**Отчет по лабораторной работе № 13** по курсу “Фундаментальная информатика”

Студент группы М80-103Б-21 Березнев Никита Вадимович, № по списку 4

Контакты e-mail: nikita.berezneff@yandex.ru

Работа выполнена: «24» ноября 2021г.

Преподаватель: каф. 805 Севастьянов Виктор Сергеевич

Отчет сдан « » \_\_\_\_\_\_\_\_\_20\_\_ г., итоговая оценка \_\_\_\_\_\_\_\_

Подпись преподавателя \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Тема:** Множества. Реализации множества в Си через битовые операции.

1. **Цель работы:** В соответствии с вариантом задания составить программу проверки характеристик введенных последовательностей слов, используя созданное с помощью битовых операций множество.
2. **Задание (вариант 4):** На вход подаётся произвольный набор английских слов, разделённых пробелами, запятыми, знаками табуляции и границами строк. Букву 'y' для простоты считать согласной (гласными или согласными бывают звуки, но не буквы). Необходимо проверить выполнение следующего условия: “Есть ли слово, хотя бы одна согласная которого повторяется?” При решении задачи необходимо реализовать математическую абстракцию множества
3. **Оборудование** (студента):

Процессор *Intel Core i3-3217U @ 4x 1.80GH* с ОП *7851* Мб, НМД *512* Гб. Монитор *1366x768*

1. **Программное обеспечение (**студента**):**

Операционная система семейства: *linux*, наименование: *ubuntu*, версия *18.10 cosmic*

интерпретатор команд: *bash* версия *4.4.19*.

Система программирования -- версия --

Утилиты операционной системы:

Прикладные системы и программы --

Местонахождение и имена файлов программ и данных на домашнем компьютере --

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

- Для реализации программы необходимо создать множество согласных букв английского алфавита, используя битовые операции. Затем помощью цикла while(scanf(“%c”, #название переменной#) != EOF) реализовать считывание элементов до конца файла. После этого каждый считанный символ заносить в множество считанного символа, а это множество объединять с множеством считанных элементов слова. Затем осуществив конъюнкцию множества считанных элементов и множества согласных букв, мы получим множество, состоящее только из согласных букв, считанных из слова. Совершая объединение этого множества с множеством считанного символа, мы можем определить, встречался ли считанный символ ранее. Если конъюнкция не равна нулю, то считанный элемент присутствует в множестве ранее считанных элементов слова, являющихся согласными буквами. Поэтому такой элемент будет подходить нам, и после окончания цикла нужно будет вывести ‘Yes’. Также для корректной работы программы необходимо при переходе к новому слову обнулять множество считанных символов слова. Если после окончания цикла не нашлось ни одной согласной буквы, которая встретилась бы в слове дважды, то вывести ‘No’.

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

- С помощью битовых операций составить множество согласных букв английского алфавита. Создать пустое множество для символов считываемого слова.

- Реализовать цикл, считывающий числа из потока до конца файла.

- При каждом проходе заполнять пустое множество считанными символами. В конце прохода осуществлять конъюнкцию множества согласных букв и множества считанных символов. Таким образом множество считанных элементов будет содержать только те считанные элементы, которые являются согласными буквами английского алфавита.

- Затем в начале следующего прохода осуществить конъюнкцию считанного символа и множества ранее считанных элементов слова. Если конъюнкция не равна 0, то считанный символ уже ранее встречался в этом слове. При этом, т.к. в конце каждого прохода происходила конъюнкция множества считанных элементов и множества согласных букв, то считанный символ, ранее встречавшийся во множестве элементов слова, будет согласной буквой английского алфавита.

- Если считанный символ ранее встречался в слове, то после окончания цикла вывести ‘Yes’. В противном случае вывести ‘No’

- При переходе к новому слову необходимо обнулять множество считанных элементов слова.

- Составить ряд тестов, с помощью которых можно было бы проверить корректность работы составленного алгоритма.

