. | Функция 0: Ввод строки | 9 |
| 2. | Функция 1: Исключение символов одной строки из другой | 10 |
| 3. | Функция 2: Преобразование символов в кодовые значения | 11 |
| 4. | Функция 3: Проверка на палиндром | 12 |
| 5. | Функция 4: Форматирование строк с заглавной буквой | 13 |
| 6. | Функция 5: Описание функций | 14 |
| 7. | Функция main | 15 |
| 8. | Инструкция по запуску приложения | 16 |
| 9. | Примеры запуска приложения с разными входными данными | 17 |
| 9.1. | Примеры работы Функции 1 | 17 |
| 9.2. | Примеры работы Функции 2 | 18 |
| 9.3. | Примеры работы Функции 3 | 18 |
| 9.4. | Примеры работы Функции 4 | 19 |
| 9.5. | Примеры работы Функции 5 | 20 |

| | | |
|-------|--|----|
| 10. | Примеры обработок ошибок | 21 |
| 10.1. | Ошибка переполнения буфера (Buffer Overflow) | 21 |
| 10.2. | Ошибка обработки ввода | 21 |
| 11. | Заключение | 22 |
| 12. | Список использованных источников | 23 |

ВВЕДЕНИЕ

Целью данной курсовой работы является разработка консольного приложения, способного выполнять различные текстовые операции над строками, что является важным аспектом при работе с данными в программировании. Основные задачи работы включают реализацию четырех функций: 1) исключение символов одной строки из другой, 2) преобразование символов строки в их кодовые представления, 3) проверка, является ли одна строка обратной записью другой, и 4) форматирование строки так, чтобы первая буква каждого слова была заглавной. Для решения этих задач используется язык программирования C с акцентом на ручное управление памятью и обработку строк без применения стандартных библиотек, что позволяет глубже понять внутренние механизмы работы с данными.

1. ФУНКЦИЯ 0: ВВОД СТРОКИ

Назначение: Считывает строку ввода в массив до максимальной длины (MaxLen) в 1000 символов.

Детали:

- Считывает символы один за другим с помощью getchar().
- Пропускает начальный символ новой строки, если он появляется первым.
- Останавливается при встрече символа новой строки.
- Проверяет переполнение буфера и выводит сообщение об ошибке, если ввод превышает MaxLen.
- Возвращает количество считанных символов.

```
int input(unsigned char(*arr)) {
    unsigned char t = ' ';
    int i = 0;
    while ((t = getchar()) && i < MaxLen) {
        if ((t == '\n') && i == 0)
            continue;
        if (t == '\n')
            break;
        arr[i] = t;
        i++;
    }
    if (i >= MaxLen) {
        printf("Error: Buffer overflow.");
        return -1;
    }
    return i;
}
```

2. ФУНКЦИЯ 1: ИСКЛЮЧЕНИЕ СИМВОЛОВ ОДНОЙ СТРОКИ ИЗ ДРУГОЙ

Назначение: Принимает две строки и удаляет из второй строки все символы, которые встречаются в первой строке.

Детали:

- Считывает две строки a и b с помощью функции input.
- Отмечает символы из первой строки a в массиве list.
- Проходит по второй строке b и выводит символы, которые не отмечены в массиве list.

```
void function1() {
    unsigned char a[MaxLen];
    unsigned char b[MaxLen];
    int list[256] = {0};
    int lenA = input(a);
    int lenB = input(b);
    if (lenA == -1 || lenB == -1)
        return;
    for (int i = 0; i < lenA; i++)
    {
        list[a[i]] = 1;
    }
    for (int i = 0; i < lenB; i++)
    {
        if (list[b[i]] != 1) {
            putchar(b[i]);
        }
    }
}
```

3. ФУНКЦИЯ 2: ПРЕОБРАЗОВАНИЕ СИМВОЛОВ В КОДОВЫЕ ЗНАЧЕНИЯ

Назначение: Принимает строку и выводит ASCII-код каждого символа, разделённые пробелами.

