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

Фундаментальная информатика

Студент группы М8О-113Б-22 Горин Никита Андреевич, № по списку 6

Контакты www, e-mail, icq, skype exprompar@gmail.com

Работа выполнена: «14» \_\_ноября\_\_\_\_\_\_\_\_\_\_202\_2\_\_г.

Преподаватель: \_\_доц.\_каф. 806 Никулин Сергей Петрович

Входной контроль знаний с оценкой \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Отчет сдан « » \_\_\_\_\_\_\_\_\_201 \_\_ г., итоговая оценка \_\_\_\_\_

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

* 1. **Тема:** Множества \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Задание** (*вариант №7* )**:** **Оборудование** (лабораторное):

ЭВМ , процессор \_\_\_\_\_\_\_\_\_\_\_\_, имя узла сети с ОП \_\_\_\_\_\_\_\_\_\_\_\_ Мб, НМД \_\_\_\_\_\_\_ Мб. Терминал \_\_\_\_\_\_\_адрес \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_. Принтер \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i7-11400 с ОП 15360 Мб, НМД 716800 Мб. Монитор \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Другие устройства Мышь Logitech G403\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Операционная система семейства \_\_\_\_\_\_\_\_\_\_\_, наименование \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

интерпретатор команд версия

Система программирования версия

Редактор текстов версия

Утилиты операционной системы

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Прикладные системы и программы \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Местонахождение и имена файлов программ и данных\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства Linux, наименование Ubuntu версия 20.04.3 LTS

интерпретатор команд bash версия 5.0.17

Система программирования \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_версия \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Редактор текстов emacs версия 28.2

Утилиты операционной системы \_\_\_\_\_\_\_\_\_\_\_\_\_\_Стандартные утилиты Linux

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Прикладные системы и программы Стандартные 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)

Cоздам множество гласных букв. Проходя по строкам, буду находить слова и проверять их первую и последнюю букву на вхождение в это множество.

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

/\* Лабораторная работа 13

Горин Никита, М8О-113Б-22

Вариант 7 \*/

#include <string.h>

#include <stdio.h>

#include <stdbool.h>

#include <ctype.h>

typedef unsigned set\_data\_elem;

enum {

bits\_per\_char = 8,

bits\_per\_elem = sizeof(set\_data\_elem) \* bits\_per\_char,

datalen = (1 << bits\_per\_char) / bits\_per\_elem

};

typedef struct {

set\_data\_elem data[datalen];

} set;

void set\_clear(set\* s) {

memset(s->data, 0, sizeof(s->data));

}

void set\_insert(set\* s, int c) {

s->data[c / bits\_per\_elem] |= 1u << (c % bits\_per\_elem);

}

void set\_generate(set\* s, bool indicator(int)) {

int i;

set\_clear(s);

for (i = 0; i != 1 << bits\_per\_char; i++) {

if (indicator(i)) {

set\_insert(s, i);

}

}

}

void set\_erase(set\* s, int c) {

s->data[c / bits\_per\_elem] &= ~(1u << c % bits\_per\_elem);

}

bool set\_in(const set\* s, int c) {

return (s->data[c / bits\_per\_elem] & (1u << c % bits\_per\_elem)) != 0;

}

int set\_size(const set\* s) {

int i, size = 0;

for (i = 0; i != 1 << bits\_per\_char; i++) {

if (set\_in(s, i)) {

size++;

}

}

return size;

}

bool set\_equal(const set\* s1, const set\* s2) {

int i;

for (i = 0; i != datalen; i++) {

if (s1->data[i] != s2->data[i]) {

return false;

}

}

return true;

}

bool set\_includes(const set\* s1, const set\* s2) {

int i;

for (i = 0; i != datalen; i++) {

if ((s1->data[i] | s2->data[i]) != s1->data[i]) {

return false;

}

}

return true;

}

set set\_union(const set\* s1, const set\* s2) {

int i;

set result;

for (i = 0; i != datalen; i++) {

result.data[i] = s1->data[i] | s2->data[i];

