

1. ЛАБОРАТОРНАЯ РАБОТА 1. ЛИНЕЙНЫЕ АЛГОРИТМЫ. ОПЕРАТОРЫ PYTHON. ЗНАЧЕНИЯ, ПЕРЕМЕННЫЕ И ТИПЫ ДАННЫХ. ВВОД-ВЫВОД. МАТЕМАТИЧЕСКИЕ ФУНКЦИИ.







1.1. СПРАВОЧНЫЙ МАТЕРИАЛ

Линейный алгоритм – это алгоритм, в котором все действия выполняются в строгом порядке, последовательно, одно за другим.

Программа, реализующая линейный алгоритм, называется программой с линейной структурой. Приступая к разработке программ с линейной структурой, следует учитывать, что:

- программы с линейной структурой являются простейшими и используются, как правило, для реализации простых вычислений по формулам;
- в программах с линейной структурой инструкции выполняются последовательно, одна за другой;
- алгоритм программы с линейной структурой может быть представлен в виде блок-схемы, использующей следующие графические символы (таблица 1).

Таблица 1. Графические символы, используемые для построения блок-схем

| Графический символ | Название и назначение графического символа |
|---|--|
|  | Начало/конец работы Начало, конец, прерывание процесса обработки данных или выполнения программы |
|  | Данные Перечисляются данные, которые необходимы для работы алгоритма |
|  | Процесс Выполнение операции или группы операций, в результате чего изменяется значение, форма представления или расположение данных |
|  | Ручной ввод данных Ручной ввод данных с клавиатуры |
|  | Дисплей Вывод данных, результатов вычислений, сообщений, запросов к пользователю и пр. на дисплей |
|  | Комментарий Связь между элементами блок-схемы и пояснениями |

Операторы Python. В языке Python встречаются операторы 7 типов (таблица 1).

Таблица 1.

| Операторы Python | | | |
|--------------------------|---|--------------------------------------|----------------------|
| Оператор | Название оператора | Пример | Результат вычисления |
| Арифметические операторы | | | |
| + | Сложение | >>>2+3 | 5 |
| - | Вычитание | >>>4-1 | 3 |
| * | Умножение | >>>4*5 | 20 |
| / | Деление | >>>8/4 | 2 |
| ** | Возведение в степень | >>>2**3 | 8 |
| // | Деление без остатка | >>>8//3 | 2 |
| % | Деление по модулю (остаток от деления) | >>>12%7 >>>12%6 | 5 0 |
| Операторы сравнения | | | |
| < | Меньше | >>>2<7 | True |
| > | Больше | >>>2>7 | False |
| <= | Меньше или равно | >>>3<=3 | True |
| >= | Больше или равно | >>>3>=3 | True |
| == | Равно (проверяет равно ли значение слева правому. В случае равенства выдает значение «TRUE») | >>>6==6 | True |
| != | Не равно | >>>6!=6 | False |
| Операторы присваивания | | | |
| = | Присваивание | >>>a=3+8 >>>print(a) ¹ | 11 |
| += | Сложение и присваивание (суммирует значения слева и справа, результат присваивает переменной слева) | >>>a=7 >>>a+=3 >>>print(a) | 10 |

¹ Print(a) - функция **print** () используется для вывода текстовой информации в консоль или на экран.

| | | | | |
|---------------------------|--|--------|--|-------|
| -= | Вычитание присваивание | и | >>>a=7 >>>a-=3 >>>print(a) | 4 |
| /= | Деление присваивание | и | >>>a=12 >>>a/=3 >>>print(a) | 4 |
| *= | Умножение присваивание | и | >>>a=5 >>>a*=3 >>>print(a) | 15 |
| %= | Деление по модулю и присваивание | | >>>a=6 >>>a%=3 >>>print(a) | 0 |
| **= | Возведение степень присваивание | в и | >>>a=2 >>>a**=3 >>>print(a) | 8 |
| //= | Деление с остатком и присваивание | | >>>a=7 >>>a//=3 >>>print(a) | 2 |
| Логические операторы | | | | |
| and | И – используется для объединения нескольких условий | | >>> a = 5 > 5 and 2 > -3 >>> print(a) | False |
| or | ИЛИ – используется для проверки истинности хотя бы одного условия | | >>> a = 5 > 5 or 2 > -3 >>> print(a) | True |
| not | НЕ – используется для инвертации булевых значений | | >>>a = not(0) >>> print(a) | True |
| Операторы принадлежности | | | | |
| in | Используются для проверки принадлежности последовательности (список, строка или кортеж) | | >>>животное = ['собака','кошка','лошадь'] >>>'собака' in животное | True |
| not in | Используются для проверки отсутствия в последовательности | | >>>животное = ['собака','кошка','лошадь'] ² >>>'попугай' not in животное | True |
| Операторы тождественности | | | | |
| is | | | >>> 2 is 10 | False |

