Отчёт по лабораторной работе № 13

по курсу 1 Фундаментальная информатика.

Выполнил студент группы **М8О-111Б-23**: Тимофеева Ирина Александровна № по списку 20

Работа выполнена: «07» декабря 2023 г.

Преподаватель: **каф. 806 Никулин Сергей Петрович**

Входной контроль знаний с оценкой: _____

Отчет сдан «07» декабря 2023 г.

Итоговая оценка: _____

Подпись преподавателя: _____

1. **Тема:** Множества
2. **Цель работы:** Создание множеств и работа с ними
3. **Задание:** вариант №20 (Есть ли слово, содержащее одну согласную, возможно несколько раз?).
4. **Оборудование**: *Оборудование ПЭВМ студента, если использовалось:*
Процессор **Intel(R) Celeron(R) CPU 5205U 1.90GHz**, ОП 4 ГБ, НМД **250** ГБ.
Монитор: встроенный. Другие устройства не использовались.
5. **Программное обеспечение:** *Программное обеспечение ПЭВМ студента, если использовалось:*

Операционная система семейства **Linux**, наименование версия **linux astra 5.15.0-33-generic**,
интерпретатор команд **bash** версия **4.4.12**.
Система программирования .
Редактор текстов **Emacs** версия **24.5.1**
Утилиты: gcc
Прикладные системы и программы:gnu
Местонахождение и имена файлов программ и данных:

6. **Код программы:**

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

```
#include <locale.h>

//английские гласные
#define VOWELS (1u<<('a'-'a') | 1u<<('e'-'a') | 1u<<('i'-'a') | 1u<<('o'-'a') |
1u<<('u'-'a'))
//русские гласные
#define VOWELS_CYRILLIC 0b11101000000010000100000100100001

//перевод символа в дбухбайтной кодировке UTF-8
int Utf8ToChar(char* chars)
{
   unsigned char a = chars[0];
   unsigned char b = chars[1];
   int res = (a & 0x1F) << 6;
   res += b & 0x3F;
   return res;
}

//двоичное представление числаа
void DecToBin(int x)
{
   int size = sizeof(int) * 8;
   for (int i = size-1; i >= 0; i--)
   {
      printf("%d", (x & (1 << i)) ? 1 : 0);
      if (i % 8 == 0)
         printf(" ");
   }
   printf("\n");
}

//количество бит числа
int count_set_bits(unsigned int n) {
   int count = 0;
   while (n) {
      count += n & 1;
      n >>= 1;
   }
   return count;
}

//пересечение множеств
unsigned int intersection_sets(unsigned int set_1, unsigned int set_2)
{
   return set_1 & set_2;
```

```
}

//объединение множеств
unsigned int union_sets(unsigned int set_1, unsigned int set_2)
{
   return set_1 | set_2;
}

//добавление к множеству букв с учетом кириллицы
unsigned int char_to_set(unsigned int c)
{
        unsigned int res = 0;
        unsigned int symb = c;
        symb = tolower(c); //регистр не учитываем
        if (symb >= 'a' && symb <= 'z')
        {
                res = 1u << (symb - 'a');
        }

        return res;
}

//добавление к множеству русских букв
unsigned int cyr_char_to_set(unsigned int c)
{
        unsigned int res = 0;
        unsigned int symb = c;
   //проверка на русские символы кодировки 1251
        if ((symb >= 0xC0 && symb <= 0xFF) || symb == 0xA7 || symb == 0xB7)
        {
                if (symb == 0xA7 || symb == 0xB7)
                        res = 1u << 5;
                else
                        res = 1u << (symb % 0xBF % 0x20 - 1);
        }
        //проверка на русские символы кодировки UTF-8
        if ((symb >= 0x410 && c <= 0x44F) || symb == 0x401 || symb == 0x451)
        {
                if (symb == 0x401 || symb == 0x451)
                        res = 1u << 5;
                else
                        res = 1u << (symb % 0x40F % 0x20 - 1);
        }

        return res;
```

```
}

int main()
{
	unsigned char c;
	unsigned int letters_set = 0; //множество литер входного потока
	unsigned int cyrillic_letters_set = 0; //множество русских литер входного потока
	unsigned int vowels_set = 0; //множество гласных букв
	unsigned char bytes[2] = { '0', '0' };
	int kol = 0;
	int word_counter = 0;
	do
	{
	   c = getc(stdin);
		unsigned int kod = c;
		if ((kod == 208) || (kod == 209))
		{
			bytes[0] = c;
			bytes[1] = getchar();
			kod = Utf8ToChar(bytes);
		}

		//добавление считанного знака к множеству английских и русских букв отдельно
		letters_set = letters_set | char_to_set(kod);
		cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);

		bytes[0] = '0';
		bytes[1] = '0';

      kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
		if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c == '.'))
		{
		   word_counter++;

	      //пересечение множества литер входного потока и множества гласных букв
	      letters_set = intersection_sets(letters_set, ~VOWELS);
	      cyrillic_letters_set = intersection_sets(cyrillic_letters_set, ~VOWELS_CYRILLIC);

	      kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);

	      if (kol == 1)
```

```
                printf("В слове %d - одна согласная буква\n", word_counter);

            //обнуление множеств
                letters_set = 0;
                cyrillic_letters_set = 0;
                if (c == '\n') word_counter = 0;
              }
        }
        while (c != EOF);

        return 0;
}
```

