

13.1 Работа со словарями: Задание 1

Напишите программу, которая будет превращать натуральное число (в диапазоне от 0 до 9) в строку, заменяя все цифры в числе на слова. *0 на ноль, 1 на один, 2 на два* и т.д.

Входные данные:

```
170435
6 8 0 2 9
56gfd24462
```

Ввод/Вывод:

```
170435
один семь ноль четыре три пять
Process finished with exit code 0
```

```
6 8 0 2 9
шесть восемь ноль два девять
Process finished with exit code 0
```

```
56gfd24462
пять шесть два четыре четыре шесть два
Process finished with exit code 0
```

13.1 Работа со словарями: Задание 2

Напишите программу, которая выполняется следующим образом.

Пользователь вводит текст. На основе этого текста создается словарь. Ключами словаря служат символы из текста, а значениями элементов словаря являются количества вхождений соответствующих символов в текст.

Например, если пользователь вводит текст *"ABBCAB"*, то словарь будет состоять из трех элементов с ключами *"A"*, *"B"* и *"C"*, а значения элементов соответственно равны 2 (в тексте 2 буквы *"A"*), 3 (в тексте 3 буквы *"B"*) и 1 (в тексте 1 буква *"C"*).

Входные данные:

```
ABBCCYYAB
JhghggHha
tt eeBB asgT TeE EAc
lr 979 r11PP1 bB
```

Ввод/Вывод:

```
Введите ваш текст: ABBCCYYAB
{'A': 2, 'Y': 2, 'B': 3, 'C': 3}
Process finished with exit code 0
```

```
Введите ваш текст: JhghggHha
{'H': 4, 'A': 1, 'J': 1, 'G': 3}
Process finished with exit code 0
```

```
Введите ваш текст: tt eeBB asgT TeE EAc
{' ': 4, 'A': 2, 'C': 1, 'E': 5, 'B': 2, 'G': 1, 'T': 4, 'S': 1}
Process finished with exit code 0
```

```
Введите ваш текст: 1r 979 r11PP1 bB
{'7': 1, 'B': 2, 'R': 2, '9': 2, '1': 4, ' ': 3, 'P': 2}
Process finished with exit code 0
```

13.1 Работа со словарями: Задание 3

В известной игре «**Эрудит**» (Scrabble) каждой букве соответствует определенное количество очков. Общая сумма очков, которую получает игрок, составивший это слово, складывается из очков за каждую букву, входящую в его состав. Чем более употребимой является буква в языке, тем меньше очков начисляется за ее использование.

В *табл.1* приведены все соответствия букв и очков из английской версии игры.

Таблица 1. «Стоимость букв в английской версии игры **Эрудит**»

№	Очки	Буквы
1	1	A, E, I, L, N, O, R, S, T, U
2	2	D, G
3	3	B, C, M, P
4	4	F, H, V, W, Y
5	5	K
6	8	J, X
7	10	Q, Z

Напишите программу, рассчитывающую и отображающую количество очков за собранное слово.

Примечание. На игровом поле *Эрудита* присутствуют специальные клетки, удваивающие и утраивающие стоимость буквы или всего слова. В данном упражнении мы для простоты реализации проигнорируем этот факт.

Входные данные:

```
BATMAN
James Bond 007
bA T maN
```

Ввод/Вывод:

```
Введите слово: BATMAN
BATMAN оценивается в 10 очков!
Process finished with exit code 0
```

```
Введите слово: James Bond 007
James Bond 007 оценивается в 21 очков!
Process finished with exit code 0
```

```
Введите слово: bA T maN
bA T maN оценивается в 10 очков!
Process finished with exit code 0
```

13.1 Работа со словарями: Задание 4

На старых мобильных телефонах текстовые сообщения набирались при помощи цифровых кнопок. При этом одна кнопка была ассоциирована сразу с несколькими буквами, а выбор зависел от количества нажатий на кнопку. Однократное нажатие приводило к появлению первой буквы в соответствующем этой кнопке списке, последующие нажатия меняли ее на следующую.

Список символов, ассоциированных с цифровой панелью, приведен в *табл.2*.