² Кавычки скобки

| | | | |
|--------|---|------------|------|
| is not | Используются для проверки, являются ли операнды одинаковыми (занимают ли они одну и ту же позицию в памяти) | >>> 3 is 3 | True |
|--------|---|------------|------|

Значения, переменные и типы данных. *Значение* - основное понятие, с которым работает программа. Например: 2, 56.0, 'собака'. Эти значения принадлежат различным *типам данных*. В Python существуют следующие типы данных:

1. Числа (Numeric Type):

- ✓ **int** – целое число;
- ✓ **float** – число с плавающей точкой;
- ✓ **complex** – комплексное число.

2. Строки (Text Sequence Type):

- ✓ **Str.**

3. Списки (Sequence Type):

- ✓ **List** – список;
- ✓ **Tuple** – кортеж;
- ✓ **Range** – диапазон.

4. Логические переменные (Boolean Type)/

5. None (неопределенное значение переменной).

Для проверки типа указанного значения можно воспользоваться функцией **type()**.

Переменная – это имя, которое ссылается на значение. Имя переменной в языке Python может иметь любую длину, включать латинские буквы и цифры. **НО!!!** Имя переменной не может начинаться с цифры!!!

Также в качестве имени переменной нельзя использовать зарезервированные слова: *False, class, finally, is, return, None, continue, for, lambda, try, True, def, from, nonlocal, while, and, del, global, not, with, as, elif, if, or, yield, Assert, else, import, pass, break, except, in, raise.*

Ввод данных и преобразование типов

Ввод данных с клавиатуры осуществляется с помощью встроенной функции **input**. Формат применения следующий: **str=input("Введите строку")**, т.е. функция **input** в качестве необязательного параметра принимает строку вывода сообщения перед вводом данных, а результат ввода записывается в виде строки символов в принимающую переменную (в нашем примере это переменная **str**). Если необходимо ввести целое число – *делается преобразование типов* с помощью функций **int()**, **float()** или **str()**, например:

```
int_var = int(input("Введите целое число: "))
```

Аналогично с вещественным:

```
float_var = float(input("Введите вещественное число: "))
```

В общем случае функция `int()` имеет два аргумента. Первый аргумент – значение, которое нужно преобразовать в целое число, второй аргумент (необязательный) система исчисления в которой задан первый аргумент (может быть от 2 до 36), по умолчанию второй аргумент равен 10. При использовании второго аргумента отличного от 10 – первый аргумент должен быть представлен в виде строки.

Для корректного ввода без ошибок преобразования необходимо использовать цикл и исключение, например, вот так:

```
def getnum():
    while (True):
        try:
            num = int(input("Введите целое число: "))
            return num
        except ValueError:
            print("Вы ввели не число. Повторите ввод")
```

Если строку ввода необходимо разбить на несколько значений используются пробелы и функция `split()`, которая разбивает строку на лексемы (подстроки), а разделителем по умолчанию служит пробел. Например,

```
a, b, c = input().split()
```

или так (если вводится три целых числа):

```
x, y, z = [int(s) for s in input().split()]
```

Для преобразования целых чисел в числа в двоичной, восьмеричной и шестнадцатеричной системы счисления могут использоваться функции `bin()`, `oct()`, `hex()`.