7. Распечатка протокола

```
irina@Irina-VivoBook:~/Prog/Prog_C$ gcc -g3 consonants.c
irina@Irina-VivoBook:~/Prog/Prog_C$ gdb ./a.out
GNU gdb (Ubuntu 12.1-0ubuntu1~22.04) 12.1
Copyright (C) 2022 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
    <http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./a.out...
(gdb) list 1
1       #include <stdio.h>
2       #include <ctype.h>
3       #include <locale.h>
4
5       //английские гласные
6       #define VOWELS (1u<<('a'-'a') | 1u<<('e'-'a') | 1u<<('i'-'a') | 1u<<('o'-'a') |
1u<<('u'-'a'))
7       //русские гласные
8       #define VOWELS_CYRILLIC 0b11101000000010000100000100100001
9
```

```
10      //перевод символа в дбухбайтной кодировке UTF-8
(gdb)
11      int Utf8ToChar(char* chars)
12      {
13          unsigned char a = chars[0];
14          unsigned char b = chars[1];
15          int res = (a & 0x1F) << 6;
16          res += b & 0x3F;
17          return res;
18      }
19
20      //двоичное представление числа
(gdb)
21      void DecToBin(int x)
22      {
23          int size = sizeof(int) * 8;
24          for (int i = size-1; i >= 0; i--)
25          {
26              printf("%d", (x & (1 << i)) ? 1 : 0);
27              if (i % 8 == 0)
28                  printf(" ");
29          }
30          printf("\n");
(gdb)
31      }
32
33      //количество бит числа
34      int count_set_bits(unsigned int n) {
35          int count = 0;
36          while (n) {
37              count += n & 1;
38              n >>= 1;
39          }
40          return count;
(gdb)
41      }
42
43      //пересечение множеств
44      unsigned int intersection_sets(unsigned int set_1, unsigned int set_2)
45      {
46          return set_1 & set_2;
47      }
48
49      //объединение множеств
50      unsigned int union_sets(unsigned int set_1, unsigned int set_2)
```

```
(gdb)
51      {
52          return set_1 | set_2;
53      }
54
55      //добавление к множеству букв с учетом кириллицы
56      unsigned int char_to_set(unsigned int c)
57      {
58              unsigned int res = 0;
59              unsigned int symb = c;
60              symb = tolower(c); //регистр не учитываем
(gdb)
61              if (symb >= 'a' && symb <= 'z')
62              {
63                      res = 1u << (symb - 'a');
64              }
65
66              return res;
67      }
68
69      //добавление к множеству русских букв
70      unsigned int cyr_char_to_set(unsigned int c)
(gdb)
71      {
72              unsigned int res = 0;
73              unsigned int symb = c;
74          //проверка на русские символы кодировки 1251
75              if ((symb >= 0xC0 && symb <= 0xFF) || symb == 0xA7 || symb ==
0xB7)
76              {
77                      if (symb == 0xA7 || symb == 0xB7)
78                              res = 1u << 5;
79                      else
80                              res = 1u << (symb % 0xBF % 0x20 - 1);
(gdb)
81              }
82              //проверка на русские символы кодировки UTF-8
83              if ((symb >= 0x410 && c <= 0x44F) || symb == 0x401 || symb ==
0x451)
84              {
85                      if (symb == 0x401 || symb == 0x451)
86                              res = 1u << 5;
87                      else
88                              res = 1u << (symb % 0x40F % 0x20 - 1);
89              }
```

```
90
(gdb)
91              return res;
92      }
93
94      int main()
95      {
96              unsigned char c;
97              unsigned int letters_set = 0; //множество литер входного потока
98              unsigned int cyrillic_letters_set = 0; //множество русских литер
входного потока
99              unsigned int vowels_set = 0; //множество гласных букв
100             unsigned char bytes[2] = { '0', '0' };
(gdb)
101             int kol = 0;
102             int word_counter = 0;
103             do
104             {
105                c = getc(stdin);
106                     unsigned int kod = c;
107                     if ((kod == 208) || (kod == 209))
108                     {
109                             bytes[0] = c;
110                             bytes[1] = getchar();
(gdb)
111                             kod = Utf8ToChar(bytes);
112                     }
113
114                     //добавление считанного знака к множеству английских и
русских букв отдельно
115                     letters_set = letters_set | char_to_set(kod);
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
117
118                     bytes[0] = '0';
119                     bytes[1] = '0';
120
(gdb)
121          kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
123                     {
124                        word_counter++;
125
126                   //пересечение множества литер входного потока и множества
гласных букв
```

```
127             letters_set = intersection_sets(letters_set, ~VOWELS);
128             cyrillic_letters_set = intersection_sets(cyrillic_letters_set,
~VOWELS_CYRILLIC);
129
130             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)
131
132             if (kol == 1)
