

Лабораторная работа №3

Создано системой Doxygen 1.9.5

1 Лабораторная работа №3	1
2 Список файлов	3
2.1 Файлы	3
3 Файлы	5
3.1 Файл D:/c++/labs/lab_3/functions.cpp	5
3.1.1 Функции	6
3.1.1.1 max()	6
3.1.1.2 print() [1/2]	6
3.1.1.3 print() [2/2]	6
3.1.1.4 read() [1/2]	7
3.1.1.5 read() [2/2]	7
3.1.1.6 solver()	8
3.1.1.7 sort()	8
3.1.1.8 swap()	8
3.1.1.9 usage()	9
3.1.2 Переменные	9
3.1.2.1 end	9
3.1.2.2 length	9
3.1.2.3 start	9
3.2 Файл D:/c++/labs/lab_3/main.cpp	9
3.2.1 Функции	10
3.2.1.1 main()	10
Предметный указатель	11

Глава 1

Лабораторная работа №3

Программа получает на вход количество отрезков и координаты их начала и конца. После чего возвращает координаты отрезков, полученных путем объединения выходных отрезков.

Ввод:

```
5
1.2 3.4
-1.8 0.1
0.5 1.5
1.3 2.8
-2 -1
```

Вывод:

```
Segment 1: -2 0.1
Segment 2: 0.5 3.4
```

Для того, чтобы скомпилировать программу, необходимо написать в консоли make. Команда make clean удаляет объектные файлы, команда make distclean удаляет объектные файлы и скомпилированную программу.

Глава 2

Список файлов

2.1 Файлы

Полный список файлов.

D:/c++/labs/lab_3/ functions.cpp	5
D:/c++/labs/lab_3/ main.cpp	9

Глава 3

Файлы

3.1 Файл D:/c++/labs/lab_3/functions.cpp

```
#include <iostream>
#include <fstream>
```

Функции

- void `usage` ()
Выводит в консоль справку.
- int `read` ()
Осуществляет чтение из стандартного потока cin.
- int `read` (char *file_name)
Осуществляет чтение из файла.
- int `print` (double cs, double ce, int seg_ind)
Осуществляет вывод результата в стандартный поток cout.
- int `print` (double cs, double ce, int seg_ind, const char file_name[])
Осуществляет вывод результата в файл.
- double `max` (double i, double k)
Возвращает максимальное значение из полученных.
- void `swap` (int i, int j)
Меняет местами элементы глобальных массивов.
- void `sort` ()
Сортирует массивы по возрастанию на основе значений элементов массива start[].
- int `solver` (int file, const char file_name[])
Находит объединения отрезков.

Переменные

- int `length`
Количество отрезков
- double * `start`
Массив начал отрезков
- double * `end`
Массив концов отрезков

3.1.1 Функции

3.1.1.1 max()

```
double max (
    double i,
    double k )
```

Возвращает максимальное значение из полученных.

Аргументы

in	i,k	- переменные для осуществления сравнения
----	-----	------------------------------------------

Возвращает

Максимальное значение

3.1.1.2 print() [1/2]

```
int print (
    double cs,
    double ce,
    int seg_ind )
```

Осуществляет вывод результата в стандартный поток cout.

Аргументы

in	cs	- координата начала отрезка
in	ce	- координата конца отрезка
in	seg_ind	- номер отрезка

Возвращает

0 - функция выполнена успешно

Функция выводит в стандартный поток cout номер сегмента, считая слева, и его координаты.

3.1.1.3 print() [2/2]

```
int print (
    double cs,
```

```
double ce,  
int seg_ind,  
const char file_name[] )
```

Осуществляет вывод результата в файл.

Аргументы

in	cs	- координата начала отрезка
in	ce	- координата конца отрезка
in	seg_ind	- номер отрезка
in	file_name	- имя файла

Возвращает

- 0 - функция выполнена успешно
- 1 - обнаружена ошибка

Функция выводит в файл номер сегмента, считая слева, и его координаты. Если файл не существует, то выведется соответствующее значение и функция прекратит свое выполнение.

3.1.1.4 read() [1/2]

```
int read ( )
```

Осуществляет чтение из стандартного потока cin.

Возвращает

- 0 - функция выполнена успешно
- 1 - обнаружена ошибка ввода

Функция выполняет чтение данных из стандартного потока cin в массивы start[] и end[]. Длина массивов задается в процессе выполнения функции. Если произошла ошибка ввода, то выведется соответствующее сообщение, и функция прекратит свое выполнение.

