Студента группы		по курсу:	1 фундаментальна	я информатика		
Работа выполнена: «23»aBrycta2022г. Преподаватель:Tutob B. K, каф. 806 Входной контроль знаний с оценкой		студента	группы <u>М80-101Б-21</u>	Тулина Ивана	, № по списку:	22
Преподаватель:				Контакты www, e-mail,	icq, skype: <u>i.tulin010</u>	7@gmail.c
Входной контроль знаний с оценкой				Работа выполнена: «23	» <u>августа</u> 2022г.	
Отчет сдан « »				Преподаватель:	в В. К. каф. 806	
Подпись преподавателя				Входной контроль знан	ний с оценкой	
Сема: Сортировка и поиск Цель работы: Составить программу на языке Си с использованием процедур и функций для сортировки таблицы заданным методом и двоичного поиска по ключу в таблице. адание (вариант № 22): Метод сортировки: сортировка Шелла. Структура таблицы: № тип ключа длина ключа в байтах хранение данных и ключей элементов таблицы минимальное число элементов таблицы 1 целый 8 вместе 11 Оборудование (лабораторное): , процессор , имя узла сети , с ОП , Мб, НМД , Принтер , Мб, НМД , Мб. Терминал , адрес , Принтер , Принтер , Принтер , Процессор , Intel Core i5-7300HQ , с ОП , 7,87 , Мб, НМД , 15360 , Мб. Монитор: встроенный , Пругие устройства , наименование , версия , натерпретатор команд , версия , наименование , версия , натерпретатор команд , версия , версия , наименование , версия , наименования , версия , наименования , версия , наименования , версия , наименования , версия , наименование , наименование , версия , наименование , наименование , наименование , версия , наименование , на				Отчет сдан « »	201 г., итоговая	і оценка
Дель работы:				Под	пись преподавателя	
Дель работы:	Тема:		Сор	гировка и поиск		
таблицы заданным методом и двоичного поиска по ключу в таблице. Тип ключа Длина ключа в байтах хранение данных и ключей минимальное число элементов таблицы:	_	_				
адание (вариант № 22): Метод сортировки: сортировка Шелла. Структура таблицы: №					едур и функций для сор	тировки
№ Тип ключа длина ключа в байтах хранение данных и ключей минимальное число элементов таблицы 1 целый 8 вместе 11 Оборудование (лабораторное):	таолиц	<u>ы заданным ме</u>	тодом и двоичного поис	ка по ключу в таолице.		
№ Тип ключа длина ключа в байтах хранение данных и ключей минимальное число элементов таблицы 1 целый 8 вместе 11 Оборудование (лабораторное): ЭВМ , процессор , имя узла сети с ОП Мб. Мб. Терминал адрес Принтер	алание (е	апиант № 22) ₂	• Метод соптиповки:	соптировка Шелла Ст	nykmyna mahauuы:	
Целый 8 Вместе 11	<u> </u>	•				
1 целый 8 вместе 11 Оборудование (лабораторное): ОВМ , процессор , имя узла сети с ОП	145	тип ключа	длина ключа в оаитах			
Оборудование (лабораторное): ОВМ, процессор, имя узла сети	1	пош н	Q	DMOCTO		
ВВМ	1	целыи	0	вместе	11	
Оборудование ПЭВМ студента, если использовалось: ПроцессорIntel Core i5-7300HQ _c ОП _7,87 _M6, НМД _15360 _M6. Монитор:встроенный Другие устройства Программное обеспечение (лабораторное): Операционная система семейства наименование версия Еистема программирования версия Редактор текстов версия Утилиты операционной системы Прикладные системы и программы: _ Местонахождение и имена файлов программ и данных	9BM	, процес	ccop			