Вывод данных

Встроенная функция для вывода на экран имеет следующий формат:

```
print(obj1, obj2, ..., sep=' ', end='\n', file=sys.stdout, flush=False)
```

Здесь:

- **obj1, obj2, ...** — объект/объекты которые необходимо вывести;

- **sep** — разделитель между объектами. В качестве своего значения можно передавать строку или None (по умолчанию пробел " ");
- **end** — символ в конце строки (по умолчанию перенос строки \n);
- **file** — file-like объект [поток] (по умолчанию sys.stdout);
- **flush** — принудительный сброс потока [работает с версии Python 3.3] (по умолчанию False).

Например, следующая функция выведет на экран строку:

Меркурий, Венера, Земля, Марс!

```
print('Меркурий', 'Венера', 'Земля', 'Марс', sep=', ', end='!')
```

Форматированный вывод данных

Для вывода на экран значений в отформатированном виде используется три способа:

- 1) старый стиль(метод), основанный на форматировании шаблонов строк, как в языке C;
- 2) новый метод с использованием функции format();
- 3) f-строки.

Форматирование в старом стиле предполагает вывод на экран в виде:

”строкаформата” % (значениедлявывода1, зн2,...)

В строке формата используются следующие шаблоны:

| | |
|----|--|
| %d | Целое число |
| %f | Вещественное число |
| %s | Строка |
| %c | Символ |
| %e | Число в научном формате, в виде мантиссы и порядка |

Также внутри строки могут использоваться управляющие символы, вот некоторые из них:

| | |
|----|--|
| \n | (newline) перевод каретки на следующую строку |
| \r | (return) перевод каретки на в начало текущей строки |
| \t | (tab) табуляция (отступ, красная строка, 8 пробелов) |
| \b | (backspace) перевод каретки на один символ назад с удалением символа |
| \\ | Позволяет записать символ обратного слеша. |

| | |
|-------------|--|
| \' | Позволяет записать один символ апострофа. |
| \" | Позволяет записать один символ кавычки. |
| \xhh | Шестнадцатеричный код символа (две шестнадцатеричные цифры hh). |
| \ooo | Восьмеричный код символа (три восьмеричные цифры ooo). |
| \0 | Символ Null. |
| \N{id} | ID (идентификатор) символа в базе данных Юникода, или, проще говоря, его название в таблице Юникода. |
| \uhhhh | Шестнадцатеричный код 16-битного символа Юникода (символ кодируемый двумя байтами). |
| \Uhhhhhhhhh | Шестнадцатеричный код 32-битного символа Юникода (символ кодируемый четырьмя байтами). |

В строке формата могут указываться дополнительные параметры:

%ширина_поля.число_знаков_после_запятой

Например,

```
print("Число:%15.5f" % (12.12345656))
```

Результат:

```
012345123456789012345  <- номер позиции на экране
Число:                12.12346
```

```
print("%.4f, %.2f" % (1.33334, 153*0.43))
#1.3333, 65.79
print("%f, %f" % (1.33334, 153*0.43))
#1.333340, 65.790000
print(153*0.43)
#65.78999999999999
```

Еще пример табличного вывода:

```
print("%5d%7d" % (10, 235))
print("%5d%7d" % (1000, 50))
```

Результат:

```
    10    235
 1000    50
```

Если нужно выровнять по левому полю добавляется знак «-»:

```
print("%-5d%7d" % (10, 235))
print("%-5d%7d" % (1000, 50))
```

Результат:

```
    10    235
 1000    50
```

Математические функции. Для доступа к большинству математических функций, необходимо импортировать **модуль (math)**:

```
>>> import math
>>> math
<module 'math' (built-in)>
>>> |
```

Объект модуля содержит функции, переменные и константы, определенные в модуле. Для доступа к функциям используется формат «точечной нотации». Указывается имя модуля и имя функции через точку. Например, чтобы вычислить десятичный логарифм, код программы будет выглядеть следующим образом:

Имя модуля Имя функции

```
>>> math.log10(1000)
3.0
>>>
```

Перечень основных математических функций Python представлен в таблице.

