



## Отчёт по лабораторной работе № IX

по курсу: \_\_\_\_\_ ЯМП \_\_\_\_\_

студента группы: Чурилов С. Э. М8О-103Б-20, № по списку: 29

Адреса www, e-mail, jabber, skype churilov.ser1204@gmail.com

Работа выполнена: “ \_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_ г.

Преподаватель: \_\_\_\_\_ каф. 806 В. К. Титов

Входной контроль знаний с оценкой \_\_\_\_\_

Отчёт сдан “ \_\_\_\_ ” \_\_\_\_\_ 20 \_\_\_\_ г., итоговая оценка \_\_\_\_\_

Подпись преподавателя \_\_\_\_\_

1 **Тема:** Сортировка и поиск

2 **Цель работы:** Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице

3 **Задание (вариант № 9/8):** Простое двухпоточное слияние/ комбинированный (строка + целое) с длиной в 32 и с совместным хранением

4 **Оборудование (лабораторное):**  
ЭВМ MSI GT70 0ND 447-RU, процессор Intel Celeron 2.4 GhZ, имя узла сети cameron с ОП 15 ГБ  
НМД \_\_\_\_\_ ГБ. Терминал \_\_\_\_\_ адрес \_\_\_\_\_, Принтер \_\_\_\_\_  
Другие устройства \_\_\_\_\_

*Оборудование ПЭВМ студента, если использовалось:*

Процессор Ryzen 3 3200U @ 4x 2.6GHz, ОП \_\_\_\_\_ 8192 \_\_\_\_\_ МБ, НМД \_\_\_\_\_ ГБ. Монитор: встроенный  
Другие устройства \_\_\_\_\_

5 **Программное обеспечение (лабораторное):**  
Операционная система семейства UNIX, наименование: Ubuntu версия 18.04.5 LTS  
Интерпретатор команд: bash версия \_\_\_\_\_  
Система программирования: C версия \_\_\_\_\_  
Редактор текстов: Emax версия \_\_\_\_\_  
Утилиты операционной системы: g++

Прикладные системы и программы: \_\_\_\_\_

Местонахождения и имена файлов программ и данных: \_\_\_\_\_

*Программное обеспечение ЭВМ студента, если использовалось:*

Операционная система семейства UNIX, наименование Ubuntu версия 20.04 LTS  
Интерпретатор команд: bash версия \_\_\_\_\_  
Система программирования: C версия \_\_\_\_\_  
Редактор текстов: Emax версия \_\_\_\_\_  
Утилиты операционной системы: g++

Прикладные системы и программы: \_\_\_\_\_

Местонахождения и имена файлов программ и данных: \_\_\_\_\_

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

Принцип работы программы:

1. Создается бесконечный цикл, внутри которого запрашиваются команды от пользователя и параметры к командам
2. Далее каждая команда обрабатывается своей функцией

Функции:

1. `print` – необходима для вывода стихотворений
2. `Sort0`, `Sort1` – необходимы для сортировки
3. `Search` – необходима для бинарного поиска в стихотворении по ключу
4. `Reverse` – переворачивает массив, меняя первый и последние э-ты местами

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

План работы:

1. Придумать способ тестирования
2. Создать ядро программы
3. Создать программу
4. Провести рефакторинг
5. Отладить основную программу

Для тестирования программы использована функции перемешивания стихотворения:

Reverse() – переворачивает стихотворение.

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

```
leninware@leninware-VirtualBox:~$ cat myinfo.txt
```

```
*****
```

```
ФИО:Чурилов Сергей Эдуардович
```

```
Группа:М8О-103Б-20
```

```
Е-mail: churilov.ser1204@gmail.com
```

```
ЛР: IX
```

```
Номер по списку: 29
```

```
*****
```

```
leninware@leninware-VirtualBox:~$ whoami
```

```
leninware
```

```
leninware@leninware-VirtualBox:~$ pwd
```

```
/home/leninware
```

```
leninware@leninware-VirtualBox:~$ cd LRIX
```

```
leninware@leninware-VirtualBox:~/LRIX$ cat kp9.c
```