ПроцессорIntel Core i5-7300HQ _ с ОП _ 7,87 _ Мб, НМД _ 15360 _ Мб. Монитор: встроенный	ЭВМ НМД	, процес Мб. Те	ссор, рминал адрес	с, Г	Тринтер	
ПроцессорIntel Core i5-7300HQ _ с ОП _ 7,87 _ Мб, НМД _ 15360 _ Мб. Монитор: встроенный Другие устройства	ЭВМ НМД	, процес Мб. Те	ссор, рминал адрес	с, Г	Тринтер	
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	ЭВМ НМД Другие уст	, процес Мб. Теј гройства	ссор <u> </u>	c I	Тринтер	
Программное обеспечение (лабораторное): Операционная система семейства наименование	ЭВМ НМД Цругие уст	, процес Мб. Теј гройства иние ПЭВМ сту	ссор, рминал адрес	С Г	Іринтер	
Операционная система семейства наименование версия, интерпретатор команд версия	ЭВМ НМД Цругие уст Оборудова Процессор	, процес Мб. Те Мб. Те гройства ние ПЭВМ сту Intel Core i5	ссор	с I ось: Иб, НМД <u>15360</u> Мб. М	Іринтер	
Операционная система семейства наименование версия, интерпретатор команд версия	ЭВМ НМД Другие уст Оборудова Процессор	, процес Мб. Те Мб. Те гройства ние ПЭВМ сту Intel Core i5	ссор	с I ось: Иб, НМД <u>15360</u> Мб. М	Іринтер	
мнтерпретатор команд	ЭВМ НМД Цругие уст Оборудова Процессор Цругие уст	, процес Мб. Те Мб. Те гройства ние ПЭВМ сту Intel Core i5 гройства	ссор адрес рминал адрес удента, если использовалс -7300HQ с ОП _7,87_ М	с I ось: Иб, НМД <u>15360</u> Мб. М	Іринтер	
Система программирования	ЭВМ НМД Другие уст Оборудова Процессор Другие уст	, процес Мб. Те гройства ние ПЭВМ сту Intel Core i5 гройства пное обеспечен	ссор адресор адресор адресор адресординал адресордина (2.7300HQc OП7,87 Мотераторное):	с I ось: Иб, НМД <u>15360</u> Мб. М	Іринтер Іонитор: <u>встроенны</u>	й
Редактор текстов	ЭВМ	, процес Мб. Теј Мб. Теј гройства мние ПЭВМ сту Intel Core i5 гройства мное обеспечениная система се	ссор адресор адресор адресор адресординал адресордина наговаторное):	с Г ось: Иб, НМД <u>15360</u> Мб. М	Іринтер Іонитор: <u>встроенны</u>	й
Прикладные системы и программы: _ Местонахождение и имена файлов программ и данных	ЭВМ	, процес Мб. Теройства пние ПЭВМ стуройства пное обеспеченная система сертор команд	ссор, рминал адресор адресор с ОП _7,87 _ М лие (лабораторное): емейства наи верс	с Г ось: Иб, НМД <u>15360</u> Мб. М именование	Іринтер Іонитор: <u>встроенны</u> версия	й
Местонахождение и имена файлов программ и данных	ЭВМ	, процес Мб. Теройства пние ПЭВМ стуройства пное обеспеченная система серограммирование систов	ссор, рминал адрес годента, если использовале в-7300НQ с ОП _7,87 М гие (лабораторное): емейства наи верс	с Г ось: Мб, НМД <u>15360</u> Мб. М именование ия версия	Іринтервстроенны версия	й
Местонахождение и имена файлов программ и данных	ЭВМ	, процес Мб. Теройства пние ПЭВМ стуройства пное обеспеченная система серограммирование систов	ссор, рминал адрес годента, если использовале в-7300НQ с ОП _7,87 М гие (лабораторное): емейства наи верс	с Г ось: Мб, НМД <u>15360</u> Мб. М именование ия версия	Іринтервстроенны версия	й
Программное обеспечение ЭВМ студента, если использовалось:	ЭВМ	, процес Мб. Теройства пние ПЭВМ стуройства пное обеспеченная система сертор команд рограммироване сестов	ссор адресор адресор адресор адресор адресор по	с Г ось: Мб, НМД <u>15360</u> Мб. М именование ия версия	Іринтервстроенны версия	й
