

ДЕНЬ 2А. Работа с файлами

2А.1. Прочитать CSV-файл и вывести записи, удовлетворяющие условию

Программа запрашивает путь к файлу формата CSV (табличные данные, в которых каждая строка — это строка таблицы, а колонки разделены запятыми или точкой с запятой). Необходимо вывести из файла только те записи, которые удовлетворяют заданному условию

Пример

Запись: фамилия (строка), возраст (целое число), результат (вещественное число).
Вывести записи, у которых возраст меньше 21

Входной файл:

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

Программа:

```
Введите путь к файлу:
> data.csv
```

```
Фамилия      | Возраст | Результат
-----
Петров       | 20      | 5.3
Сидоров      | 20      | 5
```

Варианты

№	Запись	Условие
1	Фамилия (строка), номер группы (строка), номер в группе (целое число), число выполненных заданий (целое число)	Число выполненных заданий ≥ 7
2	Название (строка), число сезонов (целое число), год выпуска первого сезона (целое число)	Число сезонов > 2
3	Номер телефона (строка), имя оператора (строка), баланс в копейках (целое число)	Баланс < 0
4	Название цеха (строка), план выпуска деталей (целое число), фактический выпуск деталей (целое число)	Значение плана выпуска деталей больше значения фактического выпуска деталей
5	Компания (строка), сумма поступлений в млн. руб. (вещественное число), сумма списаний в млн. руб. (вещественное число)	Разница между суммами поступления и списания > 0
6	Адрес отправления (строка), адрес доставки (строка), вес (вещественное число)	Вес > 500
7	Название товара (строка), количество на складе (целое число), количество зарезервированных (целое число)	Количество на складе $<$ количества зарезервированных
8	Фамилия (строка), оценка за теорию (целое число), оценка за практику (целое число)	Среднее между оценками за теорию и практику ≥ 3.5

9	Дисциплина (строка), номер курса (целое число), количество часов (целое число)	Количество часов < 40
10	Фамилия (строка), число ролей (целое число), гонорар в млн. руб. (вещественное число)	Гонорар >= 1
11	Название материала (строка), объем (вещественное число), вес (вещественное число)	Объем < 10
12	Город (строка), улица (строка), номер дома (число), номер этажа (целое число)	Номер этажа > 2
13	Адрес сайта (строка), число посетителей (целое число), число уникальных посетителей (целое число)	Число уникальных посетителей > 1000
14	Автомобильный номер (строка), год выпуска (целое число), пробег в км (целое число)	Пробег < 10000
15	Производитель (строка), объем выпуска (вещественное число), средняя цена (вещественное число)	Средняя цена > 100
16	Тема письма (строка), адресат (строка), есть ли вложения (логический тип), число слов (целое число)	Число слов > 200
17	Фамилия (строка), год поступления (целое число), средний балл (вещественное число)	Средний балл >= 4
18	Фамилия (строка), рост (вещественное число), вес (вещественное число)	Рост > 200
19	Фамилия (строка), должность (строка), оклад в руб (целое число)	Оклад > 25000
20	Номер заказа (строка), описание (строка), выполнен или нет (логический тип), сумма заказа (целое число)	Сумма заказа > 800

2А.2. Шаблонизатор писем, работающий с файлами

Пользователь вводит несколько строк шаблона письма, который может содержать специальные маркеры – участки текста, которые нужно будет заменить заданным содержимым. Чтобы завершить ввод шаблона, пользователь должен ввести пустую строку.

После ввода шаблона пользователь указывает, если требуется, значения маркеров, общих для всех писем.

Далее пользователь указывает путь до CSV-файла с данными для генерации писем, а затем путь до папки, куда нужно сохранить письма. В CSV-файле в первой колонке хранится имя файла, куда нужно записать текст письма, а в остальных – данные маркеров, уникальных для каждого из писем (порядок этих колонок выбираете сами)

После ввода всех данных, программа должна сгенерировать столько писем, сколько строк в CSV-файле, подставив в шаблон необходимые значения вместо маркеров, и далее сохранить каждое письмо в отдельный файл. Важно, что один и тот же маркер может присутствовать в шаблоне несколько раз

В зависимости от варианта программа должна поддерживать несколько маркеров из следующего списка:

