

# Отчет по лабораторной работе №13 по курсу \_\_\_\_\_ Фундаментальная информатика \_\_\_\_\_

Студент группы \_\_\_\_\_ М8О-102Б-21 \_\_\_\_\_, № по  
списку \_9\_ - Кажекин Денис Андреевич

Контакты www, e-mail, icq,  
skype \_\_\_\_\_ deniskazhekin@mail.ru \_\_\_\_\_

Работа выполнена:

Преподаватель: \_\_\_\_\_ Никулин Сергей Петрович \_\_\_\_\_  
каф. 806 \_\_\_\_\_

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

Отчет сдан « \_\_\_\_\_ » \_\_\_\_\_ 201 \_\_\_\_ г., итоговая  
оценка \_\_\_\_\_

Подпись преподавателя  
\_\_\_\_\_

1. **Тема:** Множества

2. **Цель работы** Составить программу проверки характеристик введенных последовательностей слов и печати развернутого ответа в соответствии с вариантом задания

3. **Задание:** Есть ли соседние слова, состоящие из одного и того же набора букв? (Вариант 9)

4. **Оборудование:**

*Оборудование ПЭВМ студента, если использовалось:*

Процессор: Apple M1, с ОП 8192 Мб, НМД 262144 Мб. Монитор: Retina 13,3; IPS 2560 x 1600.

5. **Программное обеспечение ЭВМ студента, если использовалось**

Операционная система семейства Linux, наименование Ubuntu, версия \_\_\_\_\_ 20.04.3 LTS \_\_\_\_\_ интерпретатор команд \_bash\_ версия \_\_\_\_\_ 5.0.17(1) \_\_\_\_\_

Редактор текстов GNU emacs, версия 27.2

Прикладные системы и программы: Компилятор языка C – gcc.

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

1. Создадим два множества (set1 и set2). Set2 будет отвечать за буквы, которые встречаются в каждом новом слове. Заполнять его мы будем с помощью множества set, которое просто на каждой новой итерации принимает одну единицу в разряд, соответствующий букве, считанной со слова. Set2 будет заполняться с помощью дизъюнкции с set. В set1 мы будем передавать значение set2, освобождая его (set2) для нового слова.
2. При считывании пробела будем сравнивать наши set1 и set2, в случае совпадения мы поднимаем флаг (flag = 1) и в конце цикла можем быть уверены, что ответ на вопрос задачи положительный, иначе мы в переменную set1 присваиваем set2 и обнуляем set2. Важным замечанием является условие на проверку пробелов и переносов строки (space):

Если встречается два пробела или переноса строки подряд, то set1 и set2 становятся равны нулю и флаг поднимается (set1(0) == set2(0)), поэтому поставим условие на то, что подряд мы можем переписать значение set1 = set2 и обнулить set2 только один раз, встретив пробел или перенос строки, тем самым подготовив set2 для обработки следующего слова, если оно, конечно же появится далее.

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

```
#include "stdio.h"
```

```
int is_letter(char a) {
```

```
    if((a >= 'A') && (a <= 'z')){
```

```

    return 1;
}
else{
    return 0;
}
}

int is_lowercase(char a){
    if((a >= 'a') && (a <= 'z')){
        return 1;
    }
    else{
        return 0;
    }
}

unsigned int union_set(unsigned int a, unsigned int b) {
    return a | b;
}

unsigned int equal_set(unsigned int a, unsigned int b){
    if(a == b){
        return 1;
    }
    else{
        return 0;
    }
}

unsigned int add_one_small(char a){
    return 1u << (a - 'a');
}

unsigned int add_one_big(char a){
    return 1u << (a - 'A');
}

unsigned int set_clear(unsigned int a){
    return a & 0;
}

int main(){
    FILE *file = fopen("/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt", "r");
    unsigned long long int set1, set2, set;
    int flag = 0, empty = 0, space = 0;
    char character = '\0';
    character = getc(file);
    while (character != EOF){
        if(is_letter(character)){
            space = 0;

```

```

    if(is_lowercase(character)){
        set = add_one_small(character);
        set2 = union_set(set2, set);
    }
    else{
        set = add_one_big(character);
        set2 = union_set(set2, set);
    }

    empty = 1;
}
else if(empty == 1) {
    space += 1;
    if(equal_set(set1, set2)){
        flag = 1;
    }
    if(space < 2) {
        set1 = set2;
        set2 = set_clear(set2);
    }
}
character = getc(file);
}

if((equal_set(set1, set2)) & (empty == 1)){
    flag = 1;
}
if(flag == 1){
    printf("Yes, there are some words with the same set of letters");
}
else{
    printf("No, there aren't such words with the same set of letters");
}

fclose(file);

return 0;
}