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

```
#include <stdlib.h>
```

```
#include <string.h>
```

```
typedef struct {
```

```
    char str[5];
```

```
    int c;
```

```
} Key;
```

```
typedef struct {
```

```
    Key key;
```

```
    char string[70];
```

```
} Item;
```

```
#define SIZE 20
```

```
int lt (Item* l, Item* r) { // Проверка на предшествование
```

```
    if (strcmp(l->key.str,r->key.str) < 0) return 1;
```

```
    if (strcmp(l->key.str,r->key.str) > 0) return 0;
```

```
    else if (l->key.c >= r->key.c) return 0;
```

```
    else return 1;
```

```
}
```

```
int Search (Item* a, int N, Item* s) { // Бинарный поиск
```

```
    int L = 0, R = N, m;
```

```
    while (L < R) {
```

```
        m = (L + R) / 2;
```

```
        if (lt(&a[m],s)) L = m + 1;
```

```
        else R = m;
```

```
    }
```

```
    if (R == N) return -1;
```

```
    else if (strcmp(a[L].key.str,s->key.str) == 0 && a[L].key.c == s->key.c) return L;
```

```
    else return -1;
```

```
}
```

```
void print (Item* a, int N) {
```

```
    printf("\n");
```

```
    for (int i = 0; i < N; i++)
```

```
        printf(" %4s | %2d | %s \n",a[i].key.str, a[i].key.c, a[i].string);
```

```
    printf("\n");
```

```
}
```

```
void Sort1(Item* a, int N) {
```

```
    int mid = N / 2;
```

```
    if (N % 2 == 1)
```

```
        mid++;
```

```
int h = 1;
```

```
Item *c = (Item*)malloc(N * sizeof(Item));
```

```
int step;
```

```
while (h < N)
```

```
{
```

```
    step = h;
```

```
    int i = 0;
```

```
    int j = mid;
```

```
    int k = 0;
```

```
    while (step <= mid)
```

```
    {
```

```
        while ((i < step) && (j < N) && (j < (mid + step)))
```

```
        {
```

```
            if (lt(&a[i], &a[j]))
```

```
            {
```

```
                c[k] = a[i];
```

```
                i++; k++;
```

```
            }
```

```
            else {
```

```
                c[k] = a[j];
```

```
                j++; k++;
```

```
            }
```

```
        }
```

```
        while (i < step)
```

```
        {
```

```
            c[k] = a[i];
```

```
            i++; k++;
```

```
        }
```

```
        while ((j < (mid + step)) && (j < N))
```

```
        {
```

```
            c[k] = a[j];
```

```
            j++; k++;
```

```
        }
```

```
        step = step + h;
```

```
    }
```

```
    h = h * 2;
```

```
    for (i = 0; i < N; i++)
```

```
        a[i] = c[i];
```

```
    }
```

```
}
```

```
void Sort0 (Item* a, int N) {
```

```
    print(a,N);
```

```
    Sort1(a,N);
```

```
    print(a,N);
```

```
}
```

```
void Reverse (Item* a, int N) {
```

```
    Item tmp;
```

```
    int j = N - 1;
```

```
    for (int i = 0; i < N/2; i++) {
```

```
        tmp = a[i];
```

```
        a[i] = a[j];
```

```
        a[j] = tmp;
```

```
        j--;
```

```
    }
```

```
}
```

```
int main() {
```

```
    int menu, sort = 0, size = 0, i;
```

```
    Item input;
```

```
    char filename[30];
```

```
    Item a[SIZE];
```

```

do {
    printf("1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit\n");
    scanf("%d",&menu);
    switch (menu) {
        case 1:{
            printf("Введите название файла: "); scanf("%s",filename); FILE* in = fopen(filename,"r"); if
(in == NULL) {printf("Файл пуст\n"); return 0;} i = 0;
            }
            while ( ( !feof(in) ) ) {
                fscanf(in,"%4s%d%c",a[i].key.str,&a[i].key.c);
                fgets(a[i].string,100,in);
                a[i].string[strlen(a[i].string) - 1] = '\0';
                i++;
            }
            fclose(in);
            sort = 0;
            size = i - 1;
            break;
        case 2:{
            if (size == 0) printf("Таблица пуста\n"); else print(a,size); break;
        }
        case 3:{
            if (size != 0) Sort0(a,size); sort = 1; break;
        }
        case 4:{
            if (sort == 0) printf("Вначале отсортируйте\n");
            else {
                printf("Введите ключ: 4 символа + число\n");
                scanf("%4s%d",input.key.str,&input.key.c); menu = Search(a,size,&input);
            }
            if (menu == -1){
                printf("\nНе найдено\n\n");
                else printf("\nНайдено:\n| %4s | %d | %s
\n\n",a[menu].key.str,a[menu].key.c,a[menu].string); };
            break;
        }

        case 5:{
            if (size == 0)
                printf("Таблица пуста\n");
            else { Reverse(a,size); sort = 0; };
            break;
        }
        case 6:{
            return 0;
            break;
        }
    }
} while (1);
return 0;
}

```

leninware@leninware-VirtualBox:~/LRIX\$ gcc kp9.c

leninware@leninware-VirtualBox:~/LRIX\$ ./a.out

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit

1

Введите название файла: in.txt

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit

1

Введите название файла: in.txt

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit

2

avcx | 12 | Враг напал на мирную страну.  
xnmk | 78 | Белой ночью, самой белой ночью  
oplf | 96 | Начал эту чёрную войну!  
qwer | 60 | Только хочет он или не хочет,  
aasx | 31 | А своё получит от войны:  
bvtu | 11 | Скоро даже дни, не только ночи,  
qwer | 44 | Станут, станут для него черны!

