**Отчет по лабораторной работе №13** по курсу Вычислительные системы

Студент группы M8О-106Б-22, **Ларченко Арсений Олегович**, № по списку **16**

Контакты e-mail: larchenkoao@gmail.com

Работа выполнена: «13» декабря 2022г. Преподаватель: Дубинин А.В. , каф. 806

Входной контроль знаний с оценкой

Отчет сдан «26» декабря 2022 г., итоговая оценка

Подпись преподавателя

# Тема: Множества

# 

1. **Цель работы:**  Составить проверки характеристик введенных последовательностей слов

# 

# 

# 

1. **Задание** (*вариант №****27***)**:**  Есть ли слово, содержащие все гласные алфавита?

# 

# 

# 

# 

# **Идея, метод, алгоритм** решения задачи (в формах: словесной, псевдокода, графической [блок-схема, диаграмма, рисунок, таблица] или формальные спецификации с пред- и постусловиями)

Из условия задачи следует, что мы обрабатываем слова, значит для реализации данной задачи нам нужно рассматривать множество литер до разделитей(« », «,», «\t», «\n»). В качестве неповторяющего множества литер слова(словарь), мы будем использовать побитовую масску изначально состаящую из всех 0, длина которой будет равна длине алфавита, в котором мы рассматриваем слово. Если буквы ещё нет, но на её порядковый номер в алфавите ставится 1. После того, как мы встретим разделитель, сравниваем масску нашего слова, с заготовленной маской гластных через „побитовое или“, если результат равен начальному слову, то выводим «Да» и выходим из цикла. Алгоритм одинаков для русского и английского алфавитов, но для русской версии добавляются функции, связанные с перекодировкой.

1. **Сценарий выполнения работы** [план работы, первоначальный текст программы в черновике (можно на отдельном листе) и тесты либо соображения по тестированию].

Первым делом я написал тесты

Ввод: Результат:

ЕЁОЮЭЯИУЫ Ja-ja

ЕЁОЮЭЯИУ Nein

Потом написал программу для английского алфавита, поняв, что всё работает исправно перенёс её на русский язык. Так как русский алфавит закодирован в UTF8, которая берёт >=2 байта на символ, в то время как обычный getchar() работает только с 1 байтом. Поэтому я реализовал функцию getchar() для UTF8 — utf8\_get\_num(). Также в самом начале по таблице UTF8 я нашёл коды критических букв (больших и заглавных).

const int aa = 1072, ya = 1103, A = 1040, YA = 1071;

const int ee= 1105, EE = 1025, e= 1077, E= 1045, J=1046, j=1078;

const int I = 1048, O = 1054, U = 1059, YI = 1067, JE = 1069, YU = 1070;

const int i=1080, o=1086,u=1091,yi=1099,je=1101,yu=1102;

//aa - "а", e - "е", j - "ж", ja - "я", ee - "ё", o - "о", u - "у", je - "э", yi - "ы", i - "и", yu - "ю"

Затем я реализовал цикл-прохода по вводимомму текст, считывая символ и проверял, является ли он разделитем с помощью функциии - int is\_space, если это гласная, то добавлял в побитовую маску. Когда программа доходит до разделителя она выполняет „побитовое или“ маски текущего слова с масской гласных русского алфавита, заданной изначально. Если результат равен начальному слову, то выводим «Да» и выходим из цикла.

*Допущен к выполнению работы.* **Подпись преподавателя**

1. **Замечания автора** по существу работы Пришлось покапаться с UTF8, но опыт интересный и полезный

# 

# 

# 

# Выводы

# В ходе выполнения задания я проделал большую работу: первым делом я придумал тестирующие примеры, потом я составил алгоритм выполнения программы, а затем перешёл к её реализации на СИ. В этой лабороторной я научисля работать с побитовыми масками, с обработкой кодировки UTF8. В конечном итоге у меня получилось написать исправно-работающую программу, поэтому я считаю, что справился с поставленной задачей.

# 

# 

Недочёты при выполнении задания могут быть устранены следующим образом:

Подпись студента