

# Задание для второго отборочного этапа олимпиады «IT-планета», март 2016.

## Конкурс «Программирование: C# и ASP.NET MVC»

### Система оценивания

За первую задачу можно получить от 0 до 10 баллов в зависимости от количества выполненных требований.

При проверке остальных задач, для каждой из них будет отобрано 10% лучших по скорингу решений. В случае близости решений, прошедших и непрошедших 10% барьер, близкие к прошедшим решениям также проходят дальше. Все не прошедшие в 10% решения получают 0, все прошедшие - от 5 до 7 баллов в зависимости от набранных очков, плюс еще от 0 до 3 баллов по результату проверки стиля кода.

### Задачи

#### Задание 1

Контур-146 - космический модуль, который исследует бескрайний космос. Как-то раз он нашел планету, на которой проживает никому неизвестный народ. На переговоры с инопланетянами был отправлен космонавт-стажер Петя и его наставник Василий Иванович. Во время общения с ними, Петя узнаёт, что это очень развитая цивилизация, но она существует без рациональных чисел. Как это вообще возможно? Помоги Пете обучить своих новых друзей рациональным числам.

В пространстве имен System создать тип Rational, соответствующий рациональному числу. Ration должен:

- быть типом-значением
- содержать два доступных для публичного чтения члена Numerator и Denominator типа int.
- реализовывать паттерн immutable
- хранить дробь в несокращенном виде

Инициализация:

- `new Rational(2,3)` должен создавать объект, равный  $\frac{2}{3}$ .
- в случае некорректных параметров, конструктор должен выбрасывать `ArgumentException`.

Пусть a,b,c - объекты Rational. Тогда:

- `a.Simplify()` должен возвращать `Rational`, полученный из исходного путем сокращения дроби по математическим правилам. Знаменатель дроби в результате должен быть положительным и наименьшим из возможных.
- `a.ToString()` должен возвращать строку вида `"2/3"`, `"-2/4"`, `"1/-2"`, `"-2/-2"`. В случае, если числитель равен нулю, либо знаменатель равен единице, строка должна иметь вид `"0"`, `"42"`, и т.д.
- `a+b`, `a*b`, `a-b`, `a/b` и `-a` должны возвращать `Rational`, являющийся **сокращенной** суммой, произведением, разностью, частным и обратным по сложению элементом аргументов. В случае, если числитель или знаменатель результирующей дроби не лежит в диапазоне `int`, должен выбрасываться `OverflowException`
- `a<b`, `a<=b`, `a==b`, `a!=b`, `a>=b`, `a>b` должны работать корректно
- должно присутствовать неявное преобразование в тип `double`, а также явное преобразование из типа `int`.

Взаимодействия с коллекциями:

- Метод `List<Rational>.Sort()` должен работать правильно
- При работе с `HashSet<Rational>` и `Dictionary<Rational,T>`, объекты типа `Rational` должны считаться одинаковыми, если они равны в математическом смысле

## Задание 2

Поймав сигнал о помощи от "Народа X", космические путешественники, Петька и Василий Иванович, решили пригласить терпящих бедствие на свой корабль. Однако вскоре выяснилось, что модуль, на котором летели инопланетяне не может пристыковаться к кораблю из-за кого-то программного сбоя. Стажер Петька быстро понял, в чём проблема. Он написал программу на языке C#, используя только лямбда-выражения и отправил ее попавшим в беду. Но его опытный наставник, Василий Иванович, знал, что компьютеры "Народа X" понимают только ассемблер и попросил Петьку сделать так:

В пространстве имен `LittleAssembler` разработать класс `Translator`, статический метод которого `Translate` переводит лямбда-выражение на языке C# в последовательность команд на псевдоассемблере, вычисляющую ту же функцию.

Вход:

```
List<string> result = LittleAssembler.Translator.Translate( (x,y,z) => (x*y + z) / x + 1 );
```

В выражении будут использоваться только операторы сложения, умножения, деления, вычитания, унарного минуса и скобки. Количество аргументов не более 10. Тип аргументов и всех значений в настоящей задаче - `int`, деление понимается как целочисленное деление.