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| crescent | Yes |
| victoriam venit ad fortem | No |
| Ride the lightning | Yes |
| asderty s s rt t sdgh | No |

**8. Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанныйпреподавателем).

#include <stdio.h>

#include <ctype.h>

#define Start ((1u << ('b' - 'a')) | (1u << ('c' - 'a')) | (1u << ('d' - 'a')) | (1u << ('f' - 'a')) | (1u << ('g' - 'a')) | (1u << ('h' - 'a')) | (1u << ('j' - 'a')) | (1u << ('k' - 'a')) | (1u << ('l' - 'a')) | (1u << ('m' - 'a')) | (1u << ('n' - 'a')) | (1u << ('p' - 'a')) | (1u << ('q' - 'a')) | (1u << ('r' - 'a')) | (1u << ('s' - 'a')) | (1u << ('t' - 'a')) | (1u << ('v' - 'a')) | (1u << ('w' - 'a')) | (1u << ('x' - 'a')) | (1u << ('y' - 'a')) | (1u << ('z' - 'a')))

int main(void)

{

unsigned int set\_sym;

unsigned int set\_in;

char sym;

int count;

set\_in = 0;

count = 0;

set\_sym = 0;

while (scanf("%c", &sym) != EOF) {

if (sym == ' ' || sym == '\n') {

set\_in = 0;

set\_sym = 0;

}

sym = tolower(sym);

if (sym >= 'a' && sym <= 'z') {

set\_sym = 1u << (sym - 'a');

} else {

set\_sym = 0;

}

if ((set\_sym & set\_in) != 0) {

count = 1;

}

set\_in = set\_in | set\_sym;

set\_in = set\_in & Start;

}

if (count == 1) {

printf("Yes\n");

}

if (count == 0) {

printf("No\n");

}

return 0;

}

**9. Дневник отладки** должен содержать дату и время сеансов отладки и основные события(ошибки в сценарии и программе,

нестандартные ситуации) и краткие комментарии к ним. В дневнике отладки приводятся сведения об использовании других ЭВМ, существенном участии преподавателя и других лиц в написании и отладке программы.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Лаб. | Дата | Время | Событие | Действие по исправлению | Примечание |
|  | или |  |  |  |  |  |
|  | дом. |  |  |  |  |  |
| 1 | дом | 23.11.21 | 19:54 | Программа при конъюнкции выводила те элементы, которых не было в считываемом слове | Обнуление множества считанных символов в новом слове | Весь ~~город~~ *мусор* должен быть очищен!  @~~Артас Менетил~~  @*Чекер Непреклонный* |
| 2 | дом | 24.11.21 | 12:36 | Программа считывала символы, не являющие буквами английского алфавита, из-за чего множество считанных символов было некорректным | Проверка каждого считываемого символа на принадлежность к буквам английского алфавита | “Будь осторожнее в своих ~~желаниях~~ *кодах*. А если уж ~~желание исполнилось~~ *код запустился*, прими последствия с достоинством”  ~~@Йеннифэр из Венгерберга~~  *@Чекер из Маёвника* |

1. **Замечания автора** по существу работы

Работу считаю полезной, весьма интересной и довольно приятной (За исключением того момента, когда для понимания того, что нужно использовать не ‘lu’, а ‘1u’ ушло около часа). Создавая программу на Си, в основе работы которой лежат действия с множествами, я узнал не только о способе реализации множеств в Си через битовые операции, но и научился работать с множествами.

1. **Выводы**

Работа мне понравилась, поскольку создание и взаимодействие с множествами было чем-то совершенно новым, а потому и интересным. Благодаря данной работе я освоил алгоритм создания множества, а также механизмы работы с ним, что расширило область моих знаний. Информация, полученная мной в ходе выполнения данной лабораторной работы, поможет мне в дальнейшем при составлении более сложных алгоритмов.

Подпись студента \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_