133                     printf("В слове %d - одна согласная буква\n", word_counter);
134
135                 //обнуление множеств
136                     letters_set = 0;
137                     cyrillic_letters_set = 0;
138                     if (c == '\n') word_counter = 0;
139                   }
140             }
(gdb)
141             while (c != EOF);
142
143             return 0;
144     }
(gdb)
Line number 145 out of range; consonants.c has 144 lines.
(gdb) run
Starting program: /home/irina/Prog/Prog_C/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
man run fast but drive slow
if  you wanna go ,   do it
В слове 1 - одна согласная буква
В слове 2 - одна согласная буква
В слове 4 - одна согласная буква
В слове 5 - одна согласная буква
В слове 6 - одна согласная буква
ехал поп на повозке, и              давай   ему   кричать - бубубу
В слове 2 - одна согласная буква
В слове 3 - одна согласная буква
В слове 7 - одна согласная буква
В слове 9 - одна согласная буква
пиши  ему   или не хочешь, но знай
В слове 2 - одна согласная буква
В слове 3 - одна согласная буква
В слове 4 - одна согласная буква
В слове 6 - одна согласная буква
^C
```

```
Program received signal SIGINT, Interrupt.
0x00007ffff7d149d2 in __GI___libc_read (fd=0, buf=0x5555555592a0,
nbytes=1024) at ../sysdeps/unix/sysv/linux/read.c:26
26      ../sysdeps/unix/sysv/linux/read.c: Нет такого файла или каталога.
(gdb) b 121
No line 121 in the current file.
Make breakpoint pending on future shared library load? (y or [n]) y
Breakpoint 1 (121) pending.
(gdb) info break
Num     Type           Disp Enb Address    What
1       breakpoint     keep y   <PENDING>  121
(gdb) run
The program being debugged has been started already.
Start it from the beginning? (y or n) y
Starting program: /home/irina/Prog/Prog_C/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
irina go on

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb) watch letters_set
Hardware watchpoint 2: letters_set
(gdb) watch kol
Hardware watchpoint 3: kol
(gdb) n

Hardware watchpoint 3: kol

Old value = 0
New value = 1
main () at consonants.c:122
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105                 c = getc(stdin);
(gdb)
106                     unsigned int kod = c;
(gdb)
107                     if ((kod == 208) || (kod == 209))
(gdb)
115                     letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set
```

```
Old value = 256
New value = 131328
main () at consonants.c:116
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 1
New value = 2
main () at consonants.c:122
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105                 c = getc(stdin);
(gdb)
106                     unsigned int kod = c;
(gdb)
107                     if ((kod == 208) || (kod == 209))
(gdb)
115                     letters_set = letters_set | char_to_set(kod);
(gdb)
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105                 c = getc(stdin);
```

```
(gdb)
106                 unsigned int kod = c;
(gdb)
107                 if ((kod == 208) || (kod == 209))
(gdb)
115                 letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 131328
New value = 139520
main () at consonants.c:116
116                 cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                 bytes[0] = '0';
(gdb)
119                 bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121         kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 2
New value = 3
main () at consonants.c:122
122                 if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105             c = getc(stdin);
(gdb)
106                 unsigned int kod = c;
(gdb)
107                 if ((kod == 208) || (kod == 209))
(gdb)
115                 letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 139520
New value = 139521
```

```
main () at consonants.c:116
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 3
New value = 4
main () at consonants.c:122
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105                 c = getc(stdin);
(gdb)
106                     unsigned int kod = c;
(gdb)
107                     if ((kod == 208) || (kod == 209))
(gdb)
115                     letters_set = letters_set | char_to_set(kod);
(gdb)
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
124                         word_counter++;
(gdb)
127                 letters_set = intersection_sets(letters_set, ~VOWELS);
(gdb)
```

```
Hardware watchpoint 2: letters_set

Old value = 139521
New value = 139264
main () at consonants.c:128
128             cyrillic_letters_set = intersection_sets(cyrillic_letters_set,
~VOWELS_CYRILLIC);
(gdb)
130             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 4
New value = 2
main () at consonants.c:132
132             if (kol == 1)
(gdb)
136                     letters_set = 0;
(gdb)

Hardware watchpoint 2: letters_set

Old value = 139264
New value = 0
main () at consonants.c:137
137                     cyrillic_letters_set = 0;
(gdb)
