|  |  |
| --- | --- |
| + | **Отчёт по лабораторной работе** №13  по курсу 1 фундаментальная информатика  студента группы М8О-105Б-21 Бондаревой Елены Евгеньвны, № по списку 2  Контакты www, e-mail, icq, skype : lena\_bondareva\_03@mail.ru  Работа выполнена: «28» ноября 2021г.  Преподаватель: каф.806 В.К.Титов  Входной контроль знаний с оценкой  Отчёт сдан «8» декабря 2021г., итоговая оценка  Подпись преподавателя |

1. **Тема**: «Множества, массивы, хедер-файл».
2. **Цель работы**: Составить программу проверки характеристик введенных последовательностей слов и печати развернутого ответа.
3. **Задание** (*вариант №2\_)*

Найти и вывести слово, в котором встречаются такие буквы, как «A», «B», «C», «O», «a», «b», «c», «o» , множество, состоящее из букв «A», «B», «C», «O», «a», «b», «c», «o», которые есть только в нашем слове(могут быть не все) и множество, состоящее из всех букв слова, которое надо вывести. Если не нашлось такого слова, вывести последнее слово и множество первого слова.( латинский алфавит).

1. **Оборудование** (*лабораторное*):

ЭВМ -, процессор -, имя узла сети-с ОП -МБ

НМД -ГБ. Терминал - адрес -. Принтер -.

Другие устройства -.

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Intel(R) Core(TM) i3-7020U CPU @ 2.30GHz , ОП 6 ГБ, НМД 240 ГБ. Монитор IPS 1920x1080

Другие устройства -.

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

Операционная система семейства -, наименование - версия -

Интерпретатор команд - версия -

Система программирования -версия -

Редактор текстов - версия -

Утилиты операционной системы -

Прикладные системы и программы -

Местонахождения и имена файлов программ и данных-

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX/GNU , наименование Ubuntu версия x86\_64

Интерпретатор команд bash

Редактор текстов emax

Утилиты операционной системы cat

Прикладные системы и программы VTM-diagram

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)

Необходимо найти и вывести слово, в котором встречаются такие буквы, как «A», «B», «C», «O», «a», «b», «c», «o» , множество, состоящее из букв «A», «B», «C», «O», «a», «b», «c», «o», которые есть только в нашем слове(могут быть не все) и множество, состоящее из всех букв слова, которое надо вывести. Для этого необходимо сначала создать множество, которое будет состоять из букв: «A», «B», «C», «O», «a», «b», «c», «o». Затем начинаем двигаться по словам: если попали на букву, перед которой стоит пробел, делаем вывод, что пришли к началу слова. Делаем проход по слову до тех пор, пока снова не попадем на пробел, после чего продолжаем движение и проходим дальше остальные слова. Подсчитываем при этом количество нужных букв(«A», «B», «C», «O», «a», «b», «c», «o»).

Если количество нужных букв слова равняется требуемому количеству нужных букв слова, тогда выводим слово, оно подошло под условие. Выводим это слово, множество букв, встретившихся с этом слове и принадлежащих заданному множеству Ao; a также все буквы, входящие в состав слова, удовлетворяющего условию. Если таких слов несколько, выводим первое встретившееся; а если таких слов нет, то выводим последнее слово, пустое множество и множество всех букв самого первого слова.

1. **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");

}

int EqualSet(char S1[], char S2[])

{ for(int i=0;i<2\*N;i++)

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

}

void UnionSet(char S1[], char S2[], char S3[])

{ for(int i=0;i<2\*N;i++) S3[i]=S1[i]||S2[i]; }

void IntersSet(char S1[], char S2[], char S3[])

{ for(int i=0;i<2\*N;i++) S3[i]=S1[i]&&S2[i]; }

int IncludeSet(char S1[], char S2[])

{ for(int i=0;i<2\*N;i++)

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

}

int EmptySet(char S[])

{ for(int i=0;i<2\*N;i++)

if(S[i]==1) return 0; return 1;

}

void InSet(char c, char S[])

{ if(Letter(c))

if(c<=z) S[c-a]=1; else S[c+N-A]=1;

}

**Основная программа:**

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

char Word[256], S[2\*N], WordMax[256], SMax[2\*N],

Ao[]="AOBCaobc", Sa[2\*N], S1[2\*N];

int main()

{ int i, n, m, s, max=-5; char oldc=' ', c;

CreateSet(Ao,Sa);

while(i!=EOF)

{ i=getchar(); c=i;

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

if(Letter(oldc)&&Letter(c)){ Word[++n]=c; }

if(Letter(oldc)&&!Letter(c)){

Word[++n]='\0';

CreateSet(Word,S); IntersSet(S,Sa,S1);

int k=0;

for(int s=0;s<N;s++)

{

if (S1[s]==1 || S1[N+s]==1)

k++;

