|  |  |
| --- | --- |
|  | **Отчёт по лабораторной работе** № 13 по курсу 1 ­  студента группы M80-108Б-18 Коростелев Д. В., № по списку 12  Адреса www, e-mail, jabber, skype  Работа выполнена: “ “ 200 г.  Преподаватель: Поповкин А.В. каф.806  Входной контроль знаний с оценкой  Отчёт сдан “ “ 20 г., итоговая оценка  Подпись преподавателя |

1. **Тема**: практикум по курсам «Основы информатики» и «Алгоритмы и структуры данных»
2. **Цель работы**: составить программу для проверки характеристик введенных последовательностей слов и печати развернутого ответа.
3. **Задание** (*вариант №12*): Составить программу для поиска согласных букв, встречающихся в каждом слове заданного текста.
4. **Оборудование** (*лабораторное*):

ЭВМ Cameron, процессор Intel Core2 Duo CPU E8500 3.16ГГц, имя узла сети cameron с ОП 16029 МБ

НМД 50 ГБ. Терминал gnome адрес 172.16.80.213. Принтер лазерный принтер с технологией spooling Другие устройства

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel Core i7-7700HQ 4x 2.808ГГц, ОП 3945 МБ, НМД 48,9 ГБ. Монитор

Другие устройства

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

Операционная система семейства UNIX , наименование UBUNTU версия 16.04

Интерпретатор команд bash версия 4.3.48

Система программирования gcc версия 7.3.0

Редактор текстов nano версия 2.9.3

Утилиты операционной системы gсс, nano

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX , наименование UBUNTU версия 18.04

Интерпретатор команд bash версия 4.3.48

Система программирования gcc версия 5.4.0

Редактор текстов nano версия 2.5.3

Утилиты операционной системы gсс, nano

Прикладные системы и программы

Местонахождения и имена файлов программ и данных

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

Для решения данной задачи будем использовать множества. Так как множества не реализованы в основных библиотеках языка Си, за это дело возьмемся мы сами. Само множество будет являться беззнаковым целочисленным числом типа unsigned int. Для решения данной задачи нам потребуется два множества, то есть две переменные, назовем их set\_1 и set\_2 ,с помощью команд “<<”, “>>”, “|”,”&” пользователь может управлять битами целого числа, использовать их в качестве маски для обозначения, сохранения каких-либо данных

Далее будем действовать по такому алгоритму: сначала посимвольно считаем первое слово и запишем все его буквы во множество set\_1, записывать будем так set\_1 = set\_1 | (unit << (symbol – ‘a’)), unit – константная переменная равная 1, symbol – полученная буква, с помощью такой операции мы будем сохранять буквы в множество.

После этого остальные слова из текста будем считывать и сохранять их буквы в set\_2. Как только слово закончилось, делаем конъюнкцию множеств set\_1 и set\_2 в таком виде set\_1 = set\_1 & set\_2. После достижения конца файла в set\_1 сохраняться все буквы, использованные в каждом слове, с помощью вспомогательной переменной checker, в которой будет только 1 значащий и перемещающийся бит, будем вытаскивать буквы из set\_1 и проверять их через специальное условие, фильтрующее гласные буквы.

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

**Тесты**

|  |  |
| --- | --- |
| **Исходные данные** | **Результат** |
| abcd abc | b c |
| abcd abc ab | b |
| abcd | b c d |
| abc def | Повторяющиеся в каждом слове согласные не найдены. |
| Hello, world!!! | l |

