

Задача №1 - Английские числительные

Важно! В этой задаче не нужно писать программу.

Цель задания: составить набор входных и выходных данных для проверки работы программы, которая записывает числительные английского языка. Программу писать в этом задании не нужно.

Вам дана эталонная программа **spelling.exe** (без исходного кода), которая записывает число, введенное пользователем, в виде числительного на английском языке. Программа поддерживает числа в диапазоне от 0 до 99 включительно.

Вы можете ознакомиться с правилами написания английских числительных по следующей ссылке: https://en.wikipedia.org/wiki/English_numerals

Формат входных данных программы

Программа считывает в окне консоли целое число.

Формат выходных данных

Программа должна вывести введенное число в виде числительного английского языка строчными буквами.

Если было введено целое число вне диапазона от 0 до 99, либо не целое число, программа должна вывести строку ERROR заглавными буквами.

Примеры

Ввод	Вывод
8	eight
38	thirty-eight
11	eleven
abc	ERROR
999	ERROR

Указания

Изучите как работает данная вам **правильная программа** spelling.exe. Запустите её и посмотрите какие данные она выведет в ответ на вводимые данные.

Предоставьте папку с набором файлов с именами **input<N>.txt** и **output<N>.txt**, где <N> – некоторое целое число. Файлы **input<N>.txt** содержат входные данные, а **output<N>.txt** ожидаемые выходные данные.

Пример: в файле output3.txt запишите данные, которые правильная программа должна вывести, если ей подать на вход данные из файла input3.txt.

Созданные вами файлы будут использованы для проверки работы правильных и неправильных версий программы spelling.exe.

Правильная программа – та, которая выдаёт результат в соответствии с требованиями.

Неправильная программа – та, которая содержит выдаёт результат, отличный от требуемого.

Набор подготовленных вами файлов должен быть таким, чтобы:

- **все правильные программы** для каждого `input<N>.txt` вывели результат, совпадающий с `output<N>.txt`;
- **каждая из неправильных программ** для данных из хотя бы одного из входных `input<N>.txt` вывела результат, не совпадающий с `output<N>.txt`.

Задача №2 - морской бой

Ученик школы программирования Иннокентий договорился с преподавателями, что играть он будет только в те компьютерные игры, которые сам написал. Одна из любимых игр Иннокентия – “Морской бой”. Помогите Иннокентию написать программу, которая проверяет правильность расстановки кораблей для этой игры.

В игре “Морской бой” на поле 10x10 клеток располагаются корабли размером от 1 до 4 клеток. Каждая клетка игрового поля задаётся двумя координатами:

- Горизонтальная координата в виде латинских букв от A до J
- Вертикальная координата в виде чисел от 1 до 10

	A	B	C	D	E	F	G	H	I	J
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										

Поле игры “Морской бой”

Корабль может располагаться горизонтально либо вертикально. При расстановке кораблей недопустимыми будут следующие ситуации:

Выходные данные программы

Программа должна вывести строку **OK**, если все введённые пользователем корабли расставлены в соответствии с правилами. Ноль кораблей всегда считаются успешно расставленными.

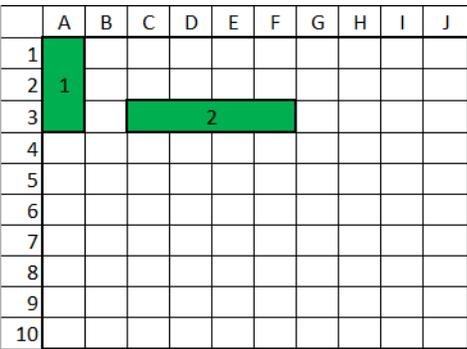
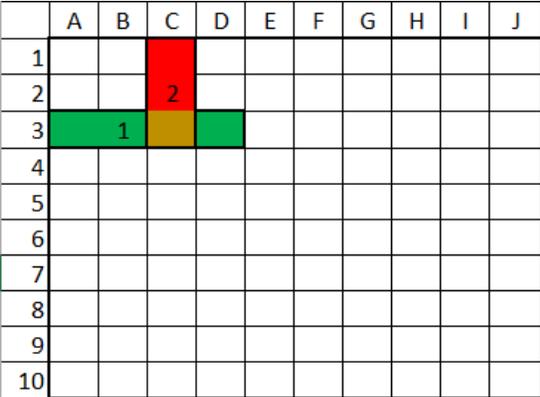
Если хотя бы один корабль установлен с нарушением правил, программа должна вывести строку **NOT OK**

Программа должна обрабатывать следующие ошибки, связанные с вводом ошибочных входных данных:

- Размер корабля выходит за пределы от 1 до 4
- Направление корабля отличается от **H** или **V**
- В качестве столбца указан символ, выходящий за пределы от 'A' до 'J'

В этих случаях программа должна вывести строку **ERROR** и завершить работу.