- %USERNAME% - берется из CSV-файла
- %FIRSTNAME% - берется из CSV-файла
- %LASTNAME% - берется из CSV-файла
- %COUPON% - берется из CSV-файла
- %ORDERNUM% - берется из CSV-файла
- %SUM% - берется из CSV-файла
- %PRODUCTNAME% - берется из CSV-файла
- %AGE% - берется из CSV-файла
- %LANGUAGE% - берется из CSV-файла
- %COUNTRY% - берется из CSV-файла
- %COMPANY% - вводит пользователь один раз для всех писем
- %ADDRESS% - вводит пользователь один раз для всех писем
- %EMAIL% - вводит пользователь один раз для всех писем
- %YEAR% - текущий год
- %DATE% - текущая дата в формате DD.MM.YYYY
- %MONTH% - название текущего месяца (на русском или английском)
- %REPEAT% - повтор предыдущего слова
- %RAND% - случайное число от 1 до 100 (используйте класс Random)
- %LINE% - разрыв строки и вставка разделительной линии (10 символов тире)

Пример

Маркеры: %FIRSTNAME%, %LASTNAME%, %COMPANY%, %RAND%

Введите шаблон:

```
> Dear %FIRSTNAME% %LASTNAME%!
> You won ONE BILLION DOLLARS. It's %RAND%% true!
> Best regards,
> Your %COMPANY%
>
```

Введите значение COMPANY:

```
> Honest Ltd.
```

Введите путь до CSV-файла с данными:

```
> data.csv
```

Введите папку для сохранения результат:

```
> folder
```

CSV-файл с данными data.csv:

```
a.txt, Ivan, Ivanov
b.txt, John, Doe
```

Результирующий файл folder/a.txt:

```
Dear Ivan Ivanov!
You won ONE BILLION DOLLARS. It's 42% true!
Best regards,
Your Honest Ltd.
```

Результирующий файл *folder/b.txt*:

Dear John Doe!
You won ONE BILLION DOLLARS. It's 42% true!
Best regards,
Your Honest Ltd.

Варианты

№	Маркеры			
1	USERNAME	ADDRESS	COMPANY	RAND
2	PRODUCTNAME	COUPON	EMAIL	LINE
3	COUPON	SUM	RAND	DATE
4	USERNAME	ORDERNUM	SUM	DATE
5	COUPON	SUM	COMPANY	DATE
6	ORDERNUM	LANGUAGE	DATE	LINE
7	AGE	ORDERNUM	DATE	MONTH
8	USERNAME	PRODUCTNAME	SUM	RAND
9	FIRSTNAME	LASTNAME	COMPANY	YEAR
10	PRODUCTNAME	SUM	ADDRESS	REPEAT
11	USERNAME	LANGUAGE	MONTH	YEAR
12	FIRSTNAME	LASTNAME	EMAIL	DATE
13	AGE	COMPANY	ADDRESS	REPEAT
14	USERNAME	LANGUAGE	COUNTRY	MONTH
15	FISTNAME	COUPON	REPEAT	MONTH
16	FIRSTNAME	LASTNAME	ADDRESS	YEAR
17	ORDERNUM	SUM	RAND	REPEAT
18	USERNAME	ORDERNUM	EMAIL	LINE
19	FISTNAME	LINE	MONTH	RAND
20	USERNAME	COUPON	COMPANY	MONTH

2А.3. Прочсть 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 и 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>

2А.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:

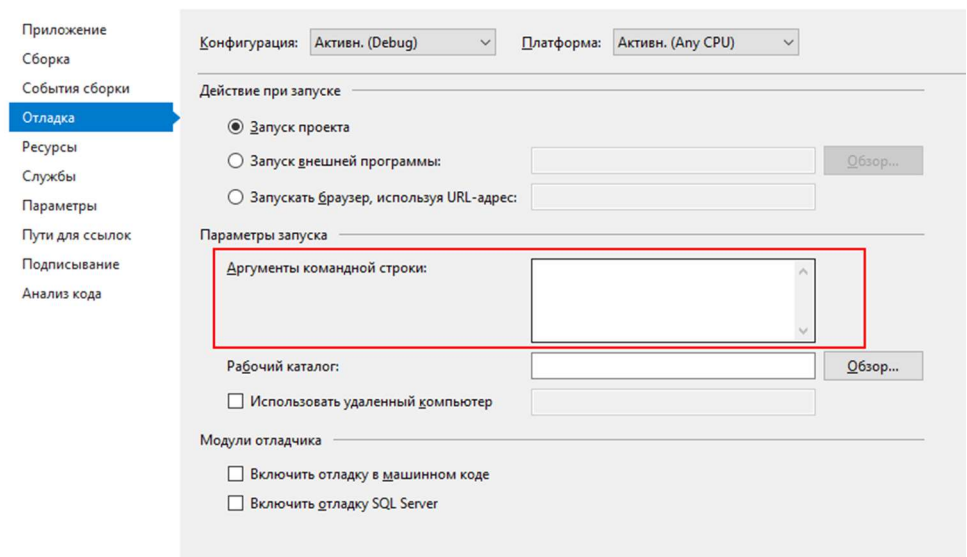
- *input.txt* – входной файл, который нужно обработать. Обязательный аргумент. Если не задан, то выдать сообщение «Укажите входной файл» и завершить работу
- *output.txt* – выходной файл, куда нужно сохранить результат. Если аргумент не передан, то результат нужно вывести в консоль. Этот аргумент всегда 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

