

## ДЕНЬ 2Е. Работа с файлами

### 2Е.1. Генератор записей CSV-файла

Пользователь вводит, сколько случайных записей он хочет получить, и имя CSV-файла, куда их необходимо сохранить. (CSV – текстовый файл с табличными данными, в которых каждая строка — это строка таблицы, а колонки разделены запятыми или точкой с запятой)

Программа должна сгенерировать заданное число записей, выбрав для каждой из них случайный формат из списка поддерживаемых форматов согласно варианту, а затем сохранить их в CSV-файл. Каждая запись должна состоять из трех колонок:

- 1) номер по порядку;
- 2) номер выбранного шаблона;
- 3) сгенерированная строка.

Для случайного выбора используйте класс `Random`.

Формат описывается следующими спец. символами:

- A – любая гласная в верхнем регистре
- B – любая согласная в верхнем регистре
- X – любая буква в верхнем регистре
- a – любая гласная в нижнем регистре
- b – любая согласная в нижнем регистре
- x – любая буква в нижнем регистре
- 9 – любая цифра
- ? – предыдущий символ может быть опущен

Любые другие символы интерпретируются, как есть. Например, формат «Abc-999?», означает: сначала любая гласная в верхнем регистре, потом любая согласная в нижнем регистре, затем символ «с», затем символ «-», а далее две или три цифры: Aгс-42

#### Пример

Форматы: 99, XX?X?, 009

Введите число записей:  
> 6

Введите путь до файла:  
> result.csv

*Выходной файл:*

```
1, 1, 12
2, 2, T
3, 2, RS
4, 3, 007
5, 3, 001
6, 1, 64
```

### Варианты

№	Форматы		
1	Ab?abab	XXX-999	B?B?B?AAA
2	A9A-XX-ха?9	х?XB	9-9?-9?
3	A---B-?-?-?A---B	999?9?9?	Xaaaaaaa
4	хABx9	baBB9(Xb)AAx	9-?9-?9
5	T?A9aBbA	b(BXX)Ab	bbb aaa BBB AAA
6	Bab?aba?	999-999	XxxxxxX
7	9b?aB9aBAx	A9?b9bAaX	9A 9B 9x
8	aXB	A? AAb? A	ab?ab?ab?ab?
9	M?aaxB	х9а9аX9а	+7 (123) 999-9999
10	aB?bAbx?	99X9	b(a9a9a9a)b
11	Aba:bXX:aaAA	9ab? а? Xa	XXX-xxx-AAA-bbb
12	8(999)999	XxXxXxX	9?9?9?AABB
13	Xba	B9XABA?	123-999-567
14	R999999?9?	AxxxxxxxxB	999A?999
15	A9X	BX?bBAB	99?9?AA?A?BB?B?
16	9xxXxxB	х-В?А	999 999 999 999
17	9X(AAB)XA	BxA	abab 999
18	(?3412)?999-999	AAAB?B?B?	xxxxxxxx
19	AX[AA9X]Bx	B99-b?xb	XXXxxx999
20	AbB9	XXX AAb XBВ	0?0?0?999

### 2Е.2. Калькулятор цены доставки, считывающий данные из файла

Необходимо разработать программу расчета стоимости доставки нескольких посылок по одному адресу.

Пользователь вводит координаты доставки (х, у) и путь до CSV-файла со списком посылок (CSV – текстовый файл с табличными данными, в которых каждая строка — это строка таблицы, а колонки разделены запятыми или точкой с запятой).

Каждая посылка характеризуется весом ( $w_i$ ) и объемом ( $v_i$ ). Во входном CSV-файле первой строкой идет заголовок таблицы, по которому определяется порядок колонок (сначала вес, обозначаемый  $w$ , или сначала объем, обозначаемый  $v$ ), а затем параметры каждой из посылок

После ввода программа выдает результат, округленный до целых, и прекращает выполнение. Формула расчета стоимости приведена в таблице вариантов.