Примеры входных и выходных данных:

Ввод	Вывод	Пояснение
2 3 V A 1 4 H C 3	OK	
2 4 H A 3 3 V C 1	NOT OK	
1 -3 H A 8	ERROR	Указан некорректный размер корабля, равный -3
1 2 M A 1	ERROR	Указано некорректное направление корабля (M)

Заготовка функции для чтения входных данных:

В файле SeaBattle.pas вам предоставлена заготовка программы. Она умеет считывать данные, но не обрабатывает ошибки. Вам также даны файлы с тестовыми данными и скрипт для облегчения проверки вашей программы.

Задача №3 - телефонный справочник

В фирме “Рога и копыта” работают не более 100 сотрудников. В фирме ведётся телефонный справочник номеров сотрудников. У сотрудника может быть от 1 до 10 номеров. Все сотрудники имеют уникальные имена. Номер телефона - целое число, содержащее не более 11 цифр. Для его представления используйте тип **LONGINT**. Раньше телефонный справочник имел вид:

3

```
7383 Petrov Vladimir Semenovich
901 Alexeev Alexey Petrovich
89037651212 Alexeev Alexey Petrovich
```

В первой строке находится количество записей (3), в остальных - номер телефона и имя владельца. В приведённом примере у сотрудника Alexeev Alexey Petrovich два номера телефона: 901 и 89037651212.

Фирма решила обновить телефонный справочник и сделать его более удобным. В новом справочнике должны располагаться записи о каждом сотруднике с его номерами телефонов:

- Записи отсортированы в алфавитном порядке по ФИО сотрудника
- У каждого сотрудника номера телефонов отсортированы в порядке возрастания и перечислены через запятую

Например, указанный выше справочник в новом формате будет иметь вид:

```
Alexeev Alexey Petrovich: 901, 89037651212
Petrov Vladimir Semenovich: 7383
```

В этом примере номера телефонов сотрудника Alexeev Alexey Petrovich объединились в одну строку. Записи отсортированы по алфавиту.

Входные данные для программы

В первой строке содержится число **N** – количество записей не меньше 0 и не больше 500. Ноль записей в таблице не являются ошибкой, просто справочник ещё пустой.

В каждой из следующих **N** строк указаны: номер телефона, за которым через пробел следует ФИО сотрудника.

ФИО сотрудника – строка, содержащая одно или несколько слов, записанных латинскими буквами.

Один и тот же номер телефона может встречаться у нескольких сотрудников. Например, если они сидят в одном кабинете.

Программа должна обрабатывать следующие ошибки:

- Если у одного сотрудника будет больше 10 номеров телефонов
- Если в качестве номера телефона введено не число
- Если будет указано больше 100 сотрудников.

В случае этих ошибок программа должна вывести строку ERROR и завершить работу.

Выходные данные программы

В каждой выходной строке программы содержится ФИО сотрудника, затем через двоеточие и пробел перечисляются его номера телефонов. Номера телефонов разделяются запятой и пробелом и выводятся в порядке возрастания.

Необходимо соблюдать требуемый формат вывода. Программы, выводящие данные в неправильном формате не пройдут автоматические проверки и будут признаны неправильными.

Примеры входных и выходных данных:

Ввод	Вывод
4 234 Petrov Anatoly 89914003030 Ivanov Ivan 19992121211 Petrov Anatoly 456 Petrov Anatoly	Ivanov Ivan: 89914003030 Petrov Anatoly: 234, 456, 89992121211
2 123 Ivanov Ivan Abc Petrov Anatoly	ERROR
0	

Заготовка

Вам дана заготовка программы PhoneBook.pas, которую вы можете использовать для написания своей программы.