



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ  
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение  
высшего образования

**«МИРЭА – Российский технологический университет»**  
**РТУ МИРЭА**

*Институт информационных технологий*  
*Кафедра Вычислительной техники*

**Практическая работа №1**

По дисциплине «Программирование на языке Python»

**Исполнитель** студент группы ИВБО-02-19 К.Ю.Денисов \_\_\_\_\_  
**Преподаватель** \_\_\_\_\_

Работа представлена к защите «\_\_\_\_\_» \_\_\_\_\_ 2021 г.

Оценка «\_\_\_\_\_»

Москва 2021

# 1 Практическое занятие №1. Вариант №6

Цель работы: изучить синтаксис языка программирования *Python*, изучить синтаксис условных операторов, операторов ветвления и математических операторов.

**Задача 1.1.** Реализовать функцию

$$f(x,y,z) = \sin y - 36y^2 + \frac{\cos 67x^6 - z^5 - 89 + \frac{z^6}{36}}{\cos x - 81x^6 + 36} - \sqrt{\frac{\cos z + 19x^8 - 19}{z^3 - y^2 - 94}}.$$

*Решение.* Для реализации данной функции воспользуемся стандартными математическими операторами и функциями языка *Python*.

```
def simple (x,y) :  
    return (x**4 + x)**2 - 69*y**8 -math.sqrt((x/8 -  
        86*x**4)/(math.sin(x) +x**3 - 24)) - ((y**8 +  
        math.exp(x))/(x**5 - math.log(y)))
```

Примеры вычисления *simple*:

1.  $simple(75, -13, -90) = -4.84e + 13$

2.  $simple(21,47, -39) = -3.22e + 11$

**Задача 1.2.** Реализовать кусочно-линейную функцию

$$f(x) = \begin{cases} 27(\cos x + \sin x)^4 + 91x^6, & \text{если } x < 188; \\ x^3 + 49x^8, & \text{если } 188 \leq x < 247; \\ 96x^7 - 76x^5, & \text{если } 247 \leq x < 285; \\ x^3 - \cos x - 78, & \text{если } 285 \leq x < 384; \\ x + x^3 + 5, & \text{если } x \geq 384. \end{cases}$$

*Решение.* Для решения данной задачи используем условный оператор *if*

```
def piecewise (x) :  
    if x < -9 :  
        return x**8 + 37*x**5 + 31  
    if x >= -9 and x < 59:  
        return x**3 - math.log(x) - 90  
    if x >= 59 and x < 83:  
        return (92*x**6 + math.sin(x) - 39)**4 -  
            59*x**5  
    if x >= 83:  
        return 67*(x**8) - x**2
```

Примеры вычисления *piecewise* :

1.  $piecewise(124) = 6.55e + 18$

2.  $piecewise(116) = 2.22e + 14$

**Задача 1.3.** Реализовать итерационную функцию

$$f(n) = 51 \sum_{i=1}^n (27i^3 + i^7) - \sum_{i=1}^n \left( e^i + \frac{i^5}{68} \right).$$

*Решение.* Для реализации данной итерационной функции используем операторы цикла *for in range*

```
def iterate (n,m):  
    acc1 = 0  
    acc2 = 0  
    for i in range(1,n+1):  
        for j in range(1,m+1):  
            acc1 += j**3 + 37*j**4 + 31  
    for i in range(1,n+1):
```

```
acc2 += math.cos(i) + math.log(i)
return 60*acc1 + 43*acc2
```

Примеры вычисления *iterate*:

$$1. \text{iterate}(45) = -5.53e + 19$$

$$2. \text{iterate}(50) = -8.20e + 21$$

**Задача 1.4.** Реализовать рекуррентную функцию

$$f(0) = 10,$$

$$f(1) = 5,$$

$$f(n) = \frac{1}{86}f(n-1) - \cos f(n-2)$$

*Решение.* Для реализации рекуррентной функции прибегнем к рекурсивному вызову разрабатываемой функции.

```
def recursion (x):
    if x == 0: return 6
    if x == 1: return 10
    return math.tan(recursion(x-1)) -
        1/30*recursion(x-2)**2
```

