**from** **\_\_future\_\_** **import** (absolute\_import, division,

print\_function, unicode\_literals)

**from** **builtins** **import** \*

1. Даны два целых числа A и B (при этом A   B). Выведите все числа от A до B включительно.
2. По данному натуральном n вычислите сумму квадратов n целых чисел.
3. По данному целому неотрицательному n вычислите значение n!.
4. По данным целым неотрицательным n и k вычислите значение числа сочетаний из n элементов по k.
5. Напишите программу, которая по данному числу n от 1 до 9 выводит на экран n пингвинов. Изображение одного пингвина имеет размер 5×9 символов, между двумя соседними пингвинами также имеется пустой (из пробелов) столбец. Разрешается вывести пустой столбец после последнего пингвина. Для упрощения рисования скопируйте пингвина из примера в среду разработки.

penguine = [" \_~\_ ",

" (o o) ",

" / V \ ",

"/( \_ )\ ",

" ^^ ^^ "]

1. Шоколадка имеет вид прямоугольника, разделенного на n×m долек. Шоколадку можно один раз разломить по прямой на две части. Определите, можно ли таким образом отломить от шоколадки ровно k долек.
2. Дано линейное уравнение ax+b=0. Решите уравнение, напечатайте ответ. Если ответов бесконечно много, выведите "INF", если их нет - "NO".
3. Для данного числа n < 100 закончите фразу “На лугу пасется...” одним из возможных продолжений: “n коров”, “n корова”, “n коровы”, правильно склоняя слово “корова”.
4. Диофантово уравнение. Даны числа a, b, c, d. Выведите в порядке возрастания все целые числа от 0 до 1000, которые являются корнями уравнения .
5. Квадрат трехзначного числа оканчивается тремя цифрами, равными этому числу. Найдите и выведите все такие числа.
6. По данному натуральному   выведите лесенку из n ступенек, i-я ступенька состоит из чисел от 1 до i без пробелов.
7. Дано три числа. Упорядочите их в порядке неубывания. Программа должна считывать три числа a, b, c, затем программа должна менять их значения так, чтобы стали выполнены условия  , затем программа выводит тройку a, b, c.
8. По данному натуральном n вычислите сумму 1! + 2! + 3! + ... +n!. В решении этой задачи можно использовать только один цикл.