2A.5. Отобразить BMP-файл с изображением в консоли

Пользователь указывает путь до файла с изображением размером от 8x8 до 24x24. Необходимо открыть, выполнить заданное преобразование, а затем вывести изображение в консоли, используя заданные тона. Каждый пиксел изображения при выводе должен состоять из двух пробельных символов. Цвет «пикселя» должен задаваться с помощью свойства *Console.BackgroundColor*

Для открытия и работы с изображением используйте класс *System.Drawing.Bitmap*. Подключите в проекте ссылку на стандартную библиотеку *System.Drawing.dll*, если класс недоступен.

Для получения цвета пикселя используйте *Bitmap.GetPixel(Int32, Int32)*. Для выбора отображаемого цвета, вычислите среднее значение из каналов R,G,B (яркость):

- $0 \leq \text{Яркость} < 60$ – цвет: Black
- $60 \leq \text{Яркость} < 120$ – цвет: <Цвет2>
- $120 \leq \text{Яркость} < 180$ – цвет: <Цвет1>
- $180 \leq \text{Яркость}$ – цвет: White

Пример

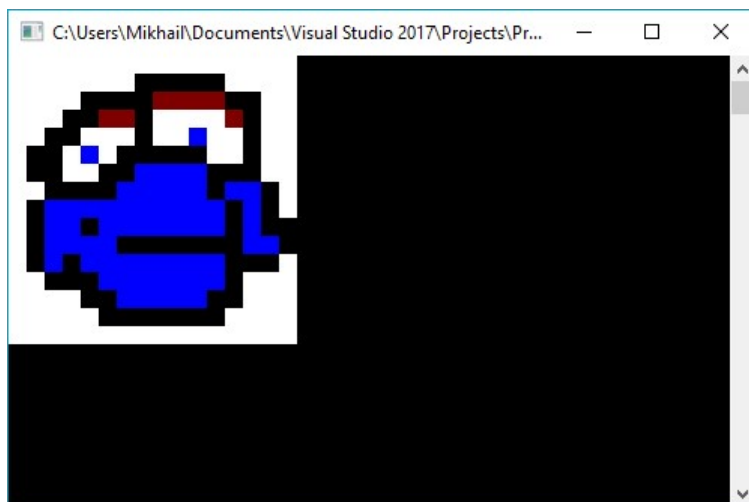
Входное изображение (16x16): 

Цвет1: Blue

Цвет2: DarkRed

Преобразование: Отражение по вертикали

Результат:



Варианты

№	Цвет 1	Цвет 2	Преобразование
1	Gray	DarkRed	Отражение по вертикали
2	Green	DarkMagenta	Поворот на 180°
3	Yellow	DarkYellow	Поворот на 180°
4	Magenta	DarkYellow	Обрезка 2 пикселей по периметру
5	Green	DarkMagenta	Уменьшение размера в два раза
6	Cyan	DarkGreen	Отражение по горизонтали
7	Blue	DarkCyan	Отражение по вертикали
8	Cyan	DarkBlue	Уменьшение размера в два раза
9	Magenta	DarkGreen	Уменьшение размера в два раза
10	Red	DarkGray	Увеличение размера в два раза
11	Cyan	DarkBlue	Увеличение размера в два раза
12	Blue	DarkRed	Отражение по вертикали
13	Magenta	DarkGreen	Обрезка 2 пикселей по периметру
14	Yellow	DarkGray	Отражение по горизонтали
15	Red	DarkGray	Поворот на 180°
16	Green	DarkYellow	Обрезка 2 пикселей по периметру
17	Gray	DarkMagenta	Отражение по горизонтали
18	Blue	DarkYellow	Отражение по вертикали
19	Red	DarkCyan	Увеличение размера в два раза
20	Gray	DarkCyan	Отражение по горизонтали