```

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

*Допущен к выполнению работы. Подпись преподавателя*

---

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

deniskazhekin@MacBook-Air-Denis Desktop % cat main.c

```
#include "stdio.h"

int is_letter(char a) {
    if((a >= 'A') && (a <= 'z')){
        return 1;
    }
    else{
        return 0;
    }
}

int is_lowercase(char a){
    if((a >= 'a') && (a <= 'z')){
        return 1;
    }
    else{
        return 0;
    }
}

int main(){
    FILE *file = fopen("/Users/deniskazhekin/untitled4/cmake-build-
debug/file.txt", "r");
    unsigned long long int set1, set2, set;
    int flag = 0, empty = 0, space = 0;
    char character = '\0';

    character = getc(file);
    while (character != EOF){
        if(is_letter(character)){
            space = 0;
            if(is_lowercase(character)){
                set = 1u << (character - 'a');
                set2 = set2 | set;
            }
            else{
                set = 1u << (character - 'A');
                set2 = set2 | set;
            }
            empty = 1;
        }
        else if(empty == 1) {
            space += 1;
            if(set1 == set2){
                flag = 1;
            }
            if(space < 2) {
                set1 = set2;
                set2 = set2 & 0;
            }
        }
        character = getc(file);
    }
}
```

```

        if((set1 == set2) & (empty == 1)){
            flag = 1;
        }
        if(flag == 1){
            printf("Yes, there are some words with the same set of
letters");
        }
        else{
            printf("No, there aren't such words with the same set of
letters");
        }
        fclose(file);
        return 0;
    }

```

```

deniskazhekin@MacBook-Air-Denis Desktop % cat >
/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt
qweqweqweqweqwe      uiouoiuioiuoiuoiuoiuoi

```

fasfasfasfasfas jlkjlkjlkjlkjlkj

```

xzczcxczcxczcxczc  m
^C

```

```

deniskazhekin@MacBook-Air-Denis Desktop % gcc main.c
deniskazhekin@MacBook-Air-Denis Desktop % ./a.out

```

**No, there aren't such words with the same set of letters%**

```

deniskazhekin@MacBook-Air-Denis Desktop % cat >
/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt

```

Let me introduce myself. My name is Mariya I am a 20-year-old student from Donetsk. I study at the university in my native town and my future profession is bookkeeping. I live with my parents and my elder sister Lena. We are a friendly family. Lena is 2 years older than me. We share our room and tell all our secrets to each **other other**.

```

^C
deniskazhekin@MacBook-Air-Denis Desktop % gcc main.c
deniskazhekin@MacBook-Air-Denis Desktop % ./a.out

```

**Yes, there are some words with the same set of letters%**

```

deniskazhekin@MacBook-Air-Denis Desktop % cat >
/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt
g

```

sdjflsldk

sldfj;sjpwep

ppoepqwoeiqp

```
^C
deniskazhekin@MacBook-Air-Denis Desktop % gcc main.c
deniskazhekin@MacBook-Air-Denis Desktop % ./a.out
```

**No, there aren't such words with the same set of letters%**

```
deniskazhekin@MacBook-Air-Denis Desktop % cat >
/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt
g
```

sdjflsldk

sldfj;sjpwep

**ppoepqwoeiqp ppoepqweiqp**

```
^C
deniskazhekin@MacBook-Air-Denis Desktop % gcc main.c
deniskazhekin@MacBook-Air-Denis Desktop % ./a.out
```

**Yes, there are some words with the same set of letters%**

```
deniskazhekin@MacBook-Air-Denis Desktop % cat >
/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt
fjdlsajflsajfl
```

eqwpeiowq

fskdflfvzzxc

asda

aaa **eqweqweqweqw** **qweqweqweqwe**

```
^C
deniskazhekin@MacBook-Air-Denis Desktop % gcc main.c
deniskazhekin@MacBook-Air-Denis Desktop % ./a.out
```

**Yes, there are some words with the same set of letters**

```
deniskazhekin@MacBook-Air-Denis Desktop % cat >
/Users/deniskazhekin/untitled4/cmake-build-debug/file.txt
uuuuuu                                lklklklklklkl
```

affafafafafaf llllllllllllll

yuyuuyuyuuyuyuuyuyuy

eqwrtqrqwerwqrq

fsdfdsfsfsdfdsfs f klllklklklklk

^C

deniskazhekin@MacBook-Air-Denis Desktop % gcc main.c

deniskazhekin@MacBook-Air-Denis Desktop % ./a.out

**No, there aren't such words with the same set of letters**

**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.

	Ла б. или до м.	Д ат а	Вре мя	Событие	Действие по исправлению	Примечание

**10. Замечания автора по существу работы**

**11. Выводы** Я научился составлять программы, написанные на Си, для работы с множествами.

Подпись студента \_\_\_\_\_