

**Московский Авиационный Институт
(Национальный Исследовательский Университет)**

**Факультет информационных технологий и прикладной
математики**

**Кафедра вычислительной математики и
программирования**

**Курсовой проект IX
по курсу «Практикум на ЭВМ»
II семестр**

«Сортировка и поиск»

Студент	Сыроежкин Кирилл Геннадьевич
Группа	М8О-104Б-18
Руководитель	Доцент кафедры 806 Никулин С.П
Оценка	
Дата	

Москва 2019.

1.Постановка задачи.

Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице. Программа должна вводить значения элементов неупорядоченной таблицы и проверять работу процедуры сортировки в трех случаях:

элементы таблицы с самого начала упорядочены;

элементы таблицы расставлены в обратном порядке;

элементы таблицы не упорядочены.

В последнем случае можно использовать встроенные процедуры генерации псевдослучайных чисел.

Для каждого вызова процедуры сортировки необходимо печатать исходное состояние таблицы и результаты сортировки. После выполнения сортировки программа должна вводить ключи и для каждого из них выполнять поиск в упорядоченной таблице с помощью процедуры двоичного поиска и печатать найденные элементы, если они присутствуют в таблице.

Вариант

$N=5$

Вариант задания определяется двумя числами: (1) - номер метода сортировки = $((N - 1) \% 11) + 1$, (2) – номер структуры таблицы = $((N + 6) \% 9) + 1$, где N - номер студента по списку в группе.

Метод сортировки (в терминах Н.Вирта):

Метод Шелла

Тип ключа

Комбинированный(int+char)

Длина ключа

5 байт

Хранение данных и ключей

Отдельно

Число элементов таблицы

2.Сведение о сортировке и поиске

Сортировка Шелла

Идея метода заключается в сравнение разделенных на группы элементов последовательности, находящихся друг от друга на некотором расстоянии. Изначально это расстояние равно d или $N/2$, где N — общее число элементов. На первом шаге каждая группа включает в себя два элемента расположенных друг от друга на расстоянии $N/2$; они сравниваются между собой, и, в случае необходимости, меняются местами. На последующих шагах также происходят проверка и обмен, но расстояние d сокращается на $d/2$, и количество групп, соответственно, уменьшается. Постепенно расстояние между элементами уменьшается, и на $d=1$ проход по массиву происходит в последний раз.

Двоичный поиск

Идея двоичного поиска состоит в следующем: искомый элемент сравнивается со средним элементом последовательности. Если значение аргумента поиска меньше среднего элемента, то исходной последовательностью считается массив элементов от 1 до $N/2$, если значение аргумента поиска больше среднего элемента, то исходной считаем последовательность от $N/2$ до N . Далее, для новой последовательности этот процесс повторяется. Продолжаем деление пополам интервала поиска до тех пор, пока не будет найден искомый элемент или длина интервала поиска не станет равна единице.

3. Общий метод решения

Программа считывает заданную таблицу по строкам и записывает ключи в поле ключей структуры таблицы, а стихотворение в структуру стихотворений. Далее с полученные данные преобразовываются в зависимости от желания пользователя (сортировка, поиск и т.д).

3. Общие сведения о программе

Имена файлов:

c9.c - программа;

F1.txt - тестовый файл 1

F2.txt - тестовый файл 2

Программное и аппаратное обеспечения для запуска данной программы на ПК не ограничено в выборе. Операционная система семейства Unix - Ubuntu. Язык программирования Си. Строк в программе prog.c: 97, dump.c: 28. В заголовочном файле pass.h:15.