Примеры вычисления *recursion(x)*:

$$1. \text{recursion}(16) = -7.31e - 01$$

$$2. \text{recursion}(14) = -7.63e - 01$$

## 2 Практическая работа №2. Вариант 6

Задача 2.1. Реализовать функцию–дерево решений

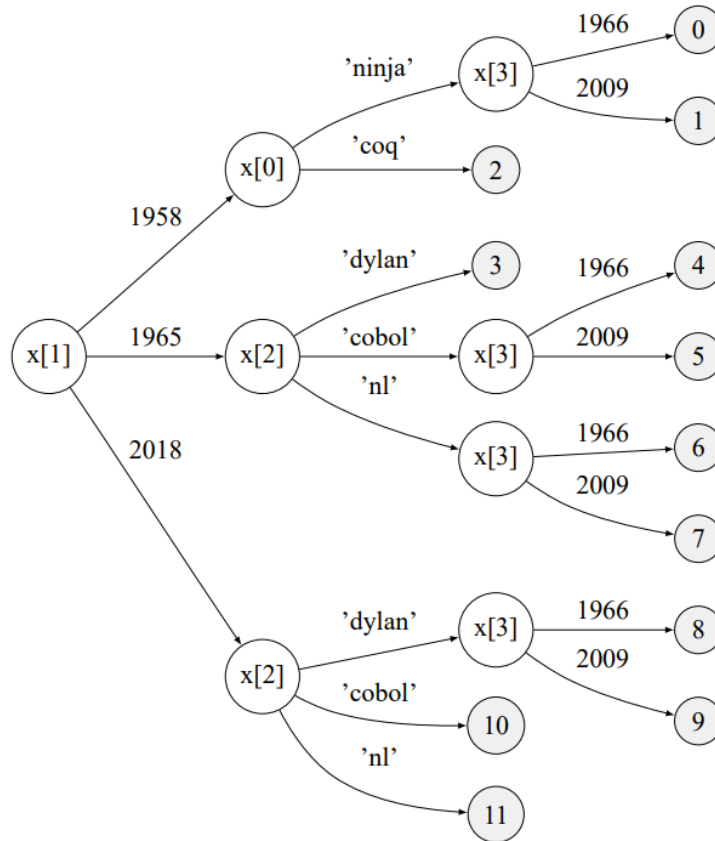


Рис. 1: Дерево решений

*Решение.* Для реализации функции–дерева решений используем условные операторы *if else* для организации вложенного ветвления.

```
def solve_tree (list):
    if list[1] == 1958:
        if list[0] == "ninja":
            if list[3] == 1966:
                return 0
            elif list[3] == 2009:
                return 1
        elif list[0] == "coq":
```

```

        return 2
    elif list[1] == 1965:
        if list[2] == "dylan":
            return 3
        elif list[2] == "cobol":
            if list[3] == 1966:
                return 4
            elif list[3] == 2009:
                return 5
        elif list[2] == "nl":
            if list[3] == 1966:
                return 6
            elif list[3] == 2009:
                return 7
    elif list[1] == 2018:
        if list[2] == "dylan":
            if list[3] == 1966:
                return 8
            elif list[3] == 2009:
                return 9
        elif list[2] == "cobol":
            return 10
        elif list[2] == "nl":
            return 11
    else:
        return None

```

Примеры вычисления дерева решений  $solve_{tree}$ :

1.  $solve_{tree}(['coq', 1958, 'dylan', 2009]) = 2$
2.  $solve_{tree}(['coq', 1965, 'nl', 1966]) = 6$

### Задача 2.2. Реализовать функцию–транскодер

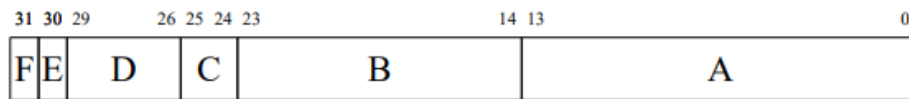


Рис. 2: Представление исходного числа



Рис. 3: Представление преобразованного числа

*Решение.* Для решение данной задачи будем использовать побитовые операции сдвига, логического умножения и логического сложения.

```
def transcoder (value):  
    a = value & 0b1111111111111111  
    b = (value >> 14) & 0b111111111111  
    c = (value >> 24) & 0b11  
    d = (value >> 26) & 0b1111  
    e = (value >> 30) & 0b1  
    f = (value >> 31) & 0b1  
    return (a << 7) | (b << 21) | (c << 5) | (d <<  
        0) | (e << 4) | (f << 31)
```

