

# Отчет по ЛР№5

## 1. Создаем структуру Str(Str-предложение)

```
struct Str
{
    int iStart;
    int Lenght;

    Str() {}

    Str(int ind, int len) {
        iStart = ind;
        Lenght = len;
    }
};
```

## 2. Открываем файл, из которого мы будем брать предложения для сортировки. И делаем возможность вводить нужный нам файл с консоли

```
cout << "Entered file vvod: ";
char filename[150];
cin.getline(filename, 150);

ifstream fin;
fin.open(filename);

if (!fin.is_open())
{
    cout << "ERROR. File is not oppened." << endl;
}
```

## 3. Считываем символы с файла в массив text, а когда встречаем комбинацию точка + пробел, то в массиве strs создаем экземпляр новой строки с длиной равной nowStrLenght и индексом начала в массиве text равном realTextLenght.

```
for (char symbol; realTextLenght < 1000 && fin.get(symbol);) {
    if (symbol == ' ' && text[realTextLenght - 1] == '.') {
        strs[realStrCount] = Str(nowStrStart, nowStrLenght);
        nowStrStart = realTextLenght;
        nowStrLenght = 0;
        realStrCount++;
    }
    else {
        text[realTextLenght] = symbol;
        nowStrLenght++;
        realTextLenght++;
    }
}
```

4. В последнем предложении нет пробела после точки , но её тоже нужно взять. Поэтому, если nowStrLenght больше нуля мы записываем последнюю строку в массив Strs

```
if (nowStrLenght > 0)
    strs[realStrCount++] = Str(nowStrStart, nowStrLenght);
```

5. Далее мы сортируем предложения по их длине с помощью пузырьковой сортировки

```
for (int i = 0; i < realStrCount; i++) {
    for (int j = i + 1; j < realStrCount; j++) {
        if (strs[i].Lenght > strs[j].Lenght) {
            Str buf = strs[i];
            strs[i] = strs[j];
            strs[j] = buf;
        }
    }
}
```

6. В конце мы выводим в консоль или записываем в файл, который указали для вывода (В зависимости от флага), весь массив text, после каждого предложения переходя на новую строку.

```
cout << "Entered file vivod: ";
char filename2[150];
cin.getline(filename2, 150);
```

```
ofstream OUT;
OUT.open(filename2);
if (!OUT.is_open()) {
    cout << "ERROR. File is not opened" << endl;
}
```

```
else
{
    for (int i = 0; i < realStrCount; i++, OUT << endl) {
        for (int iText = strs[i].iStart; iText < strs[i].iStart + strs[i].Lenght;
iText++) {
            OUT << text[iText];
        }
    }
    OUT.close();
    return 0;
}
```

Всего таких три функции, они вызываются в зависимости от флага. Отличаются только вводом текста из файла заданным пользователем или заданным в программе и выводом ответа в файл заданным пользователем или в консоль

## Makefile:

```
NN = main
```

```
CC = g++
```

```
$(NN): main.o So.o
```

```
$(CC) main.o So.o -o $(NN)
```

```
main.o: main.cpp
```

```
$(CC) -c main.cpp -o main.o
```

```
functions.o: functions.cpp
```

```
$(CC) -c So.cpp -o So.o
```

```
clean:
```

```
rm -rf $(NN) *.o
```

```
distclean: clean
```

```
rm -f make
```