Санкт-Петербургский национальный исследовательский университет

информационных технологий, механики и оптики
(Университет ИТМО)

Факультет компьютерных технологий и управления

Кафедра вычислительной техники

Лабораторная работа №1

по дисциплине

«Технологии программирования»

Выполнил
студент 4 курса,
группы P3400
Сапожников Борис
Константинович

Руководитель:
кандидат технических наук
Оголюк Александр
Александрович

Санкт-Петербург — 2018

list1.py

```
# 1.
# Вх: список строк, Возвр: кол-во строк
# где строка > 2 символов и первый символ == последнему

def me(words):
	count = 0
	for s in words:
		if len(s)>2 and s[0] == s[-1] :
			count += 1
	return count


# 2.
# Вх: список строк, Возвр: список со строками (упорядочено)
# за искл всех строк начинающихся с 'x', которые попадают в начало списка.
# ['tix', 'xyz', 'apple', 'xacadu', 'aabbbccc'] -> ['xacadu', 'xyz', 'aabbbccc', 'apple', 'tix']

def fx(words)
	return sorted(words, key=lambda s : ('0', s) if s.startswith('x') else ('1', s))

# 3.
# Вх: список непустых кортежей,
# Возвр: список сортир по возрастанию последнего элемента в каждом корт.
# [(1, 7), (1, 3), (3, 4, 5), (2, 2)] -> [(2, 2), (1, 3), (3, 4, 5), (1, 7)]

def sorttuple(words):
	return sorted(words, key=lambda item: item[-1])


def test(func, res, expt):
	print 'Function \''+str(func)+'\' supposed to return: ', expt
	print ('and' if res == expt  else 'but') + ' returns:' , res


if __name__ == '__main__':
	#test 1
	test('me', me(['f', 'oo', 'barb', 'aaaaaa', 'rrr']), 3);
	#test 2
	test('fx', fx(['tix', 'xyz', 'apple', 'xacadu', 'aabbbccc']), ['xacadu', 'xyz', 'aabbbccc',
'apple', 'tix']);
	#test 3
	test('sorttuple', sorttuple([(1, 7), (1, 3), (3, 4, 5), (2, 2)]), [(2, 2), (1, 3), (3, 4,
5), (1, 7)]);
```

string1.py

```
# 1.
# Вх: строка. Если длина > 3, добавить в конец "ing",
# если в конце нет уже "ing", иначе добавить "ly".
def v(s):
	func = lambda s : s + 'ing' if len(s)>3 and s.endswith('ing') else s + 'ly'
	return func(s)

# 2.
# Вх: строка. Заменить подстроку от 'not' до 'bad'. ('bad' после 'not')
# на 'good'.
# Пример: So 'This music is not so bad!' -> This music is good!
def nb(s):
	start = s.find('not')
	stop = s.find('bad', start)
	if start != -1 and stop != -1:
		s=s[:start] + 'good' + s[stop+3:]
	return s


def test(func, res, expt):
  print 'Function \''+str(func)+'\' supposed to return: \'' + expt + '\''
  print ('and' if res == expt  else 'but') + ' returns: \'' + res + '\''


if __name__ == '__main__':
  #main()
  #test1
  test('v', v('ling'), 'lingly')
  #test('num_of_items', num_of_items(3), 'Number of: 3')
  #test2
  test('nb', nb('This music is not so bad!'), 'This music is good!')
```

Вывод:

```
Function 'me' supposed to return:  3
and returns: 3
Function 'fx' supposed to return:  ['xacadu', 'xyz', 'aabbbccc', 'apple', 'tix']
and returns: ['xacadu', 'xyz', 'aabbbccc', 'apple', 'tix']
Function 'sorttuple' supposed to return:  [(2, 2), (1, 3), (3, 4, 5), (1, 7)]
and returns: [(2, 2), (1, 3), (3, 4, 5), (1, 7)]

Function 'v' supposed to return: 'lingly'
and returns: 'lingly'
Function 'nb' supposed to return: 'This music is good!'
and returns: 'This music is good!'
```