

Лабораторная работа №5  
Выполнил Булхуков Сергей

## Функции

- Функция `max()` определяет наибольшее число из пары чисел.

```
double max(double i, double j)
{
    if (i > j) return i;
    return j;
}
```

Где `i` и `j` – 2 числа.

Функция `swap()` меняет местами `i`-тый и `j`-тый элементы массива `strt[]`, затем массива `fnl[]`. `i, j` – индексы элементов в массивах `strt[]` и `fnl[]`

```
void swap(int i, int j)
{
    double spis;
    spis = strt[i];
    strt[i] = strt[j]; strt[j] = spis;
    spis = fnl[i];
    fnl[i] = fnl[j]; fnl[j] = spis;
}
```

- Функция `sort()` сортирует массив `strt[]` по возрастанию, в соответствии с элементом в `strt[]` сортирует элемент в массиве `fnl[]`

```
void sort()
{
```

```

int k;
for (int i = 0; i < count - 1; ++i)
{
    k = i;
    for (int j = i + 1; j < count; ++j)
        if (strt[j] < strt[k])
            k = j;

    if (k != i)
        swap(k, i);
}
}

```

- Функция rd() осуществляет чтение из стандартного потока cin.

```

int rd()
{
    std::cin >> count;
    if (std::cin.fail() || count <= 0)
    {
        std::cerr << "<<< INCORRECT NUMBER >>>\n";
        return 1;
    }
}

```

```

strt = new double[count];
fnl = new double[count];

```

```

for (int i = 0; i < count; ++i)
{
    std::cin >> strt[i] >> fnl[i];
    if (std::cin.fail())
    {
        std::cerr << "<<< INCORRECT COORDINATES >>>\n";
        delete[] strt;
        delete[] fnl;
        return 1;
    }
}

```

```

    }
}
return 0;
}

```

- Функция `rd_f(char* filename)` осуществляет чтение из файла. Filename – это название файла.

```

int rd_f(char* filename)
{
    std::ifstream file;
    file.open(filename);
    if (file.is_open() == 0)
    {
        std::cerr << "<<< FILE DOES NOT EXIST >>>\n";
        return 1;
    }

    file >> count;
    if (file.fail() || count <= 0)
    {
        std::cerr << "<<< INCORRECT NUMBER >>>\n";
        return 1;
    }

    strt = new double[count];
    fnl = new double[count];

    for (int i = 0; i < count; ++i)
    {
        file >> strt[i] >> fnl[i];
        if (file.fail())
        {
            std::cerr << "<<< INCORRECT COORDINATES >>>\n";
            return 1;
        }
    }
}

```

```

    }
}
file.close();
return 0;
}

```

- Функция `int print(double f, double m, int otrezok)` осуществляет вывод результата в стандартный поток `cout`.

```

int print(double f, double m, int otrezok)
{
    std::cout << "result " << otrezok << ": "
               << f << '\t' << m << std::endl;
    return 0;
}

```

- Функция `int print_f(double f, double m, int otrezok, const char filename[])` Вписывает результат в файл.

```

int print_f(double f, double m, int otrezok, const char filename[])
{
    std::ofstream file;
    file.open(filename, std::ios_base::app);
    if (!file.is_open())
    {
        std::cerr << "<<< FILE DOES NOT EXIST >>>\n";
        return 1;
    }
    file << "result " << otrezok << ": "
          << f << '\t' << m << '\n';
    file.close();
    return 0;
}

```

- Функция `int sl(int file, const char filename[])` производит вычисление объединения отрезков.

```
int sl(int file, const char filename[])
{
    sort();
    int seg_ind = 1;
    double f = strt[0]; double m = fnl[0];
    for (int i = 1; i < count; ++i)
    {
        if (strt[i] >= f && strt[i] <= m)
            m = max(m, fnl[i]);
        else
        {
            if (!file) print(f, m, seg_ind++);
            else if (print_f(f, m, seg_ind++, filename)) return 1;
            f = strt[i]; m = fnl[i];
        }
    }
    delete[] strt;
    delete[] fnl;

    if (file == 0) print(f, m, seg_ind++);
    else if (print_f(f, m, seg_ind++, filename)) return 1;

    return 0;
}
```