

Отчет по лабораторной работе №13

по курсу: 1 фундаментальная информатика

студента группы M80-101Б-21 Тулина Ивана, № по списку: 22

Контакты www, e-mail, icq, skype: i.tulin0107@gmail.com

Работа выполнена: «23» декабря 2021г.

Преподаватель: Титов В. К. каф. 806

Входной контроль знаний с оценкой _____

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

Подпись преподавателя _____

1.1 Тема: Множества

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

3 **Задание (вариант собственный):** Найти и вывести слова, которые ровно один раз включают каждую согласную предыдущего слова.

4 **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 i5-7300HQ с ОП 7,87 Мб, НМД 15360 Мб. Монитор: встроенный
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 UNIX, наименование Ubuntu версия 20.04.3 LTS
интерпретатор команд bash версия _____.

Система программирования _____ версия _____

Редактор текстов Emacs версия 3.22.30

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)

На вход подается текстовый файл со словами

1. Составляем множество гласных.
2. При помощи отрицания множество гласных преобразуем в множество согласных.
3. Если не встречен конец файла, формируем из полученных на вход букв слово в виде массива.
4. Составляем множество букв слова.
5. Находим множество элементов, находящихся на пересечении множества букв слова и множества согласных – множество согласных этого слова.
6. Составляем множество повторяющихся в слове букв.
7. Проверяем следующие условия:
 - Множество согласных предыдущего слова входит во множество согласных слова, проверяемого в данный момент.
 - Множество согласных предыдущего слова и множество повторяющихся букв проверяемого слова не пересекаются.
 - Множество согласных предыдущего слова непустое.
 - Множество согласных проверяемого слова непустое.

Если выполняется каждое из условий то проверяемое слово является искомым. Выводим искомое слово и предшествующее ему.

8. Записываем проверяемое слово на место предыдущего.
9. Записываем множество согласных проверяемого слова на множество согласных предыдущего.
10. Возвращаемся к пункту 3.

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

// Найти и вывести слова(найденное и предыдущее), которые ровно один раз включают каждую согласную предыдущего слова.

Программа

```
#include<stdio.h>
#include"set.h"
char Vowels[]="AEIOUYaeiouy", Word[20], OldWord[20];
int Consonant[2*N], S[2*N], SD[2*N], SC[2*N], OldSC[2*N];

int main(){int i, n, t; char c, oldc=' ';
i=getchar();
CreateSet(Vowels, Consonant); NegativeSet(Consonant); // преобразование множества гласных в согласные
while(i!=EOF){c=i;
if(!letter(oldc)&&letter(c)){Word[0]=c;n=0;}
if(letter(oldc)&&letter(c)) Word[++n]=c;
if(letter(oldc)&&!letter(c)){Word[++n]='\0';
CreateSet(Word, S); IntersSet(S,Consonant,SC); // составление множества согласных слова
DoubleSet(Word, SD); // составление множества повторяющихся букв
if(Inclusion(OldSC,SC)&&!Intersection(SD,OldSC)&&!EmptyCheck(SC)&&!EmptyCheck(OldSC))
printf("Найденное слово %s ровно один раз включает каждую согласную слова %s\n", Word, OldWord);
for(t=0;t<20;t++) OldWord[t]=Word[t];
for(t=0;t<2*N;t++) OldSC[t]=SC[t];
oldc=c; i=getchar();
}
return 0;}
```