Примеры вычисления функции транскодер *transcoder*:

1.  $transcoder(0x22b67413) = 0x5b3a09c8$
2.  $transcoder(0x3bf48b61) = 0x7a45b0ee$

**Задача 2.3.** Реализовать функцию преобразования табличных данных. Входная и выходная таблицы заданы в построчной форме, с помощью

списков. Заполненные ячейки имеют строковый тип данных. Пустые ячейки имеют значение None. Над входной таблицей провести ряд преобразований:

- Удалить пустые столбцы
- Удалить пустые столбцы
- Удалить дубли среди строк
- Преобразовать содержимое ячеек по примерам
- Отсортировать строки по столбцу №1.

*Решение.* Для реализации функции–преобразователя таблиц опишем вспомогательные функции, которые будут обрабатывать соответствующие поля исходной таблицы. Применим функциональные идиомы *map*, *filter* для удобного обращения со списками.

```
def phone_formatter (phone_num):  
    f = filter(str.isdigit, phone_num)  
    clear_phone_num = "".join(f)  
    clear_phone_num = clear_phone_num[1:]  
    return clear_phone_num  
  
def number_rounder (number):  
    return round(number, 2)  
  
def date_formatter (date):  
    date_list = date.split('/')  
    short_date = date_list[-1][2:]  
    date_list[2] = short_date  
    date_list = '.'.join(date_list)  
    return date_list
```



```

def erase_none (mylist):
    return list(filter(lambda el: not(el is
        None), mylist))

def unique (not_typed_list):
    n = []
    for i in not_typed_list:
        if i not in n:
            n.append(i)
    return n

def table_formatter (table):
    # table is list of list of string
    def pretty_row (row):
        # row is a list of string
        clean_row = erase_none(row)
        clean_row[0] = phone_formatter(clean_row[0])
        clean_row[1] =
            number_rounder(float(clean_row[1]))
        clean_row[2] = int(1) if bool(clean_row[2])
            else int(0)
        clean_row[3] = date_formatter(clean_row[3])
        return clean_row
    answer = list(map(pretty_row , table))
    answer = list(map(lambda row:
        list(map(str , row)) , answer))
    answer = unique(answer)
    answer.sort(key = operator.itemgetter(0,1))
    return answer

```

Примеры табличных преобразований:

1. Исходная таблица:

	+7 (027) 126-03-36	0.138	false		10/08/2000
	+7 (027) 126-03-36	0.138	false		10/08/2000
	+7 (027) 126-03-36	0.138	false		10/08/2000
	+7 (382) 694-53-15	0.891	false		11/09/1999
	+7 (464) 562-67-40	0.970	false		18/11/1999

Результат преобразования:

0271260336	0.14	0	10.08.00
3826945315	0.89	0	11.09.99
4645626740	0.97	0	18.11.99

2. Исходная таблица:

	+7 (154) 577-06-52	0.173	true		12/08/2003
	+7 (786) 986-44-64	0.852	true		03/08/1999
	+7 (235) 280-67-38	0.119	false		15/03/2001
	+7 (407) 870-35-27	0.730	false		19/11/2004
	+7 (407) 870-35-27	0.730	false		19/11/2004
	+7 (407) 870-35-27	0.730	false		19/11/2004

Результат преобразования:

1545770652	0.17	1	12.08.03
2352806738	0.12	0	15.03.01
4078703527	0.73	0	19.11.04
7869864464	0.85	1	03.08.99

Рис. 4: Форматирование таблиц

### 3 Вывод

В ходе выполнения данной практической работы были приобретены знания по использованию условных и циклических операторов, математических функций в языке программирования Python. Полученные знания были применены на практике.