| Название | Описание |
|-------------------------|--|
| math.ceil(a) | Округление до ближайшего большего целого числа. |
| math.cmp(a,b) | -1 если $a < b$, 0 если $a == b$, или 1, если $a > b$ |
| math.e | $e = 2,718281...$ |
| math.exp(x) | e^x |
| math.fabs(a) | Модуль числа (в отличие от встроенной функции abs, fabs возвращает всегда вещественное число). |
| math.factorial(a) | Факториал числа. |
| math.floor(a) | Округление вниз, до ближайшего меньшего целого. |
| math.fmod(a,b) | Остаток от деления a на b. В отличие от встроенной функции нахождения остатка, всегда возвращает вещественное число. |
| math.fsum(a1,a2,...,an) | Возвращает сумму чисел из списка. |
| math.log(a,[основание]) | Натуральный логарифм. Если указать доп. параметр основание, то вычисляет соответствующий логарифм. |
| math.log10(a) | Десятичный логарифм. |
| math.log2(a) | Логарифм по основанию 2. |
| math.max(a,b,c,...) | Находит максимальное среди чисел. |
| math.min(a,b,c,...) | Находит минимальное среди чисел. |
| math.pow(a,b) | Возведение в степень (a^b). |
| math.round(a) | Округление (требует уточнения). |
| math.sqrt(a) | Извлечение квадратного корня. |

1.2. ПРИМЕРЫ РЕШЕНИЯ ЗАДАЧ

Пример 1. Правительство гарантирует, что инфляция в новом году составит $p\%$ в месяц. Какой рост цен ожидаем за год? (Формула сложных процентов: $S = \left[\left(1 + \frac{p}{100} \right)^{12} - 1 \right] \cdot 100\%$.

Решение:

Для ввода исходной информации $p\%$ воспользуемся функцией **input()**. В скобках указываем сообщение для ввода. Команда **input()** по умолчанию воспринимает входные данные как строку символов. Поэтому, чтобы ввести целочисленное значение (число с плавающей точкой), следует указать тип данных **int()** или **float()**.

```
import math
math
p=int(input("Введите темп инфляции за месяц: "))
S=((1+p/100)**12-1)*100
print('Годовой темп инфляции составит: ', S, '%')
```

Для вывода используем функцию **print()**. В команде **print()** через запятую можно задавать вывод нескольких элементов, причем текстовые значения задаются в кавычках. Результат расчета и вывода:

```
Введите темп инфляции за месяц: 3
Годовой темп инфляции составит: 42.57608868461793 %
```

Пример 2. Вычислить значение функции в заданных точках и округлить результат до тысячных:

$$z = \ln \left| \lg x - \sqrt{|\cos x - e^x|} \cdot \arcsin \left| \operatorname{tg} \frac{|ax - b|}{\sin |x|} + b \right| \right|, \text{ где } a = 0.126, b = 0.842, x = 0.34$$

.

Решение:

Для упрощения, введем дополнительные переменные и присвоим им значения математических функций.

```

import math
math
a=0.126
b=0.842
x=0.34
z1=math.log(x)
z2=math.sqrt(math.fabs(math.cos(x)-math.exp(x)))
z3=math.tan((math.fabs(a*x-b))/math.sin(x))+b
z4=math.asin(math.fabs(z3))
z=round(math.log(math.fabs(z1-z2*z4)),3)
print(z)
=====
0.126

```

Пример 3. Необходимо разработать блок-схему линейного алгоритма для вычисления выражения, определяемого соотношением

$$z = 2^{-x} \cdot \sqrt{x + \sqrt[4]{|y|}} \cdot \sqrt{e^{x-1/\sin x}}$$

Решение.

Анализ задачи показывает, что вычисление выражения по этой формуле можно упростить, если это выражение представить в виде трех сомножителей: $z = a1 * a2 * a3$;

где $a1 = 2^{-x}$ – первый сомножитель;
 $a2 = \sqrt{x + \sqrt[4]{|y|}}$ – второй сомножитель;
 $a3 = \sqrt{e^{x-1/\sin x}}$ – третий сомножитель.

Таким образом, линейный алгоритм для вычисления выражения по формуле z представим в виде, как показано на рисунке 1.

