

Колледж космического машиностроения и технологий

ОТЧЕТ

По учебной практике УП.01.01 Разработка программных модулей

программного обеспечения для компьютерных систем

специальность 09.02.03 Программирование в компьютерных системах

Выполнили студенты:

Брусник Вадим Виктрович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Мырза Николай Юрьевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

Гусятинер Леонид Борисович

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(подпись)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_(оценка)

Королев, 2020

## **Раздел 1. Техника решения задач с использованием структурного и объектно-ориентированного программирования**

## Установка интерпретатора Python 3

Для установки интерпретатора Python на компьютер, первое, что нужно сделать – это скачать дистрибутив. Загрузить его  можно с официального сайта, перейдя по ссылке <https://www.python.org/downloads/>



Официальный сайт Python

Порядок установки на Windows:

1. Запустить скачанный установочный файл.

2. Выбрать способ установки.



Установщик Python

3. Отметить необходимые опции установки (доступно при выборе Customize installation)



Опции установки

На этом шаге нам предлагается отметить дополнения, устанавливаемые вместе с интерпретатором Python. Выбираю:

* Documentation – установка документаций.
* pip – установка пакетного менеджера pip.
* tcl/tk and IDLE – установка интегрированной среды разработки (IDLE) и библиотеки для построения графического интерфейса (tkinter).

4. Выбираем место установки (доступно при выборе Customize installation)



Продвинутые опции установки

5. После успешной установки:



## Техника работы в командной строке и среде IDLE программами

Выполняя (запуская) команду “python” в вашем терминале, вы получаете интерактивную оболочку Python.



Рисунок 6. Интерактивная оболочка Python

Существует несколько способов закрыть оболочку Python:

>>> exit()

или же

>>> quit()

Кроме того, CTRL + D закроет оболочку и вернет вас в командную строку терминала.

[IDLE](https://docs.python.org/2/library/idle.html) - простой редактор для Python, который поставляется вместе с Python.

Откройте IDLE в вашей системе выбора.

В оболочке есть подсказка из трех прямоугольных скобок:

>>>

Теперь напишите в подсказке следующий код:

>>> print("Hello, World")

Нажмите Enter .

>>> print("Hello, World")

Hello, World



Рисунок 7. Первая программа

## Техника работы с линейными и разветвляющимися программами

Листинг 1. Python

'''

С использованием результата задания 2 разработать программу для нахождения наименьшего

общего кратного

'''

a = int(input())

b = int(input())

i = min(a, b)

while True:

if i%a == 0 and i%b == 0:

break

i += 1

print(i)

Листинг 2.

'''

- input

Функция input() в Python, ввод данных с клавиатура.

https://docs-python.ru/tutorial/vstroennye-funktsii-interpretatora-python/funktsija-input/

'''

strenk = input()

print(strenk)

Листинг 3.

'''

Задание 2. Разработать программу с меню для демонстрации работы с типами данных:

список(list), словарь(dict), множество(set)

Меню -> выбор типа данных -> выбор метода -> краткая справка

'''

options = int(input("Choose from Menu:\n"

"list: 1.\n"

"dict: 2.\n"

"set: 3.\n"))

listMethods = ['list.append(x) Добавляет элемент в конец списка\n',

'list.extend(L) Расширяет список list, добавляя в конец все элементы списка L\n',

'list.insert(i, x) Вставляет на i-ый элемент значение x\n',

'list.remove(x) Удаляет первый элемент в списке, имеющий значение x. ValueError, если такого элемента не существует\n'

'list.pop([i]) Удаляет i-ый элемент и возвращает его. Если индекс не указан, удаляется последний элемент\n'

'list.index(x, [start [, end]]) Возвращает положение первого элемента со значением x (при этом поиск ведется от start до end)\n'

'list.count(x) Возвращает количество элементов со значением x\n'

'list.sort([key=функция]) Сортирует список на основе функции\n'

'list.reverse() Разворачивает список\n'