Выход: `List<string>`, последовательность строк вида

```
MOV [Memory] [Register]
MOV [Register] [Memory]
```

ADD [Register] [Register]  
SUB [Register] [Register]  
MUL [Register] [Register]  
DIV [Register] [Register]  
MOV [Memory] [Constant]

где [Register] - A, B, C или D, а [Memory] - целое число, означающее номер ячейки памяти, [Constant] - целочисленная константа. Размер памяти не ограничен.

Правила работы псевдоассемблера:

Перед запуском аргументы размещены в памяти в ячейках 0, 1, 2 и т.д. по порядку, соответствующую их следованию в лямбда-выражении. После этого последовательно вызываются команды. Команда MOV переносит значение из первого аргумента во второй (из памяти в регистр или наоборот). Обратите внимание, что команда MOV не может переносить константу напрямую в регистр, только в память. Команды ADD, SUB, MUL и DIV выполняют соответствующую математическую операцию над переданными регистрами, ее результат кладут в первый переданный регистр, а второй оставляют без именованности.

Можно считать, что аргументы малы и ошибка переполнения возникнуть не может. В случае попытки деления на ноль выполнение программы заканчивается, и исполняющая среда входит в состояние ошибки.

Ваша задача: написать программу, которая генерирует по лямбда-выражению оптимальную ассемблерную программу, вычисляющую это выражение и размещающую вычисленное значение в ячейке памяти 0. Расчет оптимальности ведется следующим образом: плата за каждую использованную ячейку памяти составляет 113 единиц, за каждую операцию - 7 единиц, и вам необходимо минимизировать суммарную стоимость программы.

## Задание 3

Когда-то компания СКБ Контур была маленькой, никому неизвестной. Потом она стала быстро развиваться и открывать филиалы в разных городах России. Необходимость рассылать по всей стране сувенирную продукцию, рекламные материалы и другие предметы усложнила работу многим людям. Тогда было решено организовать отдел логистики и сделать систему, которая позволит работать с заказами от всех подразделений компании. Ниже вы найдете небольшой фрагмент задачи, которая досталась студенту во время летней стажировки когда-то давным-давно. Попробуй и ты решить эту задачу.

Даны два csv-файла `dispatches.txt` и `wares.txt` в кодировке UTF-8. Первый содержит информацию об отгрузках товаров со склада, второй - описание товаров, входящих в каталог. Образцы файлов:

`dispatches.txt`

|   |
|---|
| <code>id,ware_id,start_time,end_time</code> |
|---|

```
51131,2123,31.01.2016 12:34:54.432,31.01.2016 13:45:21.311
```

wares.txt

```
ware_id,ware_name  
2123,Кружка
```

Вам необходимо вычислить среднее значение, среднеквадратичное отклонение, и максимум времени обработки в минутах одного заказа для каждого товара, и отсортировать товары по убыванию математического ожидания. Результат вывести в файл report.txt в формате csv и кодировке UTF-8 в следующем формате:

report.txt

```
id,ware_name,mean,deviation,max  
2123,Кружка,20.542,0.032,34.234  
3233,Флешка,1.892,0.100,5.000  
5613,Буклет,0.563,0.321,6.421
```

Обратите внимание на формат чисел с плавающей точкой. Данный файл report.txt не является правильным выходным файлом для представленных dispatches.txt и wares.txt

Ограничения:

1. Файл dispatches.txt имеет большой размер (несколько гигабайт) и заведомо не помещается в оперативную память
2. Файл wares.txt имеет небольшой размер, не более десяти мегабайт
3. Оба файла будут размещены в том же каталоге, в котором будет лежать программа.
4. Программа будет запущена на машине с несколькими процессорными ядрами

Цель: вывести указанный csv файл не только правильно, но максимально быстро.

