

Задачи на работу дома 1 модуль 1 урок

<<Повторение основ Python>>

Переменные

Задача 1

Условие:

В университете Полиннос рады новым талантам! Свежий студент только что вышел из ворот кампуса. Ваша задача состоит в том, чтобы приветствовать его: для этого вам необходимо написать программу, выводящую в консоль приветственное сообщение следующего формата: “Добро пожаловать в Полиннос, {имя и фамилия студента}!”.

Входные данные:

В первой строке вводится имя студента.

Во второй строке вводится фамилия студента.

Выходные данные:

В одной строке выводится строка, соответствующая условию.

Примеры:

<u>Пример</u>	<u>Ввод</u>	<u>Вывод</u>
<u>1</u>	Valerii Yartsev	Привет, Valerii Yartsev!
<u>2</u>	Ivan Ivanov	Привет, Ivan Ivanov!

Подсказки:

- 1) Используйте конкатенацию строк с помощью оператора ‘+’.

Задача 2

Условие:

После праздничного приветствия первокурсников, студент Иван заметил, что имя некоторого студента может быть получено путем объединения 4-х имен другого студента. Известно, что имена студентов могут содержать заглавные буквы в середине. Однако Ивану трудно быстро определить, является ли одно имя конкатенацией 4-х имен другого студента. Ваша задача - разработать программу, которая поможет Ивану определить, может ли имя одного студента быть получено путем объединения 4-х имен другого студента.

Входные данные:

В первой строке вводится имя первого студента.

Во второй строке вводится имя второго студента.

Выходные данные:

В одной строке выводится "True", если имя первого студента является конкатенацией 4-х имен второго студента, иначе – "False"

Примеры:

<u>Пример</u>	<u>Ввод</u>	<u>Вывод</u>
<u>1</u>	MikeMikeMikeMike Mike	True
<u>2</u>	IvanIvanIvan Ivan	False
<u>3</u>	PitPitPitPitt Pit	False

Подсказки:

- 1) Для сравнения строк используется оператор '==', который выводит 'True' или 'False'
- 2) Чтобы получить строку, составленную из 4-х строк, используйте `str * 4`

Условия

Задача 3

Условие:

Жители Полинноса, следя за своим здоровьем, внимательно отслеживают потребление калорий в течение дня. В связи с тем, что фрукты и овощи считаются самыми полезными продуктами, рацион студентов состоит только из трех видов продуктов: яблок, бананов и помидоров. Однако ручной подсчет потребленных калорий каждый день является трудоемким процессом. Поэтому жители Полинноса обратились к вам с просьбой помочь им автоматизировать подсчет калорий.

Калорийность каждого продукта известна заранее:

- Яблоки содержат 52 калории на 100 грамм
- Бананы содержат 89 калорий на 100 грамм
- Помидоры содержат 24 калории на 100 грамм

Входные данные:

В первой строке вводится название продукта: "яблоки", "бананы" или "помидоры".
Во второй строке вводится целое число - вес съеденного студентом продукта (в килограммах).

Выходные данные:

В одной строке выведите одно целое число – количество потребленных студентом калорий.

Примеры:

<u>Пример</u>	<u>Ввод</u>	<u>Вывод</u>
<u>1</u>	яблоки 3	1560
<u>2</u>	бананы 2	1780
<u>3</u>	помидоры 20	4800

Подсказки:

- 1) Используйте условный оператор if (if->elif->else)
- 2) Для считывания из консоли числа нужно использовать приведение к типу `int(input())`

Циклы

Задача 4

Условие:

Жители Полинноса выразили свою благодарность за работу программы, однако возникла проблема: студентам сложно питаться только одним продуктом весь день. Поэтому им было бы удобно, если программа смогла бы учитывать потребление яблок, бананов и помидоров одновременно. Кроме того, подсчет общего количества употребленных продуктов весьма сложен. В связи с этим, студенты записывают каждый свой прием пищи отдельно. Помогите жителям Полинноса узнать их дневной калораж!

Известна калорийность каждого продукта:

- Яблоки: 52 калории на 100 грамм
- Бананы: 89 калорий на 100 грамм
- Помидоры: 24 калории на 100 грамм

Входные данные:

В первой строке вводится целое число n – количество приемов пищи студента. Далее n раз вводится название продукта(яблоки/бананы/помидоры) и потребленный студентом вес этого продукта (в килограммах), разделенные переносом строки.

Выходные данные:

В одной строке выведите одно целое число – количество потребленных студентом калорий.

Примеры:

<u>Пример</u>	<u>Ввод</u>	<u>Вывод</u>
---------------	-------------	--------------

1	2 яблоки 3 бананы 4	5120
2	3 бананы 2 помидоры 5 бананы 1	3870

Подсказки:

- 1) Используйте цикл for для того, чтобы считывать n продуктов
- 2) Складывайте калории для каждого продукта в общую переменную

Задача 5

Условие:

Все жители Полинноса в восторге от вашей программы! Однако бакалейные магазины города столкнулись с огромным наплывом первокурсников, требующих фрукты и овощи. В связи с этим правительство Полинноса ввело ограничение на покупку овощей и фруктов для студентов: не более 50 кг в одни руки. Студенты обратились к вам с просьбой разработать программу, которая поможет им контролировать свои покупки: чтобы студент мог остановиться и не приобретать больше продуктов, если их общий вес уже достигает или превышает установленную норму.

Важно отметить, что если студент не может купить новый продукт так, чтобы общий вес покупок не превышал 50 кг, он будет расстроен и прекратит покупки. В дальнейшем он не сможет продолжить покупки, даже если следующая покупка не превышает установленную норму.

Входные данные:

В командную строку непрерывно вводится вес (в килограммах) продуктов, которые хочет купить студент. Все веса разделены переносом строки.

Выходные данные:

В первой строке выведите общий вес фруктов и овощей, который получилось купить студенту.

Во второй строке выведите количество покупок студента.

Примеры:

<u>Пример</u>	<u>Ввод</u>	<u>Вывод</u>
<u>1</u>	15 40	15 1
<u>2</u>	3 17 20 43	40 3
<u>3</u>	51 14 21	0 0

Подсказки:

- 1) Используйте цикл while для того, чтобы останавливаться, когда общий вес продуктов превышает норму
- 2) Может возникнуть проблема с последним продуктом, который уже превысит общий вес продуктов: он учитываться не должен (так как с ним вес покупок превышает норму). Эту проблему можно решить 2-мя способами: 1) написать бесконечный цикл while, и уже в самом цикле проверять на превышение нормы, и в случае превышения использовать оператор break для выхода из цикла (тем самым мы можем не добавлять вес нового продукта, соответственно, последний «превышающий» продукт добавлен не будет); 2) Хранить вес последнего добавленного продукта: тогда мы сможем вычесть его вес уже при выводе результата