	ЭВМ	, процес Мб. Теройства Мб. Теройства Ппе Соге і 5 тройства Ппе обеспечен ная система серограммирован екстов перационной сыве системы и п	ссор	с Г ось: Иб, НМД <u>15360</u> Мб. М именование <u> </u> ия <u> </u> версия <u> </u>	Іринтервстроенныверсияверсия	й
	ЭВМ	, процес Мб. Теройства Мб. Теройства Ппе Соге і 5 тройства Ппе обеспечен ная система серограммирован екстов перационной сыве системы и п	ссор	с Г ось: Иб, НМД <u>15360</u> Мб. М именование <u> </u> ия <u> </u> версия <u> </u>	Іринтервстроенныверсияверсия	й
	ЭВМ	, процес Мб. Теройства пние ПЭВМ ступать стройства пное обеспеченная система серограммированне система серограционной сове системы и прождение и иментождение	ссор, рминал адрес одента, если использовало одент	с Г ось: Мб, НМД _15360_ Мб. М. именование ия версия	Іринтервстроенныверсияверсия	й
Операционная система семейства <u>UNIX</u> , наименование <u>Ubuntu</u> версия <u>20.04.3 LTS</u>	ЭВМ	, процес Мб. Теройства мб. Теройства мб. Теройства мное обеспечени программирование системы и прождение и именное обеспечени	ссор	с Г ось: мб, НМД _15360_ Мб. М. именование ия версия нных	Іонитор: <u>встроенны</u> версия	й
интерпретатор команд <u>bash</u> версия	ЭВМ	, процес Мб. Теройства мб. Теройства метеройства метеройства метеройства метерограммирование системы и прождение и именерождение и именерое обеспечения система се	гоор, рминал адрестрента, если использовале на на на верстия рограммы: _ на файлов программ и дане ЭВМ студента, если истемы е ЭВМ студента, если истемы на на на на на на файлов программ и дане е ЭВМ студента, если истемы на	с Г ось: мб, НМД15360 Мб. М. мменование версия нных спользовалось: вименование Ubunt	Іонитор: <u>встроенны</u> версия	й
REMEMBORNAME REPORTED TO A STATE OF THE REPORT OF THE REPO	ЭВМ	, процес Мб. Теройства мб. Теройства пнее ПЭВМ ступное обеспечения системы и прождение и именное обеспечени и и и и и и и и и и и и и и и и и и	гоор	с Г ось: мб, НМД15360 Мб. М. мменование версия нных спользовалось: нименование Ubunt	Принтер встроенны версия версия версия	й
Department Type Parameter Type Param	ЭВМ	, процес Мб. Теройства Мб. Теройства Ппее Соге і 5 гройства Ппее обеспечени прождение и имения системы и программирования системы и программирования система серограммирования система серограммирования системы и программирования системы серограммирования серограммирования серойствания серограммирования серойствания серограммирования серойствания серограммирования серойствания серойствания серограммирования серойствания серойст	гоор	с Г ось: мб, НМД15360 Мб. М. мменование версия нных спользовалось: аименование Ubunt	Принтер	й
	ЭВМ	, процес Мб. Теройства мб. Теройства пнее ПЭВМ ступное обеспечения системы и прождение и именное обеспечени и и и и и и и и и и и и и и и и и и	гоор	с Г ось: мб, НМД15360 Мб. М. мменование версия нных спользовалось: нименование Ubunt	Принтер встроенны версия версия версия	й , 4.3 LTS
Система программированияверсияверсияверсияверсия Редактор текстов Еmacs версия 3.22.30 Утилиты операционной системы	ЭВМ	, процес Мб. Теройства пние ПЭВМ ступе	гоор	с Г ось: мб, НМД15360 Мб. М. мменование версия нных спользовалось: аименование Ubunt	Принтер	й