3.1.1.5 read() [2/2]

```
int read (  
    char * file_name )
```

Осуществляет чтение из файла.

Аргументы

in	file_name	- имя файла
----	-----------	-------------

Возвращает

- 0 - функция выполнена успешно
- 1 - обнаружена ошибка

Функция выполняет чтение данных из файла в массивы `start[]` и `end[]`. Длина массивов задается в процессе выполнения функции. Если произошла ошибка ввода или файл не удалось открыть, то выведется соответствующее сообщение, и функция прекратит свое выполнение.

3.1.1.6 solver()

```
int solver (
    int file,
    const char file_name[] )
```

Находит объединения отрезков.

Аргументы

in	file	- 0, 1 в зависимости от того, осуществляется вывод в файл или нет
in	file_name	- имя файла, в который осуществляется вывод

Возвращает

- 0 - функция выполнена успешно
- 1 - обнаружена ошибка

Функция сортирует массивы `start[]` и `end[]` при помощи функции `sort()`. Задаёт стартовые значения начала `cs` и конца `се` отрезка как у первого отрезка. Поочередно проходит по координатам каждого отрезка. Если координата начала данного отрезка лежит между `cs` и `се`, то она становится новой `cs`, а `се` становится максимальной из координат конца отрезка и `се`. Иначе выводятся `cs` и `се`, и им присваиваются значения данного отрезка.

3.1.1.7 sort()

```
void sort ( )
```

Сортирует массивы по возрастанию на основе значений элементов массива `start[]`.

Выполняет сортировку вставкой. Сортирует элементы массива `start[]` по возрастанию и переставляет соответствующие им элементы массива `end[]`.

3.1.1.8 swap()

```
void swap (
    int i,
    int j )
```

Меняет местами элементы глобальных массивов.

Аргументы

in	i,j	- индексы элементов
----	-----	---------------------

Меняет местами i-ый и j-ый элементы сначала массива start[], затем массива end[].

3.1.1.9 usage()

```
void usage ( )
```

Выводит в консоль справку.

Функция вызывается, если пользователь неверно ввел флаги при запуске программы.

3.1.2 Переменные

3.1.2.1 end

```
double* end
```

Массив концов отрезков

3.1.2.2 length

```
int length
```

Количество отрезков

3.1.2.3 start

```
double* start
```

Массив начал отрезков

3.2 Файл D:/c++/labs/lab_3/main.cpp

```
#include <iostream>
#include <cstring>
#include "functions.h"
```

Функции

- `int main (int argc, char **argv)`
Обрабатывает параметры запуска.

3.2.1 Функции

3.2.1.1 main()

```
int main (  
    int argc,  
    char ** argv )
```

Обрабатывает параметры запуска.

Аргументы

in	argc	- количество параметров запуска
in	argv	- список параметров запуска

Возвращает

- 0 - функция выполнена успешно
- 1 - обнаружена ошибка

В зависимости от введенных при запуске флагов определяет, откуда будут поступать данные, и куда их потом нужно будет вывести. Флаг `-tofile` сообщает, что вывод будет производиться в файл, после него указывается имя файла. Флаг `-fromfile` сообщает, что ввод будет производиться из файла, после него указывается имя файла. Возможна обработка обоих флагов одновременно. Если во время выполнения произошла ошибка, то выведется ошибка, и программа закончит выполнение с кодом 1, иначе с кодом 0.

Предметный указатель

D:/c++/labs/lab_3/functions.cpp, [5](#)

D:/c++/labs/lab_3/main.cpp, [9](#)

end

functions.cpp, [9](#)

functions.cpp

end, [9](#)

length, [9](#)

max, [6](#)

print, [6](#)

read, [7](#)

solver, [8](#)

sort, [8](#)

start, [9](#)

swap, [8](#)

usage, [9](#)

length

functions.cpp, [9](#)

main

main.cpp, [10](#)

main.cpp

main, [10](#)

max

functions.cpp, [6](#)

print

functions.cpp, [6](#)

read

functions.cpp, [7](#)

solver

functions.cpp, [8](#)

sort

functions.cpp, [8](#)

start

functions.cpp, [9](#)

swap

functions.cpp, [8](#)

usage

functions.cpp, [9](#)