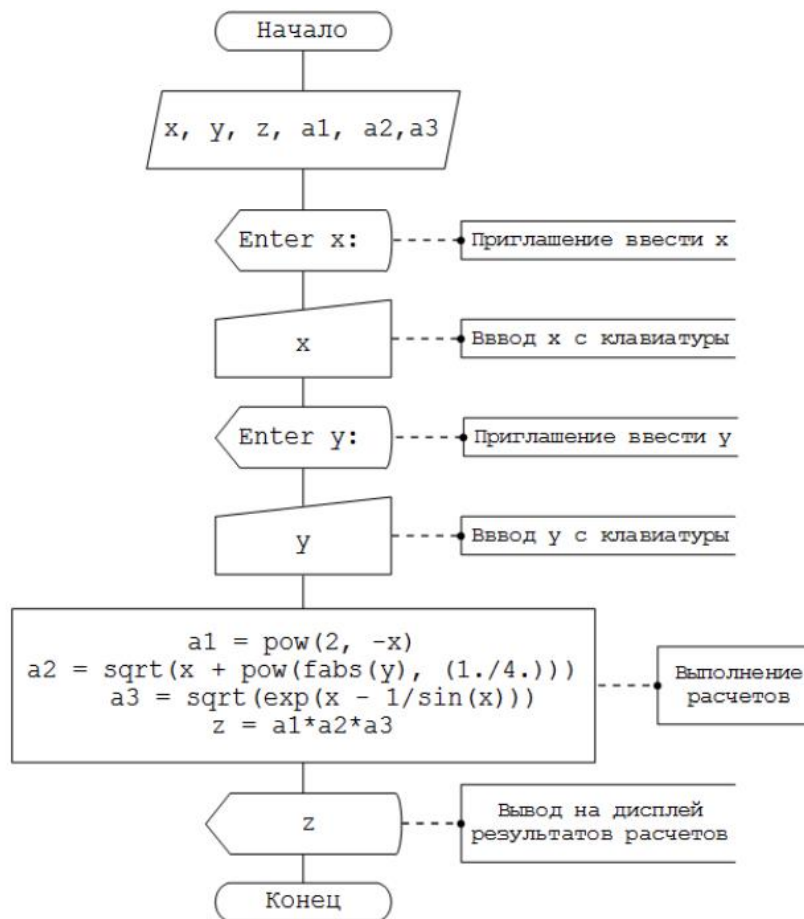


Рисунок 1. Линейный алгоритм решения задачи

Как видно на рисунке 1 алгоритм включает следующие девять блоков:

- 1) начало;
- 2) данные (переменные $x, y, z, a1, a2, a3$);
- 3) дисплей (приглашение ввести значение переменной x);
- 4) ручной ввод данных (ввод значения переменной x с клавиатуры);
- 5) дисплей (приглашение ввести значение переменной y);
- 6) ручной ввод данных (ввод значения переменной y с клавиатуры);
- 7) процесс (вычисление значения вспомогательных переменных $a1, a2, a3$ и искомой переменной z);
- 8) дисплей (вывод на дисплей результатов расчетов значения искомой переменной z);
- 9) конец.

1.3. ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

1. Изучить руководство к лабораторной работе.
2. Нарисовать блок-схемы алгоритмов для примеров 1, 2.
3. Написать и отладить программы из примеров 1, 2.
4. Написать программу для блок-схемы алгоритма в примере 3.
5. Разработать алгоритмы и написать программы согласно заданиям по вариантам. Задания и варианты приведены ниже. В каждом варианте 4 задания.
6. Подготовить отчет. В отчет включить блок-схемы для всех задач и окончательный рабочий код всех программ, а также скриншоты выполнения для каждой программы. Всего 7 задач: 3 задачи общие из примеров и 4 задачи индивидуальные по варианту.