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

Идея:

Для описания строки таблицы используется структура, которая состоит из целочисленного ключа строки и символьного массива, к которую будет помещена сама строка.

Работа программы реализована посредством следующих процедур и функций:

- Функция, которая открывает файл с входными данными;
- Функция, которая считает количество строк в файле;
- Функция, формирующая из входных данных таблицу;
- Процедура, печатающая полученную таблицу;
- Процедура, реверсирующая положение строк в таблице;
- Процедура, перемешивающая строки в таблице;
- Процедура, сортирующая строки в таблице;
- Функция двоичного поиска строки в таблице.

Для быстрой проверки отсортированности таблицы выделим отдельную переменную *isSorted* в которую будем помещать 1, если таблица отсортирована, и 0, если она не отсортирована. Изначально переменная равна нулю.

Функция, открывающая входные файлы на чтение

Функция открывает файл при помощи функции *fopen()*, проверяя успешность её выполнения в условном операторе. В случае неудачного выполнения возвращает нулевой указатель. В случае удачного выполнения выполнения она вернет указатель на файл.

Функция, считающая количество строк в файле

Функция в цикле просматривает файл до конца, увеличивая счетчик каждый раз, когда встречает знак перехода на новую строку. После прочтения файла при помощи процедуры *rewind()*, возвращает указатель к началу файла. Возвращает полученное значение счетчика.

Функция, формирующая из входных данных таблицу

Функция при помощи *new* создает массив строк таблицы (структура из ключа и строки) длины, вычисленной ранее в функции подсчета строк. Затем считывает данные из файла в созданную таблицу. Считывание происходит следующим образом: при помощи *fscanf()* считывается ключ строк, затем при помощи *fgets()* считывается сама строка. Результатом работы функции является указатель на созданный массив.

Процедура, печатающая полученную таблицу

В цикле по счетчику печатаются строки таблицы: сначала ключ, потом строка.

Процедура, реверсирующая положение строк в таблице

В цикле счетчиком процедура меняет i-ую строку таблицы с (size-i)-ой, где size – количество строк.

В конце выполнения процедура меняет значение *isSorted* на ноль.

Процедура, перемешивающая строки в таблице

При помощи *rand()%size* генерируются 2 числа – номера строк в таблице, которые процедура меняет местами. В конце выполнения процедура меняет значение *isSorted* на ноль.

Процедура сортировки строк в таблице по ключу

Процедура сортирует строки по ключу методом сортировки Шелла. Сортировка Шелла является улучшением метода вставки. Её скорость зависит от выбора шага сортировки.

Мы изначально будем использовать шаг сортировки равный половине размера массива. Сортируется массив в цикле, который останавливается, когда новый шаг сортировки принимает нулевое значение. В начале каждой итерации цикла будем уменьшать значение шага сортировки в 2 раза. В каждой итерации будет происходить сортировка вставкой для элементов, отстоящих друг от друга на один шаг сортировки. Т. е. если значение і-ой строки меньше, чем значение (s-i)-ой строки в таблице, то процедура будет менять эти строки местами.

Полезная особенность сортировки Шелла состоит в отсутствии необходимости создания дополнительных массивов для выполнения сортировки. Это очень удобно при сортировки таблиц.

В конце выполнения процедура меняет значение *isSorted* на единицу.

Функция бинарного поиска строки в таблице.

Изначально функция проверяет отсортирована ли таблица. В случае если она не отсортирована, функция печатает сообщение об ошибке и возвращает -1.

Функция принимает на вход указатель на таблицу, её размер и значение ключа искомой строки.