'list.copy() Поверхностная копия списка\n'

'list.clear() Очищает список\n']

dictMethods = ['dict.clear() - очищает словарь.\n',

'dict.copy() - возвращает копию словаря.\n',

'classmethod dict.fromkeys(seq[, value]) - создает словарь с ключами из seq и значением value (по умолчанию None).\n',

'dict.get(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а возвращает default (по умолчанию None).\n',

'dict.items() - возвращает пары (ключ, значение).\n',

'dict.keys() - возвращает ключи в словаре.\n',

'dict.pop(key[, default]) - удаляет ключ и возвращает значение. Если ключа нет, возвращает default (по умолчанию бросает исключение).\n',

'dict.popitem() - удаляет и возвращает пару (ключ, значение). Если словарь пуст, бросает исключение KeyError. Помните, что словари неупорядочены.\n',

'dict.setdefault(key[, default]) - возвращает значение ключа, но если его нет, не бросает исключение, а создает ключ с значением default (по умолчанию None).\n',

'dict.update([other]) - обновляет словарь, добавляя пары (ключ, значение) из other. Существующие ключи перезаписываются. Возвращает None (не новый словарь!).\n',

'dict.values() - возвращает значения в словаре.\n']

setMethods = ['len(s) - число элементов в множестве (размер множества).\n',

'x in s - принадлежит ли x множеству s.\n',

'set.isdisjoint(other) - истина, если set и other не имеют общих элементов.\n',

'set == other - все элементы set принадлежат other, все элементы other принадлежат set.\n',

'set.issubset(other) или set <= other - все элементы set принадлежат other.\n',

'set.issuperset(other) или set >= other - аналогично.\n',

'set.union(other, ...) или set | other | ... - объединение нескольких множеств.\n',

'set.intersection(other, ...) или set & other & ... - пересечение.\n',

'set.difference(other, ...) или set - other - ... - множество из всех элементов set, не принадлежащие ни одному из other.\n',

'set.symmetric\_difference(other); set ^ other - множество из элементов, встречающихся в одном множестве, но не встречающиеся в обоих.\n',

'set.copy() - копия множества.\n',

'И операции, непосредственно изменяющие множество:\n',

'set.update(other, ...); set |= other | ... - объединение.\n',

'set.intersection\_update(other, ...); set &= other & ... - пересечение.\n',

'set.difference\_update(other, ...); set -= other | ... - вычитание.\n',

'set.symmetric\_difference\_update(other); set ^= other - множество из элементов, встречающихся в одном множестве, но не встречающиеся в обоих.\n',

'set.add(elem) - добавляет элемент в множество.\n',

'set.remove(elem) - удаляет элемент из множества. KeyError, если такого элемента не существует.\n',

'set.discard(elem) - удаляет элемент, если он находится в множестве.\n',

'set.pop() - удаляет первый элемент из множества. Так как множества не упорядочены, нельзя точно сказать, какой элемент будет первым.\n',

'set.clear() - очистка множества.\n']

menuList = [listMethods, dictMethods, setMethods]

print(\*menuList[options - 1])

Листинг 4.

'''

- форматная строка и метод формат

'''

print('Hello, {0}, {1}!'.format(input(), input()))

Листинг 5.

'''

- Задание 1. Разработать программу для печати даты прописью

Пример ввода: 15.12.1983

Пример вывода: Пятнадцатое декабря одна тысяча девятсот восемьдесят третьего года

'''

def get\_date(date):

days = ['первое', 'второе', 'третье', 'четвёртое',

'пятое', 'шестое', 'седьмое', 'восьмое',

'девятое', 'десятое', 'одиннадцатое', 'двенадцатое',

'тринадцатое', 'четырнадцатое', 'пятнадцатое', 'шестнадцатое',

'семнадцатое', 'восемнадцатое', 'девятнадцатое', 'двадцатое',

'двадцать первое', 'двадцать второе', 'двадцать третье',

'двадацать четвёртое', 'двадцать пятое', 'двадцать шестое',

'двадцать седьмое', 'двадцать восьмое', 'двадцать девятое',

'тридцатое', 'тридцать первое']

months = ['января', 'февраля', 'марта', 'апреля', 'мая', 'июня',

'июля', 'августа', 'сентября', 'октября', 'ноября', 'декабря']

date = date.split('.')