Детали:

- Считывает строку а с помощью функции input.
- Проходит по строке а и выводит ASCII-код каждого символа.

```
void function2() {  
    unsigned char a[MaxLen];  
    int lenA = input(a);  
  
    if (lenA == -1)  
        return;  
    for (int i = 0; i < lenA; i++)  
        printf("%d ", a[i]);  
}
```

4. ФУНКЦИЯ 3: ПРОВЕРКА НА ПАЛИНДРОМ

НАЗНАЧЕНИЕ: ПРИНИМАЕТ ДВЕ СТРОКИ И ПРОВЕРЯЕТ, ЯВЛЯЕТСЯ ЛИ ПЕРВАЯ СТРОКА ОБРАТНОЙ ЗАПИСЬЮ ВТОРОЙ.

ДЕТАЛИ:

- СЧИТЫВАЕТ ДВЕ СТРОКИ А И В С ПОМОЩЬЮ ФУНКЦИИ INPUT.
- СРАВНИВАЕТ ДЛИНЫ ДВУХ СТРОК.
- ПРОХОДИТ ПО ПЕРВОЙ СТРОКЕ А И СРАВНИВАЕТ КАЖДЫЙ СИМВОЛ С СООТВЕТСТВУЮЩИМ СИМВОЛОМ С КОНЦА ВТОРОЙ СТРОКИ В.
- ВЫВОДИТ "YES", ЕСЛИ ПЕРВАЯ СТРОКА ЯВЛЯЕТСЯ ОБРАТНОЙ ЗАПИСЬЮ ВТОРОЙ, ИНАЧЕ ВЫВОДИТ "NO".

```
void function3() {
    unsigned char a[MaxLen];
    unsigned char b[MaxLen];
    int lenA = input(a);
    int lenB = input(b);

    if (lenA == -1 || lenB == -1)
        return;

    if (lenA != lenB)
    {
        printf("False");
        return;
    }

    for (int i = 0; i < lenA; i++)
    {
        if (a[i] != b[lenA - i - 1])
        {
            printf("False");
            return;
        }
    }
    printf("True");
}
```

5. ФУНКЦИЯ 4: ФОРМАТИРОВАНИЕ СТРОК С ЗАГЛАВНОЙ БУКВЫ

НАЗНАЧЕНИЕ: ПРИНИМАЕТ СТРОКУ И ПРЕОБРАЗУЕТ КАЖДОЕ СЛОВО ТАК, ЧТОБЫ ПЕРВАЯ БУКВА БЫЛА ЗАГЛАВНОЙ, А ОСТАЛЬНЫЕ — СТРОЧНЫМИ.

ДЕТАЛИ:

- СЧИТЫВАЕТ СТРОКУ А С ПОМОЩЬЮ ФУНКЦИИ INPUT.
- ПРЕОБРАЗУЕТ ПЕРВЫЙ СИМВОЛ В ЗАГЛАВНЫЙ, ЕСЛИ ОН ЯВЛЯЕТСЯ СТРОЧНОЙ БУКВОЙ.
- ПРОХОДИТ ПО СТРОКЕ И:
 - ПРЕОБРАЗУЕТ СИМВОЛ В ЗАГЛАВНЫЙ, ЕСЛИ ОН СЛЕДУЕТ ЗА ПРОБЕЛОМ.
 - ПРЕОБРАЗУЕТ СИМВОЛ В СТРОЧНЫЙ, ЕСЛИ ОН НЕ СЛЕДУЕТ ЗА ПРОБЕЛОМ.
 - ВЫВОДИТ ИЗМЕНЕННУЮ СТРОКУ.

```
void function4() {
    unsigned char a[MaxLen];
    int lenA = input(a);

    if (lenA == -1)
        return;

    if (a[0] >= 97 && a[0] <= 122)
        a[0] -= 32;

    for (int i = 1; i < lenA; i++)
        if (a[i - 1] == ' ') {
            if (a[i] >= 97 && a[i] <= 122)
                a[i] -= 32;
        }
        else {
            if (a[i] >= 65 && a[i] <= 90)
                a[i] += 32;
        }

    for (int i = 0; i < lenA; i++)
        printf("%c", a[i]);
}
```

6. ФУНКЦИЯ 5: ОПИСАНИЕ ФУНКЦИЙ

НАЗНАЧЕНИЕ: ВЫВОДИТ ОПИСАНИЯ ДРУГИХ ФУНКЦИЙ.