138                     if (c == '\n') word_counter = 0;
(gdb)
105               c = getc(stdin);
(gdb)
106                   unsigned int kod = c;
(gdb)
107                   if ((kod == 208) || (kod == 209))
(gdb)
115                   letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 0
New value = 64
main () at consonants.c:116
```

```
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 2
New value = 1
main () at consonants.c:122
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105                 c = getc(stdin);
(gdb)
106                     unsigned int kod = c;
(gdb)
107                     if ((kod == 208) || (kod == 209))
(gdb)
115                     letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 64
New value = 16448
main () at consonants.c:116
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol
```

```
Old value = 1
New value = 2
main () at consonants.c:122
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105                 c = getc(stdin);
(gdb)
106                     unsigned int kod = c;
(gdb)
107                     if ((kod == 208) || (kod == 209))
(gdb)
115                     letters_set = letters_set | char_to_set(kod);
(gdb)
116                     cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
124                         word_counter++;
(gdb)
127             letters_set = intersection_sets(letters_set, ~VOWELS);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 16448
New value = 64
main () at consonants.c:128
128             cyrillic_letters_set = intersection_sets(cyrillic_letters_set,
~VOWELS_CYRILLIC);
(gdb)
130             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol
```

```
Old value = 2
New value = 1
main () at consonants.c:132
132             if (kol == 1)
(gdb)
133                     printf("В слове %d - одна согласная буква\n", word_counter);
(gdb)
В слове 2 - одна согласная буква
136                     letters_set = 0;
(gdb)

Hardware watchpoint 2: letters_set

Old value = 64
New value = 0
main () at consonants.c:137
137                     cyrillic_letters_set = 0;
(gdb)
138                     if (c == '\n') word_counter = 0;
(gdb)
105             c = getc(stdin);
(gdb)
106                 unsigned int kod = c;
(gdb)
107                 if ((kod == 208) || (kod == 209))
(gdb)
115                 letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 0
New value = 16384
main () at consonants.c:116
116                 cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                 bytes[0] = '0';
(gdb)
119                 bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121         kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)
```

```
122                 if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105             c = getc(stdin);
(gdb)
106                 unsigned int kod = c;
(gdb)
107                 if ((kod == 208) || (kod == 209))
(gdb)
115                 letters_set = letters_set | char_to_set(kod);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 16384
New value = 24576
main () at consonants.c:116
116                 cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
(gdb)
118                 bytes[0] = '0';
(gdb)
119                 bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121         kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 1
New value = 2
main () at consonants.c:122
122                 if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
105             c = getc(stdin);
(gdb)
106                 unsigned int kod = c;
(gdb)
107                 if ((kod == 208) || (kod == 209))
(gdb)
115                 letters_set = letters_set | char_to_set(kod);
(gdb)
116                 cyrillic_letters_set = cyrillic_letters_set | cyr_char_to_set(kod);
```

```
(gdb)
118                     bytes[0] = '0';
(gdb)
119                     bytes[1] = '0';
(gdb)

Breakpoint 1, main () at consonants.c:121
121             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)
122                     if (kol > 0 && (c == ' ' || c == '\n' || c == '\t' || c == ',' || c == ';' || c
== '.'))
(gdb)
124                         word_counter++;
(gdb)
127             letters_set = intersection_sets(letters_set, ~VOWELS);
(gdb)

Hardware watchpoint 2: letters_set

Old value = 24576
New value = 8192
main () at consonants.c:128
128             cyrillic_letters_set = intersection_sets(cyrillic_letters_set,
~VOWELS_CYRILLIC);
(gdb)
130             kol = count_set_bits(letters_set) + count_set_bits(cyrillic_letters_set);
(gdb)

Hardware watchpoint 3: kol

Old value = 2
New value = 1
main () at consonants.c:132
132             if (kol == 1)
(gdb)
133                     printf("В слове %d - одна согласная буква\n", word_counter);
(gdb)
В слове 3 - одна согласная буква
136                         letters_set = 0;
(gdb)
```

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 исправлени	Примечание

10. **Выводы:** В ходе выполнения данной лабораторной работы я научилась работать со множествами на языке Си.

Подпись студента _____