1.4. ВАРИАНТЫ ЗАДАНИЙ

| № задачи | Вариант 1 | Вариант 2 | Вариант 3 | Вариант 4 | Вариант 5 | Вариант 6 | Вариант 7 | Вариант 8 | Вариант 9 | Вариант 10 | Вариант 11 | Вариант 12 | Вариант 13 | Вариант 14 | Вариант 15 | Вариант 16 | Вариант 17 | Вариант 18 | Вариант 19 | Вариант 20 |
|----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|-----------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|------------|
| 1. | * | | | | | | | * | | | | * | | | | | | | * | |
| 2. | | * | | | | | | | * | | | | * | | | | | * | | |
| 3. | | | * | | | | | * | | * | | | | * | | | | | | |
| 4. | | | | * | | | | | * | | * | | | | * | | | | | |
| 5. | * | | | | * | | | | | | | * | | | | * | | | | |
| 6. | | * | | | | * | | | | | | | * | | | | * | | | |
| 7. | | | * | | | | * | | | | | | | * | | | | * | | |
| 8. | | | | * | | | | * | | | | | | | * | | | | * | |
| 9. | | | | | * | | | | * | | | | | | | * | | | | * |
| 10. | * | | | | | * | | | | * | | | | | | | * | | | |
| 11. | | * | | | | | * | | | | * | | | | | | | * | | |
| 12. | | | * | | | | | | | | | * | | | | * | | | * | |
| 13. | | | | * | | | | | | | | | * | | | | * | | | * |
| 14. | | | | | * | | | | | * | | | | * | | | | * | | |
| 15. | | | | | | * | | | | | * | | | | * | | | | * | |
| 16. | | | | | * | | * | | | | | | | | | * | | | | * |
| 17. | * | | | | | | | * | | | | * | | | | | * | | | |
| 18. | | * | | | | * | | | * | | | | * | | | | | | | |
| 19. | | | * | | | | * | | | * | | | | * | | | | | | |
| 20. | | | | * | | | | | | | * | | | | * | | | | | * |

Варианты после 20 повторяются в циклическом порядке

1.5. ЗАДАЧИ ДЛЯ ВАРИАНТОВ

1. Вычислить периметр и площадь прямоугольного треугольника по заданным длинам двух катетов a и b .
2. Заданы координаты трех вершин треугольника (x_1, y_1) , (x_2, y_2) , (x_3, y_3) . Найти его периметр и площадь.
3. Вычислить длину окружности и площадь круга одного и того же заданного радиуса R .
4. Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
5. Вычислить расстояние между двумя точками с данными координатами (x_1, y_1) и (x_2, y_2) .
6. Даны два действительных числа x и y . Вычислить их сумму, разность, произведение и частное.
7. Дана длина ребра куба. Найти площадь грани, площадь полной поверхности и объем этого куба.
8. Дана сторона равностороннего треугольника. Найти площадь этого треугольника, его высоты, радиусы вписанной и описанной окружностей.
9. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
10. Найти площадь равнобедренной трапеции с основаниями a и b и углом α при большем основании a .
11. Найти площадь треугольника, две стороны которого равны a и b , а угол между этими сторонами равен g .
12. Три сопротивления R_1 , R_2 , R_3 соединены параллельно. Найдите сопротивление соединения.
13. Составить программу вычисления объема цилиндра и конуса, которые имеют одинаковую высоту H и одинаковый радиус основания R .
14. Даны два действительных числа. Найти среднее арифметическое этих чисел и среднее геометрическое их модулей.
15. Определить время падения камня на поверхности земли с высоты h .
16. Дана сторона равностороннего треугольника. Найти площадь этого треугольника.
17. Определить силу притяжения F между телами массы m_1 и m_2 , находящимся на расстоянии r друг от друга. Результат округлить до 2-х знаков после запятой.
18. Известны первый и пятый члены арифметической прогрессии. Найти величину члена прогрессии с номером N и сумму N членов.
19. Вычислите и выведите на экран примерное число прожитых человеком дней (без учёта високосных лет), если в году 365 дней, а год рождения и текущий год запрашиваются у пользователя вашей программы.
20. Напишите программу, вычисляющую какую сумму денег нужно платить за электроэнергию, если у пользователя программы запрашивается количество кВт электроэнергии, которое было израсходовано за расчётный месяц, и какова плата за 1 кВт электроэнергии.