ДЕТАЛИ:

○ ПРЕДОСТАВЛЯЕТ КРАТКОЕ ОПИСАНИЕ КАЖДОЙ ИЗ ЧЕТЫРЁХ ДРУГИХ ФУНКЦИЙ.

```
void function5() {  
    printf(" 1 FUNCTION: The function accepts two strings through a line break  
        \nand excludes from the second string all the characters that are in the  
        \nfirst string.\n");  
    printf(" 2 FUNCTION: The function accepts a string and replaces each  
        character \nwith its code. The codes are output, separated by a space,  
        while each space \nin the original string is converted to a code in the  
        resulting \nstring.\n");  
    printf(" 3 FUNCTION: The function accepts two strings and checks whether the  
        \nfirst string is a reverse record of the second. The strings are  
        supplied \nthrough a line break.\n");  
    printf(" 4 FUNCTION: The function accepts a string and converts it so that  
        in \neach word the first letter is capitalized, and the rest are  
        lowercase. A word \nis a set of letters and numbers, words are separated  
        by at least \none space.");  
}
```

7. ФУНКЦИЯ MAIN

НАЗНАЧЕНИЕ: УПРАВЛЯЕТ ХОДОМ ВЫПОЛНЕНИЯ ПРОГРАММЫ В ЗАВИСИМОСТИ ОТ ВВОДА ПОЛЬЗОВАТЕЛЯ.

ДЕТАЛИ:

- ВЫВОДИТ ПРИВЕТСТВЕННОЕ СООБЩЕНИЕ.
- СЧИТЫВАЕТ ЦЕЛОЕ ЧИСЛО ОТ ПОЛЬЗОВАТЕЛЯ ДЛЯ ОПРЕДЕЛЕНИЯ, КАКУЮ ФУНКЦИЮ ВЫЗВАТЬ.
- ВЫЗЫВАЕТ СООТВЕТСТВУЮЩУЮ ФУНКЦИЮ НА ОСНОВАНИИ ВВОДА ПОЛЬЗОВАТЕЛЯ.

```
int main(void)
{
    printf("Course work for option 3.5, created by Yaroslav Kharchenko\n");
    int a;
    scanf("%d", &a);
    switch (a)
    {
        case 0:
            break;
        case 1:
            function1();
            break;
        case 2:
            function2();
            break;
        case 3:
            function3();
            break;
        case 4:
            function4();
            break;
        case 5:
            function5();
            break;
        default:
            printf("Error: Can't find number of command.");
            break;
    }
    printf("\n\n");
    return 0;
}
```

8. ИНСТРУКЦИЯ ПО ЗАПУСКУ ПРИЛОЖЕНИЯ

1. УБЕДИТЕСЬ, ЧТО У ВАС УСТАНОВЛЕН КОМПИЛЯТОР ЯЗЫКА C (НАПРИМЕР, GCC).
2. СОХРАНИТЕ ПРЕДЛОЖЕННЫЙ КОД В ФАЙЛ С ИМЕНЕМ `main.c`.
3. ОТКРОЙТЕ ТЕРМИНАЛ ИЛИ КОМАНДНУЮ СТРОКУ И ПЕРЕЙДИТЕ В ДИРЕКТОРИЮ, ГДЕ СОХРАНЕН ФАЙЛ `main.c`.
4. СКОМПИЛИРУЙТЕ КОД С ПОМОЩЬЮ КОМАНДЫ:

```
gcc -o my_program main.c
```

5. ЗАПУСТИТЕ ПРОГРАММУ ИСПОЛЬЗУЯ КОМАНДУ:

```
./my_program
```

6. ВВЕДИТЕ ЦИФРУ 5 ЧТОБЫ УВИДЕТЬ, ЧТО ДЕЛАЕТ КАЖДАЯ ИЗ ФУНКЦИЙ. ПОСЛЕ ЧЕГО ПЕРЕЗАПУСТИТЕ ПРОГРАММУ И ВВЕДИТЕ НОМЕР НУЖНОЙ ФУНКЦИИ.

9. ПРИМЕРЫ ЗАПУСКА ПРИЛОЖЕНИЯ С РАЗНЫМИ ВХОДНЫМИ ДАННЫМИ

9.1. Примеры работы Функции 1

1-ЫЙ ПРИМЕР:

```
Course work for option 3.5, created by Yaroslav Kharchenko
1
abc
abcdefg
defg
```

(Из второй строки удаляются символы 'а', 'в', и 'с', которые присутствуют в первой строке.)

