

Отчет по лабораторной работе № 1 по курсу _____

Студент группы 108 Каширин Кирилл № по списку 8

Контакты www, e-mail, icq, skype _____

Работа выполнена: «26» ноября 2010 г.

Преподаватель: _____ каф. 806 _____

Входной контроль знаний с оценкой _____

Отчет сдан « » 201 ___ г., итоговая оценка _____

Подпись преподавателя _____

1. Тема: Множества

2. Цель работы: Научиться работать с множествами в С

3. Задание (вариант № 5): Есть ли слова, содержащие гласные только 1-го рода (аоуэыи)

4. Оборудование (лабораторное):

ЭВМ _____, процессор _____, имя узла сети _____ с ОП _____ Мб, НМД _____ Мб. Терминал _____ адрес _____. Принтер _____
Другие устройства _____

Оборудование ПЭВМ студента, если использовалось:

Процессор _____ с ОП _____ Мб, НМД _____ Мб. Монитор _____

Другие устройства _____

5. 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 _____, 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 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)

С помощью структуры Bitset мы создаем множество гласных первого рода и второго рода.

Считываем в цикле каждый символ из входных данных, обрабатываем с помощью функцию

utf8unicode(изменяем кодировку) . Создаем множество, которое содержит в себе символы из входных данных.

С помощью конъюнкции множеств проверяем наличие гласных 1-го рода в этом множестве и наличие гласных 2-го рода.

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

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

Допущен к выполнению работы. Подпись преподавателя _____

8. Распечатка протокола

(подклейте листинг окончательного варианта программы с тестовыми примерами, подписаный преподавателем).

```
main.c
#include <stdio.h>
#include "library.h"
#include <stdbool.h>
int main(){
    unsigned code,det = 0,counter = 0;
    unsigned char buffer[50];
    unsigned bufferSize = 50;
    struct Bitset current;
    init(&current);
    struct Bitset bs;
    init(&bs);
    unsigned char letter[2];
    letter[0]=0;
    letter[1]=0;
    const char* vowels = "аоуәыиАОУӘЫИ";
    struct Bitset vowelset;
    init(&vowelset);
    for (unsigned i = 0; i < 24;i++){
        letter[counter] = vowels[i];
        ++counter;
        if (counter == 2){
            code = utf8ToUnicode(letter[0],letter[1]);
            addElement(&vowelset,code);
            counter = 0;
            letter[0] = 0;
            letter[1] = 0;
        }
    }
    counter=0;
    const char* removevowels = "еюяЕЮЯ";
    struct Bitset removevowelset;
    init(&removevowelset);
    for (unsigned i = 0;i<12;i++){
        letter[counter] =removevowels[i];
        ++counter;
        if (counter == 2){
            code = utf8ToUnicode(letter[0],letter[1]);
            addElement(&removevowelset,code);
            counter = 0;
            letter[0] = 0;
            letter[1] = 0;
        }
    }
    counter = 0;
    letter[0] = 0;
    letter[1] = 0;
    unsigned k=0;
    unsigned f=0;
    while (fgets(buffer, bufferSize, stdin) != NULL) {
        for (int i = 0; i < bufferSize; ++i){
            if (buffer[i]!=' ' && buffer[i]!='\n' && buffer[i]!='\0'){
                letter[counter] = buffer[i];
                ++counter;
                if (counter == 2){
                    code = utf8ToUnicode(letter[0],letter[1]);
                    addElement(&current,code);
                    addElement(&bs,code);
                    if ((code == 1025) || (code == 1105)){
                        f=-1;
                    }
                    counter = 0;
                    letter[0] = 0;
                    letter[1] = 0;
                }
            }
            if (buffer[i]==' ' || buffer[i]=='\0' || buffer[i]=='\n'){
                struct Bitset bs;
                bs=current;
                makeIntersect(&bs,vowelset);
                makeIntersect(&current,removevowelset);
                if (bs.set!=0 && current.set == 0 && f!=-1){
                    det+=1;
                }
                init(&current);
                init(&bs);
                f=0;
            }
            if (buffer[i]=='\0'){
                break;
            }
        }
    }
    if (det>0){
        printf("Есть слово, содержащее гласные только 1-го рода\n");
    } else {
        printf("Нет слов, содержащие гласные только 1-го рода\n");
    }
}

functions.c
#include <stdio.h>
#include "library.h"
#include <stdbool.h>
void init (struct Bitset* bs){
    bs->set = 0;
}
void makeIntersect (struct Bitset* bs, const struct Bitset r){
    bs->set = bs->set & r.set;
}
void addElement(struct Bitset* bs, unsigned i){
    bs->set = bs->set | (1ull << (i - 1040));
}
unsigned utf8ToUnicode (unsigned char firstByte, unsigned char secondByte){
    unsigned firstDigit = firstByte & 31;
    unsigned secondDigit = secondByte & 63;
    return (firstDigit << 6) | secondDigit;
}
void text(struct Bitset* bs, unsigned element){
    bs->set = bs->set | (1u << 6);
}
void printSet (struct Bitset* const bs)
{
const unsigned maxBitsetType = sizeof(BitsetType) * 8;
for (int i = maxBitsetType - 1; i >= 0; --i)
{
    bool isCurrentValZero = (bs->set & (1u << i)) == 0;
    if (isCurrentValZero) {
        printf ("0");
    } else {
        printf ("1");
    }
    printf("\n");
}
library.h
typedef unsigned long long BitsetType;
struct Bitset {
    BitsetType set;
};
void init (struct Bitset* bs);
void makeIntersect (struct Bitset* bs, const struct Bitset r);
void addElement (struct Bitset* bs, unsigned i);
unsigned utf8ToUnicode (unsigned char firstByte, unsigned char secondByte);
void printSet (struct Bitset* bs);
void makeComplement (struct Bitset* bs);
```

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. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание

10. **Замечания автора** по существу работы _____

11. Выводы

Я научился работать с множествами в С, совершать логические операции с ними.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____