4. Описание логической структуры программы

Структуры

t_intchar_key хранит 2 ключа char и int

t_input хранит стихотворение

t_table хранит **t_intchar_key** (комбинированный ключ) и индекс стихотворения

Функции

enter - считывает из файла таблицу и перемещает её в структуры

shell_sort - сортировка шелла

binarysearch - бинарный поиск

reverse_table - реверс таблицы

random_keys - перемешать ключи

print_table - печать таблицы

5. Входные данные

Программа получает название файла из которого считывать информацию

6. Выходные данные

Результат работы программы может меняться в зависимости от выбранного пунктов меню:

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

7. Программа на СИ

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define COMPARE(a, b) ((a > b) - (a < b))
typedef struct
{
    int keyint;
    char keychar;
}t_intchar_key;

typedef struct
{
    char verse[100];
} t_input;

typedef struct
{
    t_intchar_key key;
    int index;
} t_table;
void destroy_table(t_table *table)
{
    for (int i = 0; i < 12; i++)
    {
        table[i].key.keychar=0;
        table[i].key.keyint=0;
        table[i].index=0;
    }
}
```

```

void destroy_input(t_input *input)
{
    for (int i = 0; i < 12; i++)
        input[i].verse[0]=0;
}

void enter(FILE * f, t_table *table, t_input *input, int
*count_lines)
{
    fseek(f, 0L, SEEK_SET);
    while (fscanf(f, "%d %c %s",
&table[*count_lines].key.keyint,
&table[*count_lines].key.keychar,
input[*count_lines].verse) != EOF)
    {
        table[*count_lines].index = *count_lines;
        ++(*count_lines);
    }
}

int shell_sort(int n, t_table *table)
{
    int i, j, step;
    int tmp, tmp2;
    char tmp1;
    for (step = n / 2; step > 0; step /= 2)
        for (i = step; i < n; i++)
        {
            tmp = table[i].key.keyint;
            tmp1 = table[i].key.keychar;
            tmp2= table[i].index;
            for (j = i; j >= step; j -= step)
            {
                if (tmp < table[j - step].key.keyint)
                    table[j] = table[j - step];
            }
        }
    }
}

```

```

        else if (tmp == table[j -
step].key.keyint && tmp1 < table[j -
step].key.keychar)
            table[j] = table[j - step];
        else
            break;
    }
    table[j].key.keyint = tmp;
    table[j].key.keychar = tmp1;
    table[j].index=tmp2;
}
return 1;
}

int compare_keys(const t_intchar_key *key1, const
t_intchar_key *key2)
{
    int cmp;
    if ((cmp = COMPARE(key1->keyint, key2->keyint)) != 0)
        return cmp;
    if ((cmp = COMPARE(key1->keychar, key2->keychar)) != 0)
        return cmp;
    return 0;
}

int binarysearch(int a, char c, const t_table *table, int
n)
{
    int low = 0, high = n - 1;
    const t_intchar_key key = { a, c };
    while (low <= high)
    {
        int middle = (low + high) / 2;
        int cmp = compare_keys(&key, &table[middle].key);
        if (cmp < 0)

```

```

        high = middle - 1;
    else if (cmp > 0)
        low = middle + 1;
    else
        return middle;
}
return -1;
}

void swap_table(t_table *x, t_table *y)
{
    t_table z = *x;
    *x = *y;
    *y = z;
}

void reverse_table(t_table *table, int count_lines)
{
    for (int i = 0; i < count_lines / 2; ++i)
        swap_table(&table[i], &table[count_lines - i - 1]);
}

void random_keys(t_table *table, int count_lines)
{
    for (int i = 0; i < count_lines; ++i)
    {
        int rand_key = rand() % count_lines;
        swap_table(&table[i], &table[rand_key]);
    }
}

void print_table(int count_lines, t_table *table, t_input *input)
{
    printf("\n| key | Verse\n");

```



```

printf("-----
-----\n");
for (int i = 0; i < count_lines; i++)
{
    printf("|  %d  %c  |%s\n",  table[i].key.keyint,
    table[i].key.keychar,
    input[table[i].index].verse);
    printf("-----
-----\n");
}
printf("\n");
}
void print_menu()
{
    printf("1.Обновить           таблицу.\n2.Отсортировать
таблицу.\n3.Поиск           по           ключу.\n4.Реверс
таблицы.\n5.Случайные           ключи.\n6.Печать
таблицы.\n7.Выход.\nВыберите действие: ");
}
void main()
{
    t_table table[12];
    t_input input[12];
    int bi, count_lines=0, act, a,cs;
    char filename[15],c;
    printf("Введите название файла с таблицей: ");
    scanf("%s", filename);
    FILE* f=fopen(filename, "rb");
    count_lines=0;
    enter(f, table, input, &count_lines);
    print_menu();
    while (scanf("%d", &act)!=EOF)
    {
        printf("\n");
    }
}

```