set.h

```
#define N 26
char A='A';
char Z='Z';
char a='a';
char z='z';

int letter(int c){return a<=c&& c<=z||A<=c&& c<=Z;}

void CreateSet(char W[], int S[]) // составление массива
{int i; char c;
for(i=0; i<N; i++) S[i]=S[N+i]=0;
for(i=0; c=W[i]; i++)
if (a<=c&& c<=z) S[c-a]=1; else
if (A<=c&& c<=Z) S[N+c-A]=1;
}

void DoubleSet(char W[], int S[]) // составление массива повторяющихся букв
{int i; char c, F[2*N];
for(i=0; i<N; i++) F[i]=F[i+N]=S[i]=S[N+i]=0;
for(i=0; c=W[i]; i++)
if (a<=c&& c<=z) {if(F[c-a]!=0) S[c-a]=1; F[c-a]=1;} else
if (A<=c&& c<=Z){if(F[N+c-A]!=0) S[N+c-A]=1; F[N+c-A]=1;}
}

void NegativeSet(int S[]){int i; // поэлементное отрицание
for(i=0; i<2*N; i++) if(S[i]!=0) S[i]=0; else S[i]=1; // нужно для преобразования гласных в согласные
}

void DisplaySet(int S[]) //вывод множества
{ int i; char Alph[2*N]; //будет использоваться для тестов
Alph[0]=a; Alph[N]=A; Alph[2*N]=0
for(i=1; i<N; i++)
{Alph[i]=Alph[i-1]+1; Alph[i+N]=Alph[i+N-1]+1;}
printf("\n%s\n", Alph);
for(i=0; i<2*N; i++){printf("%1d", S[i]);}
printf("\n");
}

void IntersSet(int s1[], int s2[], int s3[]){int i; // Нахождение пересечения множеств
for(i=0; i<2*N; i++) s3[i]=s1[i]&& s2[i];
}

int Inclusion(int S1[], int S2[]){int i; // Проверка наличия элемента первого множества
for(i=0; i<2*N; i++) // входящего во второе
if(S1[i]==1&& S2[i]==0) return 0; return 1;
}

int EmptyCheck(int S[]){int i; //Проверка пустоты множества
for(i=0; i<2*N; i++) if(S[i]==1) return 0; return 1;
}

int Intersection(int S1[], int S2[]){int i; // Проверка наличия совпадающих элементов
for(i=0; i<2*N; i++) if(S1[i]&& S2[i]==1) return 1; return 0; // в первом и втором множестве
}
```

Тест 1.

Ввод:

When side of BIT BITTY BOAT hit the coast in darkness, the prolonged silence was suddenly broken.

-Had mattew watermarked this stonehouse before the flood washed away all walls?

-He did not talk about that. Now we will never know.

Вывод:

Найденное слово BOAT ровно один раз включает каждую согласную слова BITTY

Найденное слово the ровно один раз включает каждую согласную слова hit

Найденное слово darkness ровно один раз включает каждую согласную слова in

Найденное слово watermarked ровно один раз включает каждую согласную слова mattew

Найденное слово will ровно один раз включает каждую согласную слова we

Тест 2. (Пустой файл)

Ввод:

Вывод:

