

Отчет по лабораторной работе № 13 по курсу «Фундаментальная информатика»

Студент группы М8О-105Б-21 Минеева Светлана Алексеевна, № по списку 14

Контакты e-mail: svetlana.mineewa2003@yandex.ru

Работа выполнена: «5» декабря 2021 г.

Преподаватель: Вячеслав Константинович Титов каф. 805

Отчет сдан «5» декабря 2021 г., итоговая оценка _____

Подпись преподавателя _____

1. **Тема:** Множества, массивы, хедер-файлы.

2. **Цель работы:** В соответствии с заданием составить программу проверки характеристик введенных последовательностей слов и печати развернутого ответа.

3. **Задание (вариант № 14):** Найти и вывести слова, содержащие буквы 'l' и 't', но не содержащие 'o' и 'u', и оканчивающиеся на букву 'r', или 'd', или 'h'.

4. **Оборудование (лабораторное):**

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

Оборудование ПЭВМ студента, если использовалось:

Процессор 2,9 GHz 2-ядерный процессор Intel Core i5 с ОП 8 Гб, НМД 500 Гб. Монитор 13,3-дюймовый (2560 x 1600).

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 Terminal версия 2.10

интерпретатор команд bash версия 3.2.

Система программирования _____ версия _____

Редактор текстов emacs версия 27.2-2

Утилиты операционной системы cat, ls, cp, mv и другие

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)

Алгоритм решения задачи:

- 1) Функция Letter (возврат элемента, если это элемент латинского алфавита).
- 2) Функция CreateSet (создаем множество для слова):
Сначала создаем множество, все элементы которого равны нулю. После начинаем проход по слову, заменяя во множестве нули, отвечающие за элемент слова, на единицы.
- 3) Функция EmptySet (проверка множества на пустоту):
Начинаем обход по множеству, если какой-то элемент множества равен единице, то возвращаем ноль, иначе возвращаем единицу.
- 4) Функция DisplaySet (вывод множества):
В первой строке выводим сначала все прописные буквы в алфавитном порядке, а после заглавные буквы. Переходим на новую строку. На новой строке выводим само множество.
- 5) Функция KonSet (проверка на то, что первое множество является подмножеством второго множества):
Начинаем обход по двум множествам. Если какой-то элемент первого множества равен единице, а соответствующий элемент второго множества равен нулю, то возвращаем ноль, иначе возвращаем единицу.
- 6) Функция IntersSet (создание новой матрицы путём пересечения двух множеств):
Начинаем обход по двум матрицам, заполняя новую матрицу: элемент матрицы равен пересечению соответствующих элементов двух матриц ($1 \text{ и } 1 = 1$, $0 \text{ и } 0 = 0$, $1 \text{ и } 0 = 0$, $0 \text{ и } 1 = 0$).
- 7) Функция IncludeSet (проверка, что два элемента одновременно содержатся в слове):
Нужно учитывать, что буква может быть как прописной, так и заглавной. Следовательно, у нас есть два варианта первого элемента и два варианта второго элемента, в сумме в первом множестве будет четыре единицы. Делаем обход по двум матрицам, и, если соответствующие элементы равны единице, увеличиваем на единицу счетчик. Если в конце концов счетчик равен трем или четырем (значит точно все два элемента присутствуют в слове), то возвращаем единицу. Если счетчик равен единице (значит лишь один элемент присутствует в слове), то возвращаем ноль. Если же у нас счетчик равен двум, то это двоякая ситуация. Элементы располагаются так: прописная первая буква, прописная вторая буква, заглавная первая буква, заглавная вторая буква. При счетчике, равном двум, в слове будет присутствовать только один элемент, если присутствует эта буква и прописная, и заглавная. В виде множества из этих четырех элементов это будет выглядеть так: 1010 или 101. При равенстве единице соответствующих элементов при обходе двух множеств мы собирали число, начиная с тысяч. Проверяем, если это число равно 1010 или 101, то возвращаем ноль, иначе единицу.
- 8) Функция NoIncludeSet (проверка на то, что первое множество не является подмножеством другого):
Начинаем обход по двум множествам. Если какой-то элемент множества равен единице и соответствующий элемент второго множества равен единице, то возвращаем ноль, иначе возвращаем единицу.
- 9) Вводим обозначения всех нужных переменных.
- 10) Создаем пять множеств: всех гласных, согласных, гласных о и у, согласных l и t, согласных r, h и d.
- 11) Считываем слово.
- 12) Проверяем, если слово оканчивается на 'r', или 'd', или 'h', не содержит буквы 'o' и 'u' и содержит буквы 'l' и 't', то выводим слово на экран.