```

switch (act)
{
    case 1:
        fclose(f);
        count_lines=0; cs=0; bi=0;
        printf("\nВведите      название      нового
        файла:");
        scanf("%s", filename);
        f=fopen(filename, "rb");
        enter(f,table,input,&count_lines);
        print_menu();
    break;
    case 2:
        cs=shell_sort(count_lines, table);
        print_menu();
    break;
    case 3:
        printf("\n");
        printf("Введите ключ: ");
        scanf("%d", &a);
        scanf("%c", &c);
        bi=binarysearch(a,c,                table,
        count_lines);
        if (cs!=1)
            printf("\nСначала      отсортируйте
            таблицу\n");
        else if (bi==-1)
            printf("\nТакого      элемента      не
            существует!\n");
        else
            printf("|      %d      %c      |%s\n",
            table[bi].key.keyint,
            table[bi].key.keychar,
            input[table[bi].index].verse);

```

```

        print_menu();
    break;
    case 4:
        reverse_table(table, count_lines);
        print_menu();
    break;
    case 5:
        random_keys(table, count_lines);
        print_menu();
    break;
    case 6:
        print_table(count_lines, table, input);
        print_menu();
    break;
    case 7:
        exit(0);
        break;
    default:
        printf("\nПожалуйста, выберите одно из
        доступных действий!\n");
        print_menu();
    break;
}
}
fclose(f);
}

```

8. Демонстрация работы программы **Содержание тестовых файлов:**

F1.txt:

32 а Волчонка Волк, начав помалу приучать
43 б Позавтракать иль пообедать!
11 l Обед готов; ничто не может быть вернее:
12 с Тут тотчас попадёшь в беду!
90 v Коль подлинно не плох пастух,
88 z Да сам пастух дурак;
11 а И съесть;
16 х И плохи, кажется, и смирны
1 б Позавтракать иль пообедать!

F2.txt:

1 б В прекрасный летний день,
1 с Бросая по долине тень,
1 а Листы на дереве с зефирами шептали,
1 f Хвалились густотой, зеленостью своей
2 v И вот как о себе зефирам толковали:
2 с Что нами дерево так пышно и кудряво,
4 g Раскидисто и величаво?
2 с Насвистывает соловей.
3 k Да вы, зефиры, сами
3 а Мы корни дерева, на коем вы цветете.

Результат работы программы:

kircatle@DESKTOP-
70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/c
ourse9\$ gcc c9.c

