

## Семестр 3, Контрольная работа 2 (вариант 1)

Требуется разработать класс `List` (ссылочная реализация одностороннего списка целых чисел), класс `Iterator` (итератор для списка `List`), функцию `run` (обработчик числовой последовательности, хранящейся в списке `List`).

### Описание класса `List`

Конструктор по умолчанию создает пустой список.

С помощью переопределенной операции `+` в конец списка добавляется новый элемент.

Метод `begin` возвращает итератор на первый элемент списка. Метод `end` возвращает итератор на правую границу списка.

### Описание класса `Iterator`

С помощью переопределенной унарной операции разыменования `*` осуществляется доступ к текущему элементу массива.

С помощью переопределенной унарной операции `++` итератор сдвигается по списку на одну позицию вправо.

### Функция `run`

Эта функция имеет следующий прототип

```
double run(Iterator l, Iterator r);
```

Входные параметры `l` и `r` задают фрагмент последовательности, хранящейся в списке, который подвергается обработке. Обработка заключается в вычислении некоторой характеристики этой подпоследовательности.

Итератор `l` указывает на первый элемент подпоследовательности, а итератор `r` указывает на элемент, следующий за последним элементом подпоследовательности.

Например, если в списке хранится последовательность 1 2 3 4 5 6 7 8 9, итератор `l` указывает на элемент со значением 3, итератор `r` указывает на элемент со значением 8, то обработке подвергается фрагмент 3 4 5 6 7.

В частности, в следующем фрагменте обрабатывается вся числовая последовательность, хранящаяся в списке `L`, а вычисленная характеристика сохраняется в переменную `x`.

```
List L;  
...  
double x = run(L.begin(), L.end());
```

### Программа

Корректными входными данными программы является числовая последовательность (возможно пустая), записанная в файл `input.txt`.

Программа создает экземпляр класса `List`, добавляет в него числовую последовательность, прочитанную из файла `input.txt`.

С помощью функции `run` вычисляется характеристика этой последовательности, которая печатается в стандартный поток вывода.

Например,

```
$ echo "1 2 3 4 5" > input.txt  
$ ./prog  
3.000000
```

### Характеристика последовательности

Характеристикой непустой числовой последовательности является среднее арифметическое ее элементов.

Пустая последовательность имеет характеристику 0.