Обозначения:  $\sum$  - сумма элементов,  $\max$  – максимальный элемент,  $\min$  – минимальный элемент,  $\sqrt{\phantom{x}}$  – квадратный корень,  $|z|$  - взятие модуля числа

#### Пример

Формула:  $\sum w_i + \max(v_i) + x + y$

Входной файл *boxes.csv*:

```
v, w
4.5, 10
9.27, 20
```

40.3, 30

**Программа:**

Введите координаты доставки (X)  
> 100

Введите координаты доставки (Y)  
> 200

Введите путь до файла с посылками  
> boxes.csv

Результат: 400

**Варианты**

№	Формула	№	Формула	№	Формула
1	$\sqrt{x^2+y^2} * \sum v_i / 95$	9	$( x + y ) * \sum v_i * 0.123$	17	$\sqrt{x * y} * \sum v_i * 0.3$
2	$\sum w_i * \max(v_i) +  x-y $	10	$\sum w_i * \sum v_i +  x - y $	18	$ x - y  * \max(v_i) * 0.2$
3	$\sqrt{x^2+y^2} + \sum w_i * \sum v_i$	11	$\sum (v_i * w_i) * \sqrt{x^2+y^2}$	19	$\max(v_i/w_i) *  x-y $
4	$\max(w_i) * \max(v_i) +  x - y $	12	$\sqrt{x^2+y^2} * \max(w_i) / 8$	20	$( x + y ) * \sum v_i * \sum w_i$
5	$ x-y  * \min(v_i^2) / \max(w_i)$	13	$\sqrt{x * y} * \max(v_i) * 0.7$		
6	$( x + y ) * \sum v_i / \min(w_i)$	14	$\sqrt{x^2+y^2} * \sum w_i * 0.1$		
7	$\sum w_i / \max(v_i) *  x-y $	15	$( x + y ) * \max(v_i^2 + w_i^2)$		
8	$ x - y  / \sqrt{x^2+y^2} * \sum v_i$	16	$\sqrt{x^2+y^2} * \sum v_i * \sum w_i / 3$		

**2Е.3. Прочитать все файлы из указанной директории и сформировать сводный CSV-файл**

Пользователь указывает путь до директории с исходными CSV файлами и путь до файла с результатом. Необходимо построить и сохранить CSV-файл отчет, согласно заданию. В отчете должен быть результат по каждому из входных файлов и общий результат.

Для получения списка файлов, см. *System.IO.Directory.GetFiles*

**Пример**

Запись: фамилия (строка), возраст (целое число), результат (вещественное число).  
Вывести среднее значение результата.

*Входной файл jan.csv:*

Петров, 20, 5.3  
Иванов, 22, 7.6  
Сидоров, 20, 5

*Входной файл feb.csv:*

Потемкин, 19, 2.3  
Пушкин, 32, 9.6

*Входной файл mar.csv:*

Журавлев, 18, 7.3  
Никитин, 33, 8.6

Жаров, 22, 3.5  
Семечкин, 55, 4

### Программа:

Введите путь к директории:  
> data/dir/

Введите путь для сохранения файла:  
> report.csv

### Выходной файл:

jan.csv, 5.97  
feb.csv, 5.95  
mar.csv, 5.85  
Общее, 5.91

### Варианты

№	Запись	Отчет
1	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)	Среднее значение <i>среднего балла</i>
2	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)	Сумма разниц между <i>суммами поступления и списания</i>
3	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)	Сумма разниц между <i>планом выпуска деталей и фактическим выпуском</i>
4	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)	Среднее значение <i>всех оценок</i>
5	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)	Сумма значений <i>пробега</i>
6	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)	Сумма <i>балансов</i>
7	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)	Среднее <i>число слов</i>
8	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)	Среднее значение <i>средней цены</i>
9	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)	Сумма <i>весов</i>
10	Дисциплина (строка), номер курса (целое число), количество часов (целое число)	Среднее <i>количество часов</i>
11	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)	Среднее число <i>выполненных заданий</i>
12	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)	Среднее значение <i>номера этажа</i>
13	Фамилия (строка), рост (вещественное число), вес (вещественное число)	Среднее значение <i>роста</i>