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).

```
yusayu@YS:~/Рабочий стол/cppProjects$ cat head
```

```
*****
*
*      Лабораторная работа №13      *
*      Множества                      *
*      Выполнил: Тулин Иван Денисович *
*      (номер по списку: 22)          *
*      Группа: М8О-101Б-21            *
*
*****
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ cat lab13.c
```

```
// Найти и вывести слова(найденное и предыдущее), которые ровно один раз включают каждую согласную предыдущего слова.
```

```
#include<stdio.h>
```

```
#include"set.h"
```

```
char Vowels[]="AEIOUYaeiouy", Word[20], OldWord[20];
```

```
int Consonant[2*N], S[2*N], SD[2*N], SC[2*N], SDC[2*N], OldSC[2*N];
```

```
int main(){int i, n, t; char c, oldc=' ';
```

```
    i=getchar();
```

```
    CreateSet(Vowels, Consonant); NegativeSet(Consonant);
```

```
    while(i!=EOF){c=i;
```

```
        if(!letter(oldc)&&letter(c)){Word[ 0 ]=c;n=0;}
```

```
        if(letter(oldc)&&letter(c)) Word[++n]=c;
```

```
        if(letter(oldc)&&!letter(c)){Word[++n]='\0';
```

```
        CreateSet(Word, S); IntersSet(S,Consonant,SC);
```

```
        DoubleSet(Word, SD);
```

```
        if(Inclusion(OldSC,SC)&&!Intersection(SD,OldSC)&&!EmptyCheck(SC)&&!EmptyCheck(OldSC))
```

```
            printf("Найденное слово %s ровно один раз включает каждую согласную слова %s\n", Word, OldWord);
```

```
        for(t=0;t<20;t++) OldWord[t]=Word[t];
```

```
        for(t=0;t<2*N;t++) OldSC[t]=SC[t];}
```

```
        oldc=c; i=getchar();
```

```
    }
```

```
    return 0;}
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ cat set.h
```

```
#define N 26
```

```
char A='A';
```

```
char Z='Z';
```

```
char a='a';
```

```
char z='z';
```

```
int letter(int c){return a<=c&&c<=z||A<=c&&c<=Z;}
```

```
void CreateSet(char W[], int S[])
```

```
{int i; char c;
```

```
for(i=0; i<N; i++) S[i]=S[N+i]=0;
```

```
for(i=0;c=W[i];i++)
```

```
if (a<=c&&c<=z) S[c-a]=1; else
```

```
if (A<=c&&c<=Z) S[N+c-A]=1;
```

```
}
```

```
void DoubleSet(char W[], int S[])
```

```
{int i; char c, F[2*N];
```

```
for(i=0; i<N; i++) F[i]=F[i+N]=S[i]=S[N+i]=0;
```

```
for(i=0;c=W[i];i++)
```

```
if (a<=c&&c<=z) {if(F[c-a]!=0) S[c-a]=1;F[c-a]=1;} else
```

```
if (A<=c&&c<=Z){if(F[N+c-A]!=0) S[N+c-A]=1; F[N+c-A]=1;}
```

```
}
```

```
void NegativeSet(int S[]){int i;
for(i=0;i<2*N;i++) if(S[i]!=0) S[i]=0; else S[i]=1;
}
```

```
void DisplaySet(int S[])
{ int i; char Alph[2*N];
Alph[0]=a; Alph[N]=A; Alph[2*N]=0;
for(i=1;i<N;i++)
{ Alph[i]=Alph[i-1]+1;Alph[i+N]=Alph[i+N-1]+1;}
printf("\n%s\n", Alph);
for(i=0; i<2*N;i++){printf("%1d", S[i]);}
printf("\n");
}
```

```
void IntersSet(int s1[],int s2[],int s3[]){int i;
for(i=0; i<2*N; i++) s3[i]=s1[i]&&S2[i];
}
```

```
int Inclusion(int S1[], int S2[]){int i;
for(i=0;i<2*N;i++)
if(S1[i]==1&&S2[i]==0) return 0; return 1;
}
```

```
int EmptyCheck(int S[]){int i;
for(i=0; i<2*N; i++) if(S[i]==1) return 0; return 1;
}
int Intersection(int S1[],int S2[]){int i;
for(i=0; i<2*N; i++) if(S1[i]&&S2[i]==1) return 1; return 0;
}
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ gcc -o lab13.out lab13.c
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ cat lab13_1.txt
```

When side of BIT BITTY BOAT hit the coast in darkness, the prolonged silence was suddenly broken.

-Had mattew watermarked this stonehouse before the flood washed away all walls?

-He did not talk about that. Now we will never know.

```
yusayu@YS:~/Рабочий стол/cppProjects$ cat lab13_2.txt
```

```
yusayu@YS:~/Рабочий стол/cppProjects$ ./lab13.out < lab13_1.txt
```

Найденное слово BOAT ровно один раз включает каждую согласную слова BITTY

Найденное слово the ровно один раз включает каждую согласную слова hit

Найденное слово darkness ровно один раз включает каждую согласную слова in

Найденное слово watermarked ровно один раз включает каждую согласную слова mattew

Найденное слово will ровно один раз включает каждую согласную слова we

```
yusayu@YS:~/Рабочий стол/cppProjects$ ./lab13.out < lab13_2.txt
```

```
yusayu@YS:~/Рабочий стол/cppProjects$
```

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

В ходе лабораторной работы я научился составлять программы на языке Си, описывающие модели
символьных множеств и операции выполняемые над этими множествами.

Недочёты при выполнении задания могут быть устранены следующим образом: _____

Подпись студента _____