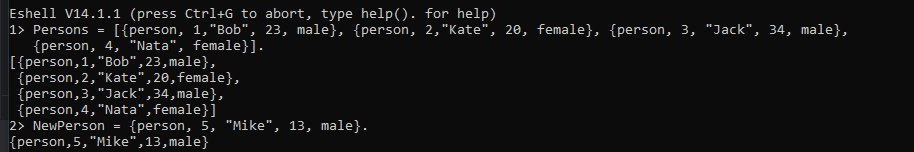
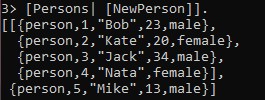
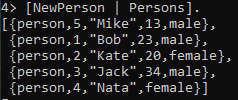
Задание 1

Присваивание значений переменным:   


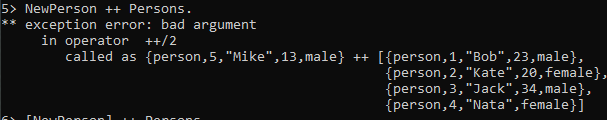
1. [Persons | [NewPerson]] – Сопоставление к списку (head)Persons, (tail)списка с элементом NewPerson.



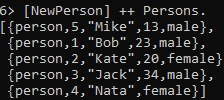
1. [NewPerson | Persons] – сопоставление NewPerson в начало списка Persons.



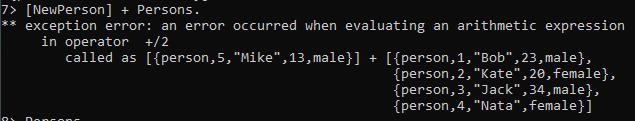
1. NewPerson ++ Persons. – Такой вид операции невозможен. Из-за невозможности складывания картежей и списоков напрямую.



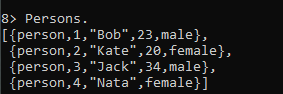
1. [NewPerson] ++ Persons. – Складывание сиписка с элементом NewPerson с списоком Persons.



1. [NewPerson] + Persons. – Такая операция невозможна для данного типа данных. Сумма предназначена для типа данных Float и Integer.



1. Persons. – Проверка на изменения переменной.



Задание 2

Так как сложение фактический происходит с значением 0, то в результате получается значение 10.



Аналогично с прошлым случаем. Происходит вычитание из 0 и получается значение -10.



Стандартная сумма двух слагаемых с результатом 12.



При делении возвращается формат Float, вне зависимости от типа чисел. В результате возвращается значение 4.0.



Возвращает значение, полученное при делении числа на число без остатка. Функция предназначена только для целочисленных значений.



Метод rem предназначен для получения остатка от деления числа на число, следовательно при делении получается остаток от деления.



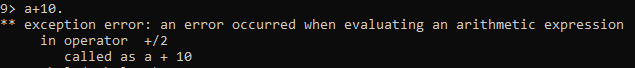
Выполняется побитовое AND, возвращающая число, соответсвуеющих битов обоих операндов 1.



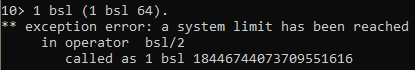
Выполняется побитовое OR, возвращающая число, которой соответствующие биты одного или обоих соответсвуеющих операндов равны 1.



Для Atom и Intergern не выполняется арифметическая сумма.



Первый сдвиг пройдет нормально, но результат второго будет превышать системный лимит в 2305843009213693951 bytes.



Задание 3

Night – в виде строки.



Night – в виде битовых значений.



Night – в виде списка символов.



Night – в виде бит-строки.