}

if (k>max)

{

for (int s=0;s<256;s++){

WordMax[s]=Word[s]; }

CreateSet(WordMax,SMax);

max=k;

}

}

if (max==4){

break;

}

oldc=c;

}

printf("%s ",Word); printf("\n");

DisplaySet(S1);

DisplaySet(SMax);

}

**Test1:**

alkhcv amnbcv amcnbcaaioc anbv amnbkjhwewwrwr iu

**Test2:**

akjh amnbv aoiuoibc abckjbcbamncaca ytb aa oo cnbv

**Test3:**

kjhga akjhgJKGCVHC kjhab aaaAfocb ccbcvcbc bcaoc

**Test4:**

akjhf lghj otryij try a rrrr lqkjhg

**Test5:**

alkahjfg AABBOOCCi kljhsgjh slkgj aa poi DFGHJ

*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)

elena@elena-Aspire-A315-53G:~$ cat tit.txt

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

~ ~

~ Лабораторная работа №13 ~

~ Множества ~

~ ~

~ Бондарева Елена ~

~ M8O-105Б-21 ~

~ ~

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

elena@elena-Aspire-A315-53G:~$ 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");

}

int EqualSet(char S1[], char S2[])

{ for(int i=0;i<2\*N;i++)

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

}

void UnionSet(char S1[], char S2[], char S3[])

{ for(int i=0;i<2\*N;i++) S3[i]=S1[i]||S2[i]; }

void IntersSet(char S1[], char S2[], char S3[])

{ for(int i=0;i<2\*N;i++) S3[i]=S1[i]&&S2[i]; }

int IncludeSet(char S1[], char S2[])

{ for(int i=0;i<2\*N;i++)

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

}

int EmptySet(char S[])

{ for(int i=0;i<2\*N;i++)

if(S[i]==1) return 0; return 1;

}

void InSet(char c, char S[])

{ if(Letter(c))

if(c<=z) S[c-a]=1; else S[c+N-A]=1;

}

elena@elena-Aspire-A315-53G:~$ cat lab13.cpp

#include<stdio.h>

#include"Set.h"

char Word[256], S[2\*N], WordMax[256], SMax[2\*N],

Ao[]="AOBCaobc", Sa[2\*N], S1[2\*N];

int main()

{ int i, n, m, s, max=-5; char oldc=' ', c;

CreateSet(Ao,Sa);

while(i!=EOF)

{ i=getchar(); c=i;

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

if(Letter(oldc)&&Letter(c)){ Word[++n]=c; }

if(Letter(oldc)&&!Letter(c)){

Word[++n]='\0';

CreateSet(Word,S); IntersSet(S,Sa,S1);

int k=0;

for(int s=0;s<N;s++)

{

if (S1[s]==1 || S1[N+s]==1)

k++;

}

if (k>max)

{

for (int s=0;s<256;s++){

WordMax[s]=Word[s]; }

CreateSet(WordMax,SMax);

max=k;

}

}

if (max==4){

break;

}

oldc=c;

}

printf("%s ",Word); printf("\n");

DisplaySet(S1);

DisplaySet(SMax);

}

elena@elena-Aspire-A315-53G:~$ ./a.out <test131.cpp

amcnbcaaioc

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1110000000000010000000000000000000000000000000000000

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1110000010001110000000000000000000000000000000000000

elena@elena-Aspire-A315-53G:~$ ./a.out <test132.cpp

aoiuoibc

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1110000000000010000000000000000000000000000000000000

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1110000010000010000010000000000000000000000000000000

elena@elena-Aspire-A315-53G:~$ ./a.out <test133.cpp

aaaAfocb

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1110000000000010000000000010000000000000000000000000

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1110010000000010000000000010000000000000000000000000

elena@elena-Aspire-A315-53G:~$ ./a.out <test134.cpp

lqkjhg

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

0000000000000000000000000000000000000000000000000000

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

1000010101100000000000000000000000000000000000000000

elena@elena-Aspire-A315-53G:~$ ./a.out <test135.cpp

AABBOOCCi

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

0000000000000000000000000011100000000000100000000000

abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ

0000000010000000000000000011100000000000100000000000

**9. 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  2 | Дом  Дом | 5.12.  2021  5.12.  2021 | 14:15  14:15 | При написании условия ( if ) поставила лишнюю скобку  Поставила вместо «;» запятую. | Внимательно прописывать условие, следить за тем, чтобы не поставить лишние скобки.  Ставить правильно «;» и «,», где это необходимо и не путать их между собой. |  |

**10**. Замечание автора по существу работы

**11.** Выводы

В результате выполнения работы я составила программу, в которой нашла и вывела слово(слова), которое(которые) начинается с гласной буквы, а заканчивается согласной буквой, и при этом в слове(словах) ровно 2 любые гласные буквы. (латинский алфавит)

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

Подпись студента