14	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)	Сумма разниц между <i>числом товара на складе</i> и <i>числом зарезервированных товаров</i>
15	Название материала (строка), объем (вещественное число), вес (вещественное число)	Сумма <i>объемов</i>
16	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)	Среднее <i>число сезонов</i>
17	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)	Среднее значение <i>гонорара</i>
18	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)	Сумма <i>сумм заказов</i>
19	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)	Среднее <i>число уникальных посетителей</i>
20	Фамилия (строка), должность (строка), оклад в руб (целое число)	Сумма <i>окладов</i>

## 2Е.4. Обработка файла с учетом аргументов командной строки

Вам необходимо реализовать программу, принимающую входную информацию через аргументы командной строки.

Пример запуска программы:

```
program.exe input.txt output.txt -from 5 -to 8
```

Аргументы:

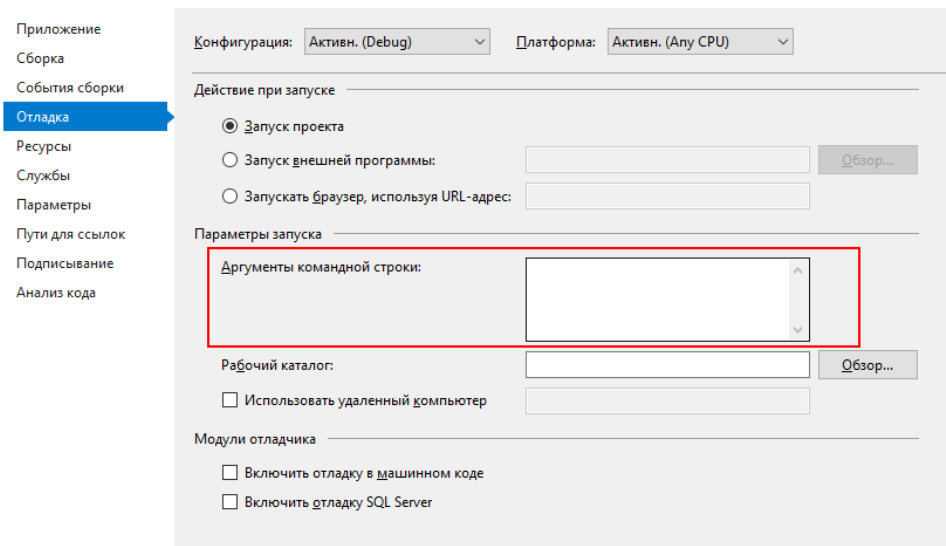
- *input.txt* – входной файл, который нужно обработать. Обязательный аргумент. Если не задан, то выдать сообщение «Укажите входной файл» и завершить работу
- *output.txt* – выходной файл, куда нужно сохранить результат. Если аргумент не передан, то результат нужно вывести в консоль. Этот аргумент всегда идет после аргумента с входным файлом
- *-from 5* – с какой строки файла начинать обработку. Если не задан, то обработка начинается с начала файла
- *-to 8* – до какой строки файла, включительно, проводить обработку. Если не задан, то обработка производится до конца файла

Порядок аргументов может быть произвольным

Получить значение аргументов можно в главной функции программы:

```
static void Main(string[] args) // args – аргументы
```

Для тестирования указать аргументы командной строки можно в настройках проекта:



## Пример

Обработка: удаление каждого второго слова

Файл *poem.txt*

В томленьях грусти безнадежной  
В тревогах шумной суеты,  
Звучал мне долго голос нежный  
И снились милые черты.

Шли годы. Бурь порыв мятежный  
Рассеял прежние мечты,  
И я забыл твой голос нежный,  
Твои небесные черты.