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
3

sdfg | 56 | Против нас полки сосредоточив,  
avcx | 12 | Враг напал на мирную страну.  
xnmk | 78 | Белой ночью, самой белой ночью  
oplf | 96 | Начал эту чёрную войну!  
qwer | 60 | Только хочет он или не хочет,  
aasx | 31 | А своё получит от войны:  
bvtu | 11 | Скоро даже дни, не только ночи,  
qwer | 44 | Станут, станут для него черны!

aasx | 31 | А своё получит от войны:  
avcx | 12 | Враг напал на мирную страну.  
bvtu | 11 | Скоро даже дни, не только ночи,  
oplf | 96 | Начал эту чёрную войну!  
qwer | 44 | Станут, станут для него черны!  
qwer | 60 | Только хочет он или не хочет,  
sdfg | 56 | Против нас полки сосредоточив,  
xnmk | 78 | Белой ночью, самой белой ночью

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
3

aasx | 31 | А своё получит от войны:  
avcx | 12 | Враг напал на мирную страну.  
bvtu | 11 | Скоро даже дни, не только ночи,  
oplf | 96 | Начал эту чёрную войну!  
qwer | 44 | Станут, станут для него черны!  
qwer | 60 | Только хочет он или не хочет,  
sdfg | 56 | Против нас полки сосредоточив,  
xnmk | 78 | Белой ночью, самой белой ночью

aasx | 31 | А своё получит от войны:  
avcx | 12 | Враг напал на мирную страну.  
bvtu | 11 | Скоро даже дни, не только ночи,  
oplf | 96 | Начал эту чёрную войну!  
qwer | 44 | Станут, станут для него черны!  
qwer | 60 | Только хочет он или не хочет,  
sdfg | 56 | Против нас полки сосредоточив,  
xnmk | 78 | Белой ночью, самой белой ночью

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
5

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
2

xnmk | 78 | Белой ночью, самой белой ночью  
sdfg | 56 | Против нас полки сосредоточив,  
qwer | 60 | Только хочет он или не хочет,  
qwer | 44 | Станут, станут для него черны!  
oplf | 96 | Начал эту чёрную войну!  
bvtu | 11 | Скоро даже дни, не только ночи,

avcx | 12 | Враг напал на мирную страну.  
aasx | 31 | А своё получит от войны:

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
3

xnmk | 78 | Белой ночью, самой белой ночью  
sdfg | 56 | Против нас полки сосредоточив,  
qwer | 60 | Только хочет он или не хочет,  
qwer | 44 | Станут, станут для него черны!  
oplf | 96 | Начал эту чёрную войну!  
bvtu | 11 | Скоро даже дни, не только ночи,  
avcx | 12 | Враг напал на мирную страну.  
aasx | 31 | А своё получит от войны:

aasx | 31 | А своё получит от войны:  
avcx | 12 | Враг напал на мирную страну.  
bvtu | 11 | Скоро даже дни, не только ночи,  
oplf | 96 | Начал эту чёрную войну!  
qwer | 44 | Станут, станут для него черны!  
qwer | 60 | Только хочет он или не хочет,  
sdfg | 56 | Против нас полки сосредоточив,  
xnmk | 78 | Белой ночью, самой белой ночью

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
4

Введите ключ: 4 символа + число  
oplf 96

Найдено:  
| oplf | 96 | Начал эту чёрную войну! |

1.Add 2.Print 3.Sort 4.Find 5.Reverse 6.Exit  
6

leninware@leninware-VirtualBox:~/LRIX\$



- 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	дом	17.05.2021	10:00	бинарный поиск не находит первый элемент	необходимо нижнюю границу сделать -1	

- 10 Замечание автора по существу работы \_\_\_\_\_

- 11 Выводы Благодаря выполнению данной лабораторной работы я узнал новые алгоритмы и научился их реализовывать

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

Подпись студента \_\_\_\_\_