```
kircatle@DESKTOP-70J5NO3:/mnt/c/Users/joker/Desktop/LabsMAI/SecondSem/course9$ ./a.out
```

Введите название файла с таблицей: f1.txt

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 6

| key | Verse

| 32 a |Волчонка_Волк_начав_помалу_приучать

| 43 b |Позавтракать_иль_ообедать!

| 11 l |Обед_готов;_ничто_не_может_быть_вернее:

| 12 c |Тут_тотчас_попадёшь_в_беду!

| 90 v |Коль_подлинно_не_плох_пастух,

| 88 z |Да_сам_пастух_дурак;

| 11 a |И_съесть;

| 16 x |И_плохи,_кажется,_и_смирны

| 1 b |Позавтракать_иль_пообедать!

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 2

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 6

| key | Verse

| 1 b |Позавтракать_иль_пообедать!

| 11 a |И_съесть;

| 11 l |Обед_готов;_ничто_не_может_быть_вернее:

| 12 c |Тут_тотчас_попадёшь_в_беду!

| 16 x |И_плохи,_кажется,_и_смирны

| 32 a |Волчонка_Волк_начав_помалу_приучать

| 43 b |Позавтракать_иль_ообедать!

| 88 z |Да_сам_пастух_дурак;

| 90 v |Коль_подлинно_не_плох_пастух,

-
- 1.Обновить таблицу.
 - 2.Отсортировать таблицу.
 - 3.Поиск по ключу.
 - 4.Реверс таблицы.
 - 5.Случайные ключи.
 - 6.Печать таблицы.
 - 7.Выход.

Выберите действие: 4

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 6

| key | Verse

| 90 v |Коль_подлинно_не_плох_пастух,

| 88 z |Да_сам_пастух_дурак;

| 43 b |Позавтракать_иль_ообедать!

| 32 a |Волчонка_Волк_начав_помалу_приучать

| 16 x |И_плохи,_кажется,_и_смирны

| 12 c |Тут_тотчас_попадёшь_в_беду!

| 11 l |Обед_готов;_ничто_не_может_быть_вернее:

| 11 a |И_съесть;

| 1 b |Позавтракать_иль_пообедать!

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.

7.Выход.

Выберите действие: 2

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 6

| key | Verse

| 1 b |Позавтракать_иль_пообедать!

| 11 a |И_съесть;

| 11 1 |Обед_готов;_ничто_не_может_быть_вернее:

| 12 c |Тут_тотчас_попадёшь_в_беду!

| 16 x |И_плохи,_кажется,_и_смирны

| 32 a |Волчонка_Волк_начав_помалу_приучать

| 43 b |Позавтракать_иль_ообедать!

| 88 z |Да_сам_пастух_дурак;

| 90 v |Коль_подлинно_не_плох_пастух,

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 5

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 6

| key | Verse

| 11 l |Обед_готов;_ничто_не_может_быть_вернее:

| 43 b |Позавтракать_иль_ообедать!

| 11 a |И_съесть;

| 16 x |И_плохи,_кажется,_и_смирны

| 32 a |Волчонка_Волк_начав_помалу_приучать

| 12 c |Тут_тотчас_попадёшь_в_беду!

| 90 v |Коль_подлинно_не_плох_пастух,

| 1 b |Позавтракать_иль_пообедать!

| 88 z |Да_сам_пастух_дурак;

-
- 1.Обновить таблицу.
 - 2.Отсортировать таблицу.
 - 3.Поиск по ключу.
 - 4.Реверс таблицы.
 - 5.Случайные ключи.
 - 6.Печать таблицы.
 - 7.Выход.

Выберите действие: 2

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 6

| key | Verse

| 1 b |Позавтракать_иль_пообедать!

| 11 a |И_съесть;

| 11 l |Обед_готов;_ничто_не_может_быть_вернее:

| 12 c |Тут_тотчас_попадёшь_в_беду!

| 16 x |И_плохи,_кажется,_и_смирны

| 32 a |Волчонка_Волк_начав_помалу_приучать

| 43 b |Позавтракать_иль_ообедать!

| 88 z |Да_сам_пастух_дурак;

| 90 v |Коль_подлинно_не_плох_пастух,

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.

7.Выход.

Выберите действие: 3

Введите ключ: 88 z

| 88 z |Да_сам_пастух_дурак;

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 3

Введите ключ: 1 b

| 1 b |Позавтракать_иль_пообедать!

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 1

Введите название нового файла:f2.txt

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 6

key Verse

1 b В_прекрасный_летний_день,

1 c Бросая_по_долине_тень,

1 a Листы_на_дереве_с_зефирами_шептали,