## Задание 4

Петя подготовил доклад к финальной конференции, которая завершает ежегодную летнюю стажировку в Контуре. Распечатал текст, начал репетировать речь и вдруг понял, что часто повторяет некоторые слова. Доклад получился очень большим, поэтому Петя решил написать программу, которая найдет эти часто повторяющиеся слова в докладе. Помогите Пете.

В одном файле — некое литературное произведение. В другом — в каждой строке поисковый запрос.

Вам нужно для каждого запроса вывести номера всех строк исходного произведения, в которых содержатся все слова из поискового запроса. Если таких строк слишком много, выведите 20 самых первых по порядку. Для каждого запроса номера строк нужно выводить в порядке их возрастания.

## Детали

Имена файлов — с произведением и с поисковыми запросами будут передаваться вашей программе в качестве аргументов командной строки именно в таком порядке. Оба файла будут в формате простого текста в кодировке UTF-8, перевод строк windows-style.

Объем литературного произведения - от 1 до 2 МБ. Файл с запросами не превосходит 1 МБ.

Слово может состоять из русских или латинских букв или знака подчеркивания или цифр. Сравнивать слова нужно регистронезависимо.

Нумерация строк осуществляется с нуля.

Результат ваша программа должна напечатать на консоль, по одной строке на каждый запрос. В каждой такой строке — номера строк исходного произведения, разделенные запятой.

Пример ниже должен прояснить все тонкие моменты в условии:

Файл текста:

```
Foo foo bar
Bazz foo bar
foobar
```

Файл запроса:

```
bar
foobar
XYZ
foo
baz
FOO FOO FOO FOO
```

Ожидаемый выход программы:

```
0,1
```

2

0,1

0,1

Цель: вывести результат не только правильно, но максимально быстро.

## Задание 5

Однажды наши космонавты решили доставить “Народу X” гуманитарную помощь. Василий Иванович недавно посмотрел фильм “Марсианин” и, чтобы не получилось, как в фильме, дал Петьке задание оперативно и максимально равномерно распределить груз между носовым и хвостовым отсеками грузового корабля. Это если кратко. А на самом деле формулировка была такая:

Есть достаточно большое количество натуральных чисел, нужно разбить их на два подмножества так, чтобы модуль разности сумм элементов в каждом подмножестве был минимальным. При этом нужно уложиться точно в отведенное время.

Формат входа:

[Целое количество миллисекунд, которые вы можете потратить на задание]  
[Количество чисел в множестве]  
[Элемент множества 1]  
[Элемент множества 2]  
[Элемент множества 3]

Формат выхода

[Количество элементов в первом подмножестве]  
[Элемент первого подмножества 1]  
[Элемент первого подмножества 2]  
[Элемент первого подмножества 3]

Порядок элементов подмножества произволен и может не совпадать с порядком элементов в исходном множестве.

Вы можете считать, что сумма всех исходных чисел укладывается в формат long

Ваша программа обязана точно уложиться в указанное время, в противном случае ее результат не засчитывается. Программы, уложившиеся в указанное время, набирают очки в зависимости от того, насколько мал модуль разности сумм между подмножествами.

Отсчет времени начинается с момента запуска вашей программы, и ведется с использованием Stopwatch. Проведите эксперименты на вашем компьютере самостоятельно, чтобы убедиться, что ограничение времени соблюдается.

Пример входного файла

Формат входа:

```
1000
4
40
30
20
50
```

Формат выхода

```
2
40
30
```

## Оформление результатов

Результатом выполнения задания должен быть zip-архив, имеющий следующую внутреннюю структуру

```
.
  name.txt
  Solution1.cs
  Solution2.cs
  Solution3.cs
  Solution4.cs
  Solution5.cs
```

Файл name.txt должен содержать почтовый адрес, использованный для регистрации на соревнованиях.

Файлы Solution1.cs,..., Solution5.cs должны содержать весь код, необходимый для выполнения соответствующих заданий. Мы рекомендуем вам дополнительно написать небольшую утилиту, которая бы собирала код, написанный для выполнения задания, в один cs-файл.

Вам следует использовать язык C# версии 4.0 для платформы .NET Framework 4.5. При создании кода проверьте версию языка и платформы.