# Лабораторная работа №1: «Линейные программы»

# Постановка задачи

Напишите программу для расчета по заданным формулам. Предварительно подготовьте тестовые примеры с помощью калькулятора или электронной таблицы Excel.

Для математических вычислений в Python имеются как встроенные, так и дополнительные функции и методы. Применить дополнительные математические функций можно после подключения модуля math:

import math

либо

from math import \*

В первом случае функции рассматриваются как методы объекта math и должны записываться так:

import math

print(math.sin(math.pi/4)) print(math.sqrt(2)/2)

Во втором случае вызов функции может быть сделан в более привычной для нас форме:

from math import \*

print(sin(pi/4))

print(sqrt(2)/2)

Вместе с тем, такой способ импорта может нарушить пространство имен программы, поскольку может возникнуть конфликт между именами переменных, которые использует программист и именами импортируемых функций. При импорте можно ограничиться только необходимыми функциями, например:

from math import (pi, sin, cos,tan, log)

# Ввод данных

Ввод данных можно выполнить с клавиатуры функцией input():

m = input([str])

При этом на экран будет выведена строка str, а переменная m получит значение строкового типа, введённое пользователем. Строковый тип может быть преобразован, например, к типу int или float, если введённое значение – число.

# Вывод данных

Вывод данных на экран монитора может быть выполнен функцией print(). Эта функция позволяет выполнять форматированный вывод, как с использованием Си-подобного форматирования, так и с использованием форматной строки Python.

Следующие строки демонстрируют, как можно форматировать вывод.

for x in range(1,11):

print('%2d %3d %7.2f' % (x, x\*x, x\*x\*x))

print("{0:.2f} {1:.2f} {2:.4f}".format(a, x, y))

Буква в формате числа определяет тип выводимого числа. Так, d – это целый тип, f – вещественное число. Число в формате означает то число позиций, которое будет использовано для вывода числа. Для вещественного числа указывается, после точки, количество выводимых десятичных знаков.

Во второй строке использован Си-подобный формат, в котором формат числа начинается с процента "%". В этом формате аргументы отделяются от форматной части строки так же символом % – процент.

В третьей строке используется форматная строка Python, в которой в форматной строке позиции для значений аргументов выделяются фигурными скобками.

Обратите внимание на то, что сами форматные строки начинаются и завершаются одиночной или двойной кавычкой. В Python допускаются оба вида кавычек для выделения строки. Важно только что бы начало и конец были одинаковыми.

Так же следует понимать, что в промежутках между символами форматирования могут находиться и другие символы или слова:

print('x=%2d x^2=%3d x^3=%7.2f'% (x, x\*x, x\*x\*x))

print("a={0:.2f} x={1:.2f} y={2:.4f}".format(a, x, y))

Использование форматных строк делает вывод данных более внятным.

# Решение задания

Вернемся к нашим примерам и запишем их, используя правила языка Python. Первое выражение примет вид:

1. y = tan(x\*\*2/2-1)\*\*2+2\*cos(x-pi/6)/(1/2+sin(a)\*\*2)

Второе выражение представим в виде двух:

2. tmp=log(3-cos(pi/4+2\*x),3+sin(x))/(1+tan(2\*x/pi)\*\*2)

y = pow(2, tmp)

# Описание алгоритма

Для вычислений необходимо обеспечить ввод двух переменных x и a. Поскольку по условиям задачи их тип и точность представления не заданы, выберем для них вещественный тип (float). Для оптимизации записи выражения используем промежуточную переменную tmp.

1. Ввести значения a и x, преобразовать к типу float.

2. Вычислить выражение 1.

3. Вывести результат вычисления.

4. Вычислить значение переменной tmp;

5. Вычислить выражение 2.

6. Вывести результат вычисления.

# Листинг программы

from math import \*

a = float(input('Введите параметр а: '))

x = float(input('Введите значение x: '))

y=tan(x\*\*2/2-1)\*\*2+(2\*cos(x-pi/6))/(1/2+sin(a)\*\*2)

print("{0:.2f} {1:.2f} {2:.4f}".format(a, x, y))

tmp=log(3-cos(pi/4+2\*x),3+sin(x))/(1+tan(2\*x/pi)\*\*2)

y=pow(2,tmp)

print("{0:.2f} {1:.4f}".format(x, y))

# Результаты тестирования программы



# Псевдографика

Для создания таблиц при печати отчетов с помощью символов псевдографики можно использовать следующую таблицу:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ─ 196 |  | ┌ 218 | ┬ 194 | ┐ 191 |  | ╒ 213 | ╤ 209 | ╕ 184 |  | ╓ 214 | ╥ 210 | ╖ 183 |  | ╔ 201 | ╦ 203 | ╗ 187 |
| │ 179 | ├ 195 | ┼ 197 | ┤ 180 | ╞ 198 | ╪ 216 | ╡ 181 | ╟ 199 | ╫ 215 | ╢ 182 | ╠ 204 | ╬ 206 | ╣ 185 |
| ═ 205 | └ 192 | ┴ 193 | ┘ 217 | ╘ 212 | ╧ 207 | ╛ 190 | ╙ 211 | ╨ 208 | ╜ 189 | ╚ 200 | ╩ 202 | ╝ 188 |
| ║ 186 |  | | | | | | | | | | | | | | |

# Задания к лабораторной работе №1 «Линейные программы»

Напишите программу для расчета по двум формулам. Подготовьте не менее пяти тестовых примеров. Предварительно выполните вычисления с использованием калькулятора или офисного приложения, например, Excel или Calc. Результаты вычисления по обеим формулам должны совпадать. Отсутствующие в языке функции выразите через имеющиеся.

 

# ПРИЛОЖЕНИЕ. Функции и методы в Python