}

return result;

}

set set\_intersection(const set\* s1, const set\* s2) {

int i;

set result;

for (i = 0; i != datalen; i++) {

result.data[i] = s1->data[i] & s2->data[i];

}

return result;

}

set set\_difference(const set\* s1, const set\* s2) {

int i;

set result;

for (i = 0; i != datalen; i++) {

result.data[i] = s1->data[i] & ~(s2->data[i]);

}

return result;

}

set set\_symmetric\_difference(const set\* s1, const set\* s2) {

int i;

set result;

for (i = 0; i != datalen; i++) {

result.data[i] = s1->data[i] ^ s2->data[i];

}

return result;

}

bool is\_glasn(int c) {

if (c == 'a' || c == 'A' ||

c == 'e' || c == 'E' ||

c == 'i' || c == 'I' ||

c == 'o' || c == 'O' ||

c == 'u' || c == 'U' ||

c == 'y' || c == 'Y') {

return true;

}

else {

return false;

}

}

bool is\_space(char c) {

if ((c == '\t' || c == '\n' || c == ' ' || c == ',')) {

return true;

}

else return false;

}

int main()

{

int i;

set s\_glasn;

set\_clear(&s\_glasn);

set\_generate(&s\_glasn, is\_glasn);

FILE\* f1, \* f2;

f1 = fopen("input.txt", "r");

f2 = fopen("output.txt", "w");

if (f1 == NULL) {

printf("Error");

getchar();

return 0;

}

if (f2 == NULL) {

printf("Error");

getchar();

return 0;

}

int state = 0;

//0 - символ

//1 - пробел

//остальные switch написаны по диаграмме состояний

int OK = 0;

char c = fgetc(f1);

char last = 'Δ';

bool first = true;

int z;

int cnt = 0;

while (!feof(f1)) {

if (!first) {

if (!(is\_space(c))) {

if (is\_space(last)) {

if (set\_in(&s\_glasn, c)) {

OK++;

}

}

fprintf(f2, "%c", c);

}

else {

if (!is\_space(last)) {

if (is\_glasn(last))

OK++;

if (OK == 2) {

fprintf(f2, " - подходит\n");

cnt++;

}

else {

fprintf(f2, " - не подходит\n");

}

OK = 0;

}

}

}

else {

first = false;

}

last = c;

c = fgetc(f1);

}

if (OK != 2 && !is\_space(last)) {

fprintf(f2, " - не подходит\n");

}

else if (OK == 2 && !is\_space(last)) {

fprintf(f2, " - подходит\n");

cnt++;

}

fprintf(f2, "Найденных слов, подходящих под условие: %d", cnt);

fclose(f1);

fclose(f2);

}

*Пункты 1-7 отчета составляются сторого до начала лабораторной работы.*

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

1. **Распечатка протокола** (подклеить листинг окончательного варианта программы с тестовыми примерами,подписанный

преподавателем).

user@user-Katana:~$ gcc LAB\_13.c -lm

user@user-Katana:~$ cat input.txt

What a beautiful day, my dear Anna

I love computer science, it is

my future job

Aloy,Evgeniy are my friends

user@user-Katana:~$ cat output.txt

What - не подходит

a - подходит

beautiful - не подходит

day - не подходит

my - не подходит

dear - не подходит

Anna - подходит

I - подходит

love - не подходит

computer - не подходит

science - не подходит

it - не подходит

is - не подходит

my - не подходит

future - не подходит

job - не подходит

Aloy - подходит

Evgeniy - подходит

are - подходит

my - не подходит

friends - не подходит

Найденных слов, подходящих под условие: 6

**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. **Замечания автора** по существу работы замечания отсутствуют

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Выводы**

Я научился писать программы для работы с множествами на языке Си

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Недочёты при выполнении задания могут быть устранены следующим образом: \_\_\_\_Недочетов нет

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