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

Хедер-файл (set.h):

```
#define N 26
```

```
char A='A'; char Z='Z'; char a='a'; char z='z'; char Alph[2*N];
```

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

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

```
void DisplaySet(char S[]) {  
    int i;  
    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(char S1[], char S2[], char S3[]) {  
    for(int i=0; i<2*N; i++) S3[i]=S1[i]&& S2[i]; }
```

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

```
int KonSet(char S1[], char S2[]) {  
    for(int i=0; i<2*N; i++)
```

```

    if (S1[i]==1&&S2[i]==0) return 0; return 1;}

int IncludeSet(char S1[], char S2[]) {
    int k=0, s=0, n=1000;
    for(int i=0; i<2*N; i++){
        if(S1[i]==1&&S2[i]==1){ k+=1; s+=n; n/=10;}
        else if(S1[i]==0&&S2[i]==1){ n/=10;}
        if(n==0) n=1;}
    if(k==4||k==3) return 1;
    else if(k==1) return 0;
    else {
        if (s==1010||s==101) return 0;
        else return 1;}}

int EmptySet(char S[]) {
    for(int i=0; i<2*N; i++){
        if(S[i]==1) return 0; return 1;}
}

```

Основная программа:

```

#include <stdio.h>
#include "set.h"

char Word[20], S[2*N], H[1],
Vowels[]="AEIOUYaeiouy", Sv[2*N],
Ou[]="OUou", Sou[2*N], S1[2*N],
Cons[]="BCDFGHJKLMNPQRSTVWXYZbcd fghjklmnpqrstvwxyz", Sc[2*N],
Lt[]="LTlt", Slt[2*N], S2[2*N],
R[]="DdHhRr", Sr[2*N], S3[2*N];

int main() {
    int i, n, m=0; char oldc=' ', c, k=' ';
    CreateSet(Vowels, Sv); CreateSet(Ou, Sou); CreateSet(Cons, Sc); CreateSet(Lt, Slt); CreateSet(R, Sr);
    while(i!=EOF) {
        i=getchar(); c=i;
        if(!Letter(oldc)&&Letter(c)) {n=0; Word[0]=c;}
        if(Letter(oldc)&&Letter(c)) {H[0]=c; Word[++n]=c;}
        if(Letter(oldc)&&!Letter(c)) {
            Word[++n]='\0';
            CreateSet(H, S3); CreateSet(Word, S); IntersSet(S, Sv, S1); IntersSet(S, Slt, S2);
            if(!EmptySet(S)&&KonSet(S3, Sr)){
                if(!EmptySet(S1)&&NoIncludeSet(S1, Sou)){
                    if(!EmptySet(S2)&&IncludeSet(S2, Slt)){
                        if (m==0) printf("Result:\n");
                        printf("%s", Word); if(!(++m%10)) printf("\n"); } } } }
        oldc=c; } }
}

```

Тесты:

- 1) Letter les development looter lot trebled

Вывод: Letter trebled

- 2) Trotled Deflater lot red frescoed Trelter Trelstar 789 prod\$sed more prelteh dream sun greilted prod in it team 9098 break drelted

Вывод: Deflater Trelter Trelstar prelteh greilted drelted

- 3) Bear go let&ed 89789 Fir Fair labyrinth lowered deflater lotted bolted dril%ted cloner lest vred thank you precision clatter crembl letter Greeted realter knotter joi&kj varlets crehleted repleted grow 678 brak family grah 9090 beetled mat

Вывод: labyrinth deflater clatter letter realter crehleted repleted beetled

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

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

Last login: Thu Dec 5 19:54:58 on ttys000

The default interactive shell is now zsh.

To update your account to use zsh, please run `chsh -s /bin/zsh`.

For more details, please visit <https://support.apple.com/kb/HT208050>.

MacBook-Pro-MacBook:~ macbookpro\$ cat zag.txt

```
* Минеева Светлана Алексеевна *
*      М8О-105Б-21      *
*  Лабораторная работа №13  *
*****
```

MacBook-Pro-MacBook:~ macbookpro\$ ls -l |tail -5

```
-rwxr-xr-x  1 macbookpro  staff  50496  9 дек 19:55 fl.out
-rw-r--r--  1 macbookpro  staff   967  9 дек 19:54 lab13.c
-rw-r--r--  1 macbookpro  staff  2895  7 июл  2020 pslog_20200707_123036.log
-rw-r--r--  1 macbookpro  staff  1350  9 дек 19:48 set.h
-rw-r--r--  1 macbookpro  staff    0  9 дек 19:10 zag.txt
```