Для поиска задаются левая и правая граница поиска, изначально равные граница массива. Поиск по ключу происходит в цикле, который останавливается, когда значение левой границы оказывается больше правой. В каждой итерации случайным образом выбирается строка между левой и правой границей. Значение её ключа сравнивается со значением ключа искомой строки. Если значения ключей совпадают, то функция возвращает номер найденной строки (индекс элемента массива). Если значение ключа искомой строки больше чем то, с которым происходит сравнение— левая граница поиска смещается к индексу строки, с которой проводилось сравнение. В противоположном случае— смещается правая граница. Если за все итерации цикла строка не была найдена, функция возвращает -1.

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

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
const int N=128;
int is_sorted=0;
struct iString
{ long key;
 char *string;
void randomize()
{ long a=time(0);
 srand(a);
FILE* openFile(int argc, char *argv)
{ FILE *input;
 if(argc==1)
   if(!(input=fopen("input.txt", "r")))
    { printf("Error: can't open file input.txt\n");
     char n=getchar(); printf("%c\n", n);
     return 0;
 else if(argc==2)
  { if(!(input=fopen(argv, "r")))
    { printf("Error: can't open file %s\n", argv);
     return 0;
    }
 else
  { printf("Error: wrong input\n");
   return 0;
 return input;
int iSize(FILE* input)
{ char buffer; int n=0;
 while(!feof(input))
 { buffer=fgetc(input);
  if(buffer=='\n')
     n++;
 rewind(input);
 return n;
}
iString *makeTable(FILE *input, int size)
{ iString *table = new iString[size];
 for(int i=0; i<size; i++)
  table[i].string= new char[N];
 int l=N*sizeof(char);
for(int i=0; i<size; i++)</pre>
 { fscanf(input, "%d", &table[i].key);
  fgets(table[i].string, l, input);
return table;
void printBoard()
{ printf("+---+-"
                     ----\n" );
}
```

```
void printTable(iString *table, int size)
{ printEoard(); printf("|KEY|STRING\n");
 printBoard();
 for(int i=0; i<size; i++) { printf("|%3d|", table[i].key); printf(" %s", table[i].string);
 printBoard();
printf("\n\n");
void swap(iString *table, int first, int second)
{ iString buffer;
 buffer = table[first];
 table[first] = table[second];
table[second] = buffer;
void reverse(iString *table, int size)
{ for(int i=0; i<size/2; i++)
   swap(table, i, size-i-1);
 is_sorted=0;
void mix(iString *table, int size)
{ int first, second;
 for(int i=0; i<size/2; i++) { first=rand()%size;
   second=rand()%size;
   swap(table, first, second);
 is_sorted=0;
void sortTable(iString *table, int size)
{ int s=1, i, j;
 while(s<size) s*=2;
 while(s>0)
  \{ s=(s-1)/2;
   for(i=0; i<size-s; i++)
   for(j=i+s; j-s>=0; j-=s)
if(table[j-s].key>table[j].key)
      swap(table, j-s, j);
    else break;
 is_sorted=1;
int binSearch(iString *table, int size, long x)
{ if(!is_sorted)
 { printf("Error: table isn`t sorted\n");
   return -1;
 int l=0, r=size-1, m;
 while(l<=r)
  { m=rand()%(r-l)+l;
   if(table[m].key==x) return m;
   if(table[m].key<x) \hat{l}=m+1;
   else r=m;
 return -1;
```

