МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

### федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

(ГУАП)

КАФЕДРА ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ И СЕТЕЙ

Преподаватель

канд. техн. наук, доцент Л.Н. Бариков

Отчёт

по лабораторной работе №15

по дисциплине ОСНОВЫ ПРОГРАММИРОВАНИЯ

на тему: «**Стек, дек, очередь**»

Работу выполнил

студент гр. 4141 В.С. Сыворотнев

Санкт-Петербург

2022

***Цель лабораторной работы:*** *изучение способов создания и принципов использования односвязных линейных списков; изучение стандартных средств языка C/C++ для работы с динамической памятью; совершенствование навыков модульного программирования на языке C/C++ при решении задач обработки линейных списков; изучение способов разработки многофайловых проектов.*

***Задание на программирование:*** *используя технологию процедурного программирования разработать программу обработки односвязных линейных списков с числом элементов в списке* *не менее десяти в соответствии с индивидуальным заданием.*

***Вариант № 45:***

Дана произвольная строка. Под *словом* понимается последовательность цифр. Между соседними словами - не менее одного пробела. Перед первым и за последним словом строки произвольное число пробелов.

Используя одну ***очередь*,** за один просмотр исходной строки напечатать сначала все слова с четным значением суммы входящих в нее цифр, затем все слова с нечетным значением суммы цифр, сохраняя исходный порядок в каждой группе. Например:

- исходная последовательность: 15 21 32 48 59 65 76 88 98;

- результат: 15 48 59 88 21 32 65 76 98.

***Порядок выполнения работы:***

1. Получить у преподавателя индивидуальное задание.

2. Разработать математическую модель: описать с помощью формул и рисунков вид используемой динамической структуры и процессы её создания и использования.

3. Построить схему алгоритма решения задачи.

4. Использовать функции, реализующие полный набор операций для этой структуры:

- допустимые операции для стека: инициализация, проверка на пустоту, добавление нового элемента в начало, извлечение элемента из начала;

- допустимые операции для очереди: инициализация, проверка на пустоту, добавление нового элемента в конец, извлечение элемента из начала;

- допустимые операции для дека: инициализация, проверка на пустоту, добавление нового элемента в начало, добавление нового элемента в конец, извлечение элемента из начала, извлечение элемента из конца.

5. Создать многофайловый проект на языке C/C++.

6. Проверить и продемонстрировать преподавателю работу программы на полном наборе тестов. Обеспечить одновременный показ на экране входной и выходной информации.

7. Выходные данные должны выводиться на экран с пояснениями. Операторы вывода результатов работы должны находиться либо в функции main(), либо в специальной функции вывода результата, вызов которой осуществляется из функции main().

8. Оформить отчет о лабораторной работе в составе: постановка задачи, математическая модель, схема алгоритма решения основной (основных) задач, текст программы, контрольные примеры.

9. Текст программы в отчете не должен представлять из себя скриншот.

10. Скриншоты тестов должны легко читаться. Все их неинформативные части должны быть удалены.

***Описание процесса решения***

В процессе решения задействованы 2 функции – isOddSum() и generateAnswer(). Функиця isOddSum() проверят сумму цифр числа на нечетность, а функция generateAnswer() выполняет непосредственную работу с очередью и формирование результирующей строки. Работа алгоритма состоит в следующем:

На вход поступает строка, затем каждое число проверяется на нечетность суммы цифр помощью функции isOddSum(), и если она является нечетной, то данное число отправляется в конец очереди. Иначе, записывается в результирующую строку (динамический массив строк). После идет опустошение очереди с начала и окончательное завершение формирования результирующего динамического массива строк.

***Текст программы***

***Заголовочный файл queue.h***

#include <string>

#include <vector>

#include <sstream>

#include <queue>

bool isOddSum(std::string\_view word) {

int sum = 0;

for (auto ch: word) {

sum += ch - '0';

}

return sum % 2;

}

std::vector<std::string> generateAnswer(const std::string& line) {

std::deque<std::string> queue;

std::vector<std::string> result;

std::stringstream stream(line);

for (std::string word; stream >> word;) {

if (isOddSum(word)) {

queue.push\_back(word);

} else {

result.push\_back(word);

}

}

while (!queue.empty()) {

result.push\_back(queue.front());

queue.pop\_front();

}

return result;

}

***Основной файл main.cpp***

#include <iostream>

#include <fstream>

#include <string>

#include "queue.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "Russian");

ifstream fin("input.txt");

ofstream fout("output.txt");

cout << "Входной файл: " << endl;

for (string str; getline(fin, str);) {

cout << str << endl;

}

fin.close();

fin.open("input.txt");

for (string line; getline(fin, line);) {

for (string\_view word: generateAnswer(line)) {

fout << word << " ";

}

fout << endl;

}

fin.close();

fout.close();

fin.open("output.txt");

cout << "Выходной файл: " << endl;

for (string str; getline(fin, str);) {

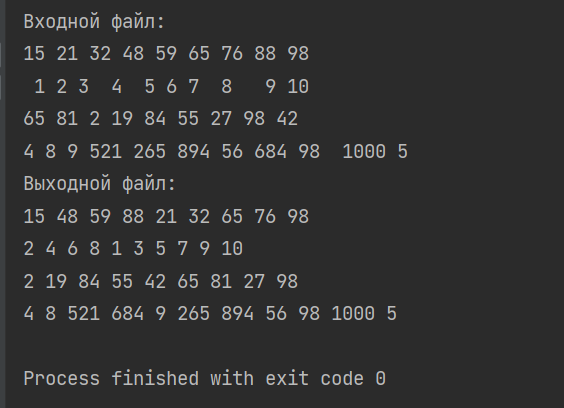
cout << str << endl;

}

return 0;

}

***Скриншот результатов выполнения программы***

******

***Вывод:*** *я изучил способы создания и принципы использования динамических структур данных типа стек, дек, очередь; изучил стандартные средства языка C/C++ для работы с динамической памятью; усовершенствование навыки модульного программирования на языке C/C++ при решении задач обработки динамических структур данных; усовершенствовал способы разработки многофайловых проектов.*