Напишите программу, отображающую последовательность кнопок, которую необходимо нажать, чтобы на экране телефона появился текст, введенный пользователем.

Например, на ввод строки "Hello, World!" ваша программа должна откликнуться следующим выводом:
4433555555666110966677755531111.

Таблица 2. «Таблица клавиатуры T9»

№	Цифра	Символы
1	1	.,?!:
2	2	ABC
3	3	DEF
4	4	GHI
5	5	JKL
6	6	MNO
7	7	PQRS
8	8	TUV
9	9	WXYZ
10	0	пробел

Примечание: Удостоверьтесь, что ваша программа корректно обрабатывает строчные и прописные буквы. При преобразовании букв в цифры игнорируйте символы, не входящие в указанный перечень, такие как точка с запятой или скобки.

Входные данные:

```
Hello, World!
python&java
C++ and C#
```

Ввод/Вывод:

```
Введите ваш текст: Hello, World!
4433555555666110966677755531111
Process finished with exit code 0
```

```
Введите ваш текст: python&java
799984466666528882
Process finished with exit code 0
```

```
Введите ваш текст: C++ and C#
222026630222
Process finished with exit code 0
```

13.1 Работа со словарями: Задание 5

Азбука Морзе зашифровывает буквы и цифры при помощи точек и тире. В *табл.3* приведена та часть азбуки, которая вам понадобится при решении этого задания.

В основной программе вам необходимо запросить у пользователя строку. После этого программа должна преобразовать его в соответствующую последовательность точек и тире, вставляя пробелы между отдельными символами. Символы, не представленные в таблице, можно игнорировать.

Например, сообщение "Hello, World!" может быть представлено следующей последовательностью:-. .-. --- .-- --- .-.-

Таблица 3. «Азбука Морзе»

Символ	Код	Символ	Код	Символ	Код	Символ	Код
A	.-	J	.-.-	S	...	1	.-.-.-
B	-...	K	-.-	T	-	2	..-.-
C	-.-.	L	.-..	U	..-	3	...--
D	-..	M	--	V	...-	4-
E	.	N	-.	W	.-.-	5
F	..-.	O	---	X	-..-	6	-....
G	--.	P	.-.-	Y	-.--	7	--...
H	Q	--.-	Z	--..	8	---..
I	..	R	.-.	0	-----	9	----.

Входные данные:

```
The Simpsons
Rick & Morty
(Micky)! (Mouse);
```

Ввод/Вывод:

```
Введите ваш текст: The Simpsons
- .... .-. .-. --- .-- --- .-. ....
Process finished with exit code 0
```

```
Введите ваш текст: Rick & Morty
.-. .. -. .-. -. - - - .-. - - -
Process finished with exit code 0
```

```
Введите ваш текст: (Micky)! (Mouse);
-- .. -. .-. -. - - - .-. --- .-. ....
Process finished with exit code 0
```

13.1 Работа со словарями: Задание 6

На вход программе подается строка, содержащая строки-идентификаторы.

Напишите программу, которая исправляет их так, чтобы в результирующей строке не было дубликатов. Для этого необходимо прибавлять к повторяющимся идентификаторам постфикс `_n`, где `n` – количество раз, сколько такой идентификатор уже встречался.

Входные данные:

```
hi cool hello hi good cool hi hello cool good nice hi cool good nice  
ab cba abc ab ba cab cba abc ab bca cba ccb aab ab cba bca ab
```

Ввод/Вывод:

```
Введите ваш текст: hi cool hello hi good cool hi hello cool good nice hi cool good nice  
Результат: hi cool hello hi_1 good cool_1 hi_2 hello_1 cool_2 good_1 nice hi_3 cool_3 good_2 nice_1  
Process finished with exit code 0
```

```
Введите ваш текст: ab cba abc ab ba cab cba abc ab bca cba ccb aab ab cba bca ab  
Результат: ab cba abc ab_1 ba cab cba_1 abc_1 ab_2 bca cba_2 ccb aab ab_3 cba_3 bca_1 ab_4  
Process finished with exit code 0
```