```
void menu()
{ printf("\n=
                        _\n''
            _Menu_
     "1. Print table\n"
     "2. Binary search\n"
     "3. Sort\n"
     "4. Mix\n"
     "5. Reverse\n"
     "6. Menu\n"
     "0. Exit\n"
     "=======\n");
}
int main(int argc, char *argv[])
{ FILE *input; int size;
 iString *table; randomize();
 input = openFile(argc,argv[1]);
 if(!input) return 1;
 size=iSize(input);
 table = makeTable(input, size);
 int action=6;
 while(action)
 { if(action==1)
     printTable(table, size);
  else if(action==2)
  { printf("\nInput key\n");
    long x; int i;
    scanf("%d", &x);
    i=binSearch(table, size, x);
   if(i<0)
    printf("String not found\n");
   eÎse
     printf("Found the string:\n %s",
               table[i].string);
  else if(action==3)
  { sortTable(table, size);
   printf("\nTable was sorted\n");
  else if(action==4)
  { mix(table, size);
    printf("\nTable was mixed\n");
  else if(action==5)
  { reverse(table, size);
   printf("\nTable was reversed\n");
  else if(action==6)
  menu();
printf("\nInput number of action\n");
  scanf("%d", &action);
return 0;}
Тестовый файл input.txt:
241
333
357
412
431 /
479
500
742
788
890
999 /
```

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

```
yusayu@YS:~/Рабочий стол/сррРгојесts/kr9$ cat head
                      Курсовая работа ІХ
                      Сортировка и поиск
                Выполнил: Тулин Иван Денисович
                     (номер по списку: 22)
                      Группа: М80-101Б-21
******************
yusayu@YS:~/Рабочий стол/сррРгојесts/kr9$ cat kr9.cpp
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
const int N=128;
int is_sorted=0;
struct iString
{ long key;
 char *string;
};
void randomize()
{ long a=time(0);
  srand(a);
}
FILE* openFile(int argc, char *argv)
{ FILE *input;
  if(argc==1)
    { if(!(input=fopen("input.txt", "r")))
      { printf("Error: can't open file input.txt\n");
        char n=getchar(); printf("%c\n", n);
        return 0;
      }
   }
  else if(argc==2)
    { if(!(input=fopen(argv, "r")))
      { printf("Error: can't open file %s\n", argv);
        return 0;
   }
  else
    { printf("Error: wrong input\n");
     return 0;
  return input;
int iSize(FILE* input)
{ char buffer; int n=0;
  while(!feof(input))
  { buffer=fgetc(input);
    if(buffer=='\n')
       n++;
  rewind(input);
  return n;
```

```
}
iString *makeTable(FILE *input, int size)
{ iString *table = new iString[size];
 for(int i=0; i<size; i++)</pre>
   table[i].string= new char[N];
 int l=N*sizeof(char);
 for(int i=0; i<size; i++)</pre>
 { fscanf(input, "%ld", &table[i].key);
   fgets(table[i].string, l, input);
return table;
void printBoard()
{ printf("+---+
        "----\n" );
}
void printTable(iString *table, int size)
{ printBoard(); printf("|KEY|STRING\n");
 printBoard();
 for(int i=0; i<size; i++)</pre>
 { printf("|%3ld|", table[i].key);
   printf(" %s", table[i].string);
 printBoard();
 printf("\n\n");
void swap(iString *table, int first, int second)
{ iString buffer;
 buffer = table[first];
 table[first] = table[second];
 table[second] = buffer;
}
void reverse(iString *table, int size)
{ for(int i=0; i<size/2; i++)
   swap(table, i, size-i-1);
 is_sorted=0;
}
void mix(iString *table, int size)
{ int first, second;
 for(int i=0; i<size/2; i++)
 { first=rand()%size;
   second=rand()%size;
   swap(table, first, second);
 is_sorted=0;
void sortTable(iString *table, int size)
{ int s=1, i, j;
 while(s<size) s*=2;
 while(s>0)
```