--

1 f Хвалились_густотой,_зеленостью_своей

| 2 v |И_вот_как_о_себе_зефирам_толковали:

| 2 с |Что_нами_дерево_так_пышно_и_кудряво,

| 4 g |Раскидисто_и_величаво?

| 2 с |Насвистывает_соловей.

| 3 k |Да_вы,_зефиры,_сами

| 3 а |Мы_корни_дерева,_на_коем_вы_цветете.

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 2

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 6

| key | Verse

| 1 a |Листы_на_дереве_с_зефирами_шептали,

| 1 b |В_прекрасный_летний_день,

| 1 c |Бросая_по_долине_тень,

| 1 f |Хвалились_густотой,_зеленостью_своей

| 2 c |Что_нами_дерево_так_пышно_и_кудряво,

| 2 c |Насвистывает_соловей.

| 2 v |И_вот_как_о_себе_зефирам_толковали:

| 3 a |Мы_корни_дерева,_на_коем_вы_цветете.

| 3 k |Да_вы,_зефиры,_сами

| 4 g |Раскидисто_и_величаво?

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 4

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 6

| key | Verse

| 4 g | Раскидисто_и_величаво?

| 3 k | Да_вы,_зефиры,_сами

| 3 a | Мы_корни_дерева,_на_коем_вы_цветете.

| 2 v | И_вот_как_о_себе_зефирам_толковали:

| 2 c | Насвистывает_соловей.

| 2 c | Что_нами_дерево_так_пышно_и_кудряво,

| 1 f | Хвалились_густотой,_зеленостью_своей

| 1 c | Бросая_по_долине_тень,

| 1 b | В_прекрасный_летний_день,

| 1 a | Листы_на_дереве_с_зефирами_шептали,

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 2

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 6

| key | Verse

| 1 а |Листы_на_дереве_с_зефирами_шептали,

| 1 b |В_прекрасный_летний_день,

| 1 c |Бросая_по_долине_тень,

| 1 f |Хвалились_густотой,_зеленостью_своей

| 2 c |Что_нами_дерево_так_пышно_и_кудряво,

| 2 c |Насвистывает_соловей.

| 2 v |И_вот_как_о_себе_зефирам_толковали:

| 3 a |Мы_корни_дерева,_на_коем_вы_цветете.

| 3 k |Да_вы,_зефиры,_сами

| 4 g |Раскидисто_и_величаво?

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 5

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 6

| key | Verse

| 1 f |Хвалились_густотой,_зеленостью_своей

| 3 k |Да_вы,_зефиры,_сами

| 1 с |Бросая_по_долине_тень,

| 2 с |Что_нами_дерево_так_пышно_и_кудряво,

| 2 c |Насвистывает_соловей.

| 1 a |Листы_на_дереве_с_зефирами_шептали,

| 1 b |В_прекрасный_летний_день,

| 3 a |Мы_корни_дерева,_на_коем_вы_цветете.

| 4 g |Раскидисто_и_величаво?

| 2 v |И_вот_как_о_себе_зефирам_толковали:

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 2

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие: 6

| key | Verse

| 1 a |Листы_на_дереве_с_зефирами_шептали,

| 1 b |В_прекрасный_летний_день,

| 1 c |Бросая_по_долине_тень,

| 1 f |Хвалились_густотой,_зеленостью_своей

| 2 c |Насвистывает_соловей.

| 2 c |Что_нами_дерево_так_пышно_и_кудряво,

| 2 v |И_вот_как_о_себе_зефирам_толковали:

| 3 a |Мы_корни_дерева,_на_коем_вы_цветете.

| 3 k |Да_вы,_зефиры,_сами

| 4 g |Раскидисто_и_величаво?

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.
- 7.Выход.

Выберите действие: 3

Введите ключ: 3 k

| 3 k |Да_вы,_зефиры,_сами

- 1.Обновить таблицу.
- 2.Отсортировать таблицу.
- 3.Поиск по ключу.
- 4.Реверс таблицы.
- 5.Случайные ключи.
- 6.Печать таблицы.

7.Выход.

Выберите действие: 3

Введите ключ: 2 v

| 2 v |И_вот_как_о_себе_зефирам_толковали:

1.Обновить таблицу.

2.Отсортировать таблицу.

3.Поиск по ключу.

4.Реверс таблицы.

5.Случайные ключи.

6.Печать таблицы.

7.Выход.

Выберите действие:

Литература

Методические указания к выполнению курсовых работ. Зайцев В. Е.

https://learn.c.info/c/text_files.html

<http://www.sbp-program.ru/c/sbp-file-c.htm>

http://files.mai.ru/site/priem/documents/orders/2017/111_03.08.2017.pdf