MacBook-Pro-MacBook:~ macbookpro\$ cat set.h

```
#define N 26
```

```
char A='A'; char Z='Z'; char a='a'; char z='z'; char Alph[2*N];
```

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

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

```
void DisplaySet(char S[]) {
    int i;
    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(char S1[], char S2[], char S3[]) {
    for(int i=0;i<2*N;i++) S3[i]=S1[i]&&S2[i]; }
```

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

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

```
int IncludeSet(char S1[], char S2[]) {
    int k=0, s=0,n=1000;
    for(int i=0;i<2*N;i++){
        if(S1[i]==1&&S2[i]==1){ k+=1; s+=n; n/=10;}
        else if(S1[i]==0&&S2[i]==1){ n/=10;}
        if(n==0) n=1;}
    if(k==4||k==3) return 1;
```

```

else if(k==1) return 0;
else {
    if (s==1010||s==101) return 0;
    else return 1;}}

```

```

int EmptySet(char S[]) {
    for(int i=0;i<2*N;i++)
        if(S[i]==1) return 0; return 1;}

```

MacBook-Pro-MacBook:~ macbookpro\$ cat lab13.c

```

#include <stdio.h>
#include "set.h"

```

```

char Word[20], S[2*N], H[1],
Vowels[]="AEIOUYaeiouy", Sv[2*N],
Ou[]="OUou", Sou[2*N], S1[2*N],
Cons[]="BCDFGHJKLMNPQRSTVWXYZbcdfghjklmnpqrstvwxyz", Sc[2*N],
Lt[]="LTlt", Slt[2*N], S2[2*N],
R[]="DdHhRr", Sr[2*N], S3[2*N];

```

```

int main() {
    int i, n, m=0; char oldc=' ', c, k=' ';
    CreateSet(Vowels,Sv); CreateSet(Ou,Sou); CreateSet(Cons,Sc); CreateSet(Lt,SlT); CreateSet(R,Sr);
    while(i!=EOF) {
        i=getchar(); c=i;
        if(!Letter(oldc)&&Letter(c)){n=0;Word[0]=c;}
        if(Letter(oldc)&&Letter(c)) {H[0]=c;Word[++n]=c;}
        if(Letter(oldc)&&!Letter(c)) {
            Word[++n]='\0';
            CreateSet(H,S3); CreateSet(Word,S); IntersSet(S,Sv,S1); IntersSet(S,SlT,S2);
            if(!EmptySet(S)&&KonSet(S3,Sr)){
                if(!EmptySet(S1)&&NoIncludeSet(S1,Sou)){
                    if(!EmptySet(S2)&&IncludeSet(S2,SlT)){
                        if (m==0) printf("Result:\n");
                        printf("%s ",Word); if(!(++m%10)) printf("\n"); } } }
            oldc=c;}}

```

MacBook-Pro-MacBook:~ macbookpro\$ gcc lab13.c -o fl.out

MacBook-Pro-MacBook:~ macbookpro\$./fl.out

Letter les development looter lot trebled

Result:

Letter trebled

MacBook-Pro-MacBook:~ macbookpro\$./fl.out

Trotled Deflater lot red frescoed Trelter Trelstar 789 prod\$sed more prelteh dream sun greilted prod in it team 9098

break drelted

Result:

Deflater Trelter Trelstar prelteh greilted drelted

MacBook-Pro-MacBook:~ macbookpro\$./fl.out

Bear go let&ed 89789 Fir Fair labyrinth lowered deflater lotted bolted dril%ted cloner lest vred thank you precision

clatter crembl letter Greeted realter knotter joi&kj varlets crehleted repleted grow 678 brak family grah 9090 beetled

mat

Result:

labyrinth deflater clatter letter realter crehleted repleted beetled

MacBook-Pro-MacBook:~ macbookpro\$

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
1	Дом.	05.11.21	19:00	Ошибка в выполнении функции CreateSet	Ошибка исправлена; описание переменной Н изменено на char H[1]	Изначально переменная Н описывалась как char H=' ', однако функция CreateSet принимает переменные вида char S[]

10. Замечания автора по существу работы

Лабораторная работа очень полезна, она отлично развивает мышление и учит программированию на языке Си.

11. Выводы

Я в соответствии с заданием составила программу проверки характеристик введенных последовательностей слов и печати развёрнутого ответа.

Недочёты при выполнении задания могут быть устранены следующим образом: больше практиковаться в написании программ на языке Си.

Подпись студента Минеева С.А