2-ОЙ ПРИМЕР:

```
Course work for option 3.5, created by Yaroslav Kharchenko
1
xyz
hello world!
hello world!
```

(В этой ситуации из второй строки не удаляется ни один символ, так как ни один из 'х', 'у', или 'z' не присутствует.)

9.2. Примеры работы Функции 2

1-ый пример:

```
Course work for option 3.5, created by Yaroslav Kharchenko
2
Привет
208 159 209 128 208 184 208 178 208 181 209 130
```

2-ой пример:

```
Course work for option 3.5, created by Yaroslav Kharchenko
2
Hello, World!
72 101 108 108 111 208 44 32 208 166 209 87 111 114 108 100 33
```

9.3. Примеры работы Функции 3

1-ый пример:

```
Course work for option 3.5, created by Yaroslav Kharchenko
3
hello
olleh
True
```

2-ой пример:

```
Course work for option 3.5, created by Yaroslav Kharchenko
3
level
level
True
```

3-ий ПРИМЕР:

```
Course work for option 3.5, created by Yaroslav Kharchenko
3
test
sett
False
```

9.4. Примеры работы Функции 4

1-ый ПРИМЕР:

```
Course work for option 3.5, created by Yaroslav Kharchenko
4
hello world
Hello World
```

2-ой ПРИМЕР:

```
Course work for option 3.5, created by Yaroslav Kharchenko
4
tHIS is A teSt
This Is A Test
```

9.5. ПРИМЕР РАБОТЫ ФУНКЦИИ 5

Course work for option 3.5, created by Yaroslav Kharchenko

5

1 FUNCTION: The function accepts two strings through a line break and excludes from the second string all the characters that are in the first string.

2 FUNCTION: The function accepts a string and replaces each character with its code. The codes are output, separated by a space, while each space in the original string is converted to a code in the resulting string.

3 FUNCTION: The function accepts two strings and checks whether the first string is a reverse record of the second. The strings are supplied through a line break.

4 FUNCTION: The function accepts a string and converts it so that in each word the first letter is capitalized, and the rest are lowercase. A word is a set of letters and numbers, words are separated by at least one space.

10. ПРИМЕРЫ ОБРАБОТОК ОШИБОК

10.1. Ошибка переполнения буфера (Buffer Overflow)

В функции ``input`` имеется защитная логика, которая предупреждает о переполнении, если пользователь вводит больше символов, чем допустимо (больше ``MaxLen``). Однако фактически, программа не завершит свою работу правильно после вывода сообщения об ошибке. Это может привести к непредсказуемому поведению программы или даже к уязвимостям при выполнении, если дополнительно не обработать такой случай.

10.2. Ошибка обработки ввода

В функции ``main`` присутствует возможность возникновения ошибки при выборе числа для вызова соответствующей функции. Если пользователь вводит некорректное значение (например, нечисловой символ или число, выходящее за пределы предусмотренных случаев), может возникнуть неопределенное поведение программы. В случае ввода числа, не принадлежащего диапазону от 0 до 5, программа выводит сообщение об ошибке, но не предоставляет пользователю возможность повторного ввода. Это может создать путаницу и негативный опыт пользователя, так как неясно, что можно делать дальше.

ЗАКЛЮЧЕНИЕ

В данной работе был разработан набор функций для обработки строк: исключение символов, преобразование символов в коды, проверка на обратный порядок, а также преобразование регистра букв для форматирования текста. Все функции успешно реализовали свои цели, в том числе удовлетворяли требованиям пользователя, но были выявлены проблемы, такие как ошибки обработки ввода и потенциальное переполнение буфера. Эти недостатки указывают на необходимость улучшений, особенно в области обработки пользовательских вводов и безопасности программы. Таким образом, поставленные цели были в основном достигнуты, однако требуют доработки для повышения надежности и удобства использования.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Керниган, Б. У. и Ричи, Д. М. "Язык программирования С". – 2-е издание, 1988.
2. Паттерсон, Д. А. и Xенnessy, Дж. Л. "Компьютерные системы: проектирование и архитектура". – 5-е издание, 2013.
3. "C Standard Library" – документация на официальном сайте ISO.
4. "Buffer Overflow Attacks: Detecting and Preventing" – статьи и исследования о безопасности и уязвимостях, связанных с переполнением буфера.
5. Онлайн-ресурсы и форумы, такие как Stack Overflow, для получения советов по программированию и решения конкретных проблем.