```
{s=(s-1)/2;}
   for(i=0; i<size-s; i++)
    for(j=i+s; j-s>=0; j-=s)
      if(table[j-s].key>table[j].key)
        swap(table, j-s, j);
     else break;
  is_sorted=1;
}
int binSearch(iString *table, int size, long x)
{ if(!is_sorted)
  { printf("Error: table isn`t sorted\n");
    return -1;
  int l=0, r=size-1, m;
 while(l<=r)
  { m=rand()%(r-l)+l;
    if(table[m].key==x) return m;
   if(table[m].key<x) l=m+1;</pre>
   else r=m;
  return -1;
void menu()
{ printf("\n=======\n"
          ____\n"
        "1. Print table\n"
        "2. Binary search\n"
        "3. Sort\n"
        "4. Mix\n"
        "5. Reverse\n"
        "6. Menu\n"
        "0. Exit\n"
        "======\n");
}
int main(int argc, char *argv[])
{ FILE *input; int size;
  iString *table;
  randomize();
  input = openFile(argc,argv[1]);
  if(!input)
    return 1;
  size=iSize(input);
  table = makeTable(input, size);
  int action=6;
  while(action)
  { if(action==1)
        printTable(table, size);
    else if(action==2)
    { printf("\nInput key\n");
      long x; int i;
      scanf("%ld", &x);
      i=binSearch(table, size, x);
      if(i<0)
        printf("String not found\n");
      else
        printf("Found the string:\n %s",
                        table[i].string);
```

```
else if(action==3)
    { sortTable(table, size);
     printf("\nTable was sorted\n");
   else if(action==4)
    { mix(table, size);
     printf("\nTable was mixed\n");
   else if(action==5)
    { reverse(table, size);
     printf("\nTable was reversed\n");
   else if(action==6)
     menu();
   printf("\nInput number of action\n");
   scanf("%d", &action);
  }
return 0;}
yusayu@YS:~/Рабочий стол/cppProjects/kr9$ cat input.txt
241
333
                \/ /
                                     / /| | / /_/ / /
357
412
431 /_/
479
500
742
            \_ \/ /
788
890
                                        _ |/ _, _/ / /
                                       |_/_/ |_| /_/
999 /_/
yusayu@YS:~/Рабочий стол/cppProjects/kr9$ g++ -o kr9.out kr9.cpp
yusayu@YS:~/Рабочий стол/cppProjects/kr9$ ./kr9.out
___Menu__
1. Print table
2. Binary search
3. Sort
4. Mix
5. Reverse
6. Menu
0. Exit
Input number of action
1
|KEY|STRING
|241|
|333|
|412|
|431|
|479|
|500|
|742|
|788|
                                       //| | / /_/ / / /
                                           _ |/
|890|
|999|
```

}

```
Table was reversed
Input number of action
|KEY|STRING
                          17881
      / | / ___// __
|742|
15001
1479|
431
|412|
|357|
                            / i / _ \/_
|333|
|241|
Input number of action
6
____Menu___
1. Print table
2. Binary search
3. Sort
4. Mix
5. Reverse
6. Menu
0. Exit
Input number of action
3
Table was sorted
Input number of action
|KEY|STRING
+---+-----
|241|
|333|
|357|
|412|
|431|
|479|
|500|
|742|
      |788|
|890|
```

Input number of action

|999|

```
Input number of action
Table was mixed
Input number of action
|KEY|STRING
12411
|742|
|890|
|412|
|357|
|479|
|500|
|999|
|788|
|333|
|431|
Input number of action
Table was sorted
Input number of action
|KEY|STRING
|241|
13331
13571
|412|
|431|
|479|
|500|
|742|
|788|
|890|
|999|
Input number of action
Input key
500
Found the string:
```

Input number of action

Table was reversed

Input number of action

Input key 999

Error: table isn`t sorted String not found

Input number of action

2	Лаб. или дом.	Дата	Время	Событие	Действие по исправлению	Примечание
1	0. Зам	ечания	автора по су	ществу работы	•	
1	1. Вы					
	D XC	де раос	<u>лы я научи</u>	іся создавать и отла	живать программы сортировки и	поиска в массиве на Си
_						
					_	
	Нелоці	ты при	выполнении за	адания могут быть устр	ранены следующим образом:	

Подпись студента _____

9. Дневник отладки должен содержать дату и время сеансов отладки, и основные события (ошибки в сценарии и