Запуск программы:

```
program.exe -from 2 poem.txt res.txt -to 6
```

После запуска программы должен создаться *res.txt*:

В шумной,  
Звучал долго нежный  
И милые

Шли Бурь мятежный

## Варианты

№	Преобразование	№	Преобразование
1	Удалить все слова без гласных букв	11	Удалить все гласные
2	Удалить все слова размером меньше, чем из 5 букв	12	Удалить знаки препинания
3	Удалить все слова без согласных букв	13	Перевернуть каждое слово в строке (абг деж -> гба жед)
4	Перевести в верхний регистр все гласные	14	Перевернуть строки (абг деж -> жед гба)

5	Перевести в верхний регистр все согласные	15	Перевести все буквы первого слова каждой строке в верхний регистр
6	Перемешать в случайном порядке все слова в строке	16	Удалить все слова размером больше, чем из 5 букв
7	Удалить первое и последнее слова в строках	17	Поменять местами слова в каждой паре слов: аб вг де жз -> вг аб жз де
8	Перевести первую букву каждого слова в верхний регистр	18	Удалить все согласные
9	Удалить все слова, в которых больше 5 согласных	19	Удалить все слова, в которых меньше 5 согласных
10	Удаление каждого второго слова	20	Перевести в верхний регистр каждую вторую букву слов: абв гдеж -> аБв гДеЖ

## 2Е.5. Интерактивный выбор обрабатываемой строки текстового файла

Программа должна запросить путь к входному файлу, а затем очистить экран и вывести на него содержимое указанного файла.

Если строки файла не помещаются по ширине, то при выводе они должны обрезаться. Если файл не помещается по высоте, то он выводится постранично. Для того, чтобы узнать размер окна используйте *Console.WindowWidth* и *Console.WindowHeight*

При постраничном выводе предпоследняя строка экрана должна быть пустой, а последняя – содержать номер текущей страницы и общее число страниц в формате:  
Страница 2 из 4

Одна из строк (при запуске – первая) является активной. Активная строка подсвечивается желтым цветом (используйте *Console.BackgroundColor*). Смена активной строки осуществляется с помощью стрелок клавиатуры.

Нажатие стрелки «Вниз» делает активной следующую строку. Если текущей строкой была последняя строка страницы, то должна отобразиться следующая страница (если есть) и активной стать первая строка.

Нажатие стрелки «Вверх» делает активной предыдущую строку. Если текущей строкой была первая строка страницы, то должна отобразиться предыдущая страница (если есть) и активной стать последняя строка.

Нажатие кнопки «Вправо» переключает файл на следующую страницу (если есть), активной становится первая строка.

Нажатие кнопки «Влево» переключает файл на предыдущую страницу (если есть), активной становится последняя строка

Нажатие кнопки «Пробел» преобразует текущую активную строку согласно варианту. Важно отметить, что преобразовываться должна вся строка целиком, а не только ее обрезанная при выводе часть. Если пользователь нажимает «Пробел» вновь, то строка

должна быть преобразована вновь. Сохранять выполненное преобразование обратно в файл не нужно.

Нажатие кнопки «Esc» завершает работу программы

### Варианты

№	Преобразование	№	Преобразование
1	Перевести в верхний регистр все согласные	11	Удалить все согласные
2	Удалить все слова без гласных букв	12	Удаление каждого второго слова
3	Перевести в верхний регистр все гласные	13	Перевернуть каждое слово в строке (абг деж -> гба жед)
4	Удалить все слова, в которых меньше 5 согласных	14	Перевести первую букву каждого слова в верхний регистр
5	Перевернуть строки (абг деж -> жед гба)	15	Удалить все слова размером меньше, чем из 5 букв
6	Удалить первое и последнее слова в строках	16	Удалить все слова без согласных букв
7	Перемешать в случайном порядке все слова в строке	17	Удалить знаки препинания
8	Перевести все буквы первого слова каждой строке в верхний регистр	18	Поменять местами слова в каждой паре слов: аб вг де жз -> вг аб жз де
9	Удалить все гласные	19	Удалить все слова размером больше, чем из 5 букв
10	Перевести в верхний регистр каждую вторую букву слов: абв гдеж -> аБв гДеЖ	20	Удалить все слова, в которых больше 5 согласных