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

Допущен к выполнению работы. Подпись преподавателя

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

|  |
| --- |
| dmitry@dmitry-VirtualBox:~/lubs/13$ gcc -std=c99 -pedantic -Wall -o 13.out 13.c  dmitry@dmitry-VirtualBox:~/lubs/13$ cat 13.c  #include<stdio.h>  #include<ctype.h>  typedef enum {false,true} bool;  const unsigned int unit = 1;  const int s\_alf = 26;  int main(void) {  FILE \*file;  file = fopen("text.txt", "r");  unsigned int set\_1 = 0;  unsigned int set\_2 = 0;  bool was\_spc = 1;  char symbol;  while (tolower(symbol = fgetc(file)) < 'a'|| tolower(symbol)> 'z') {}  if(feof(file)){  printf("Повторяющиеся в каждом слове согласные не найдены.\n");  return 0;  }  do {  set\_1 = set\_1 | unit << (symbol - 'a');  } while ((symbol = fgetc(file)) != ' ' && symbol != '\n');  while (1) {  symbol = fgetc(file);  if(feof(file))  break;  if (tolower(symbol) >= 'a' && tolower(symbol) <= 'z') {  set\_2 = set\_2 | unit << (symbol - 'a');  was\_spc = 0;  }  if ((symbol == ' ' || symbol == '\n') && was\_spc == 0) {  set\_1 = set\_1 & set\_2;  set\_2 = 0;  was\_spc = 1;  }  }  unsigned int checker = 0;  bool s\_mess = 0;  for (int i = 0; i < s\_alf; i++) {  checker = set\_1 & (1u << i);  if (checker != 0 && (i+'a') != 'a' && (i+'a')!= 'e' && (i+'a')!= 'i' && (i+'a') != 'o' && (i+'a')!= 'u' && (i+'a')!= 'y'){  if(s\_mess == 0){  printf("Найдены согласные, которые встречаются в каждом слове: ");  s\_mess = 1;  }  printf("\"%c\", ", i + 'a');  }  checker = 0;  }  if(s\_mess == 0)  printf("Повторяющиеся в каждом слове согласные не найдены ");  printf("\b\b.\n");  return 0;  }  dmitry@dmitry-VirtualBox:~/lubs/13$ nano text.txt  dmitry@dmitry-VirtualBox:~/lubs/13$ cat text.txt  abcd abc  dmitry@dmitry-VirtualBox:~/lubs/13$ ./13.out  Найдены согласные, которые встречаются в каждом слове: "b", "c".  dmitry@dmitry-VirtualBox:~/lubs/13$ nano text.txt  dmitry@dmitry-VirtualBox:~/lubs/13$ cat text.txt  abcd abc ab  dmitry@dmitry-VirtualBox:~/lubs/13$ ./13.out  Найдены согласные, которые встречаются в каждом слове: "b".  dmitry@dmitry-VirtualBox:~/lubs/13$ nano text.txt  dmitry@dmitry-VirtualBox:~/lubs/13$ cat text.txt  abcd  dmitry@dmitry-VirtualBox:~/lubs/13$ ./13.out  Найдены согласные, которые встречаются в каждом слове: "b", "c", "d".  dmitry@dmitry-VirtualBox:~/lubs/13$ nano text.txt  dmitry@dmitry-VirtualBox:~/lubs/13$ cat text.txt  abc def  dmitry@dmitry-VirtualBox:~/lubs/13$ ./13.out  Повторяющиеся в каждом слове согласные не найдены.  dmitry@dmitry-VirtualBox:~/lubs/13$ nano text.txt  dmitry@dmitry-VirtualBox:~/lubs/13$ cat text.txt  Hello, world!!!  dmitry@dmitry-VirtualBox:~/lubs/13$ ./13.out  Найдены согласные, которые встречаются в каждом слове: "l". |

1. **Дневник отладки** должен содержать дату и время сеансов отладки, и основные ошибки (ошибки в сценарии и программе, не стандартные операции) 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 |
|  |  |  |  |  |  |  |

1. Замечание автора по существу работы
2. Выводы Благодаря данной лабораторной работе студент может научиться реализовывать математические множества на язике Си, а также понять как устроены множества в других различных языках, также студент может научиться оперировать с различными множествами, с помощью специальных команд: обрабатывать, записывать, и читать элементы множества.

Недочеты, допущенные при выполнении задания, могут быть устранены следующим образом

Подпись студента