return (days[int(date[0]) - 1] + ' ' + months[int(date[1]) - 1] + ' ' + date[2] + ' года')

date = input()

print(get\_date(date))

## Техника работы с циклическими программами, цикл while

Листинг 5.

'''

Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу

в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

'''

sum = 0

a=int(input())

while a !=0:

sum +=a

a=int(input())

print(sum)

Листинг 6.

'''

Напишите программу, которая выводит часть последовательности 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ...

(число повторяется столько раз, чему равно).

На вход программе передаётся неотрицательное целое число n — столько элементов

последовательности должна отобразить программа.

На выходе ожидается последовательность чисел, записанных через пробел в одну строку.

Например, если n = 7, то программа должна вывести 1 2 2 3 3 3 4.

'''

n = int(input())

a = []

i = 1

while len(a) < n:

a +=[i] \* i

i +=1

print(\*a[:n])

Листинг 7.

'''

https://stepik.org/lesson/3364/step/11?unit=947

Напишите программу, которая считывает со стандартного ввода целые числа, по одному числу

в строке, и после первого введенного нуля выводит сумму полученных на вход чисел.

'''

b = int(input())

s = b

while b != 0:

c = int(input())

s += c

b = c

print(s)

Листинг 8.

'''

С использованием результата задания 2 разработать программу для нахождения наименьшего

общего кратного

'''

a = int(input())

b = int(input())

i = min(a, b)

while True:

if i%a==0 and i%b==0:

break

i += 1

print(i)

## Техника работы с числами

Листинг 9.

'''

Задание 1. Составить и выполнить по 3 примера использования модулей для работы

с дробными числами (fractions), для точных вычислений (decimal).

'''

from decimal import Decimal

from fractions import Fraction

number = Decimal("0.444")

print(number)

number = number.quantize(Decimal("1.00"))

print(number)

number = Decimal("0.555678")

print(number.quantize(Decimal("1.00")))

number = Decimal("0.999")

print(number.quantize(Decimal("1.00")))

print(Fraction(1, 3))

print(Fraction(3.1415))

a = Fraction(1, 7)

b = Fraction(1, 3)

print(a + b)

Листинг 10.

'''

https://pythontutor.ru/lessons/lists/problems/more\_than\_neighbours/

Задача «Больше своих соседей»

Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух

своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда

не учитываются, поскольку у них недостаточно соседей.

'''

arr = []

size = int(input())

count = 0

for i in range(size):

arr.append(int(input()))

for i in range(size):

if i > 0 and i < size - 1:

if arr[i - 1] < arr[i] and arr[i] > arr[i + 1]:

count += 1

print(count)

## Техника работы со строками

Листинг 11.

'''

?Уберите точки из введенного IP-адреса. Выведите сначала четыре числа через пробел,

а затем сумму получившихся чисел.

Sample Input: 192.168.0.1

Sample Output:

192 168 0 1

361

'''

string = input().replace('.', ' ')

print(string)

string = string.split()

print(int(string[0]) + int(string[1]) + int(string[2]) + int(string[3]))

Листинг 12.

'''

?Известно, что для логина часто не разрешается использовать строки содержащие пробелы.

Но пользователю нашего сервиса особенно понравилась какая-то строка.

Замените пробелы в строке на символы нижнего подчеркивания, чтобы строка

могла сгодиться для логина. Если строка состоит из одного слова, менять ничего не нужно.

Sample Input: python sila

Sample Output: python\_sila

'''

print(input().replace(' ', '\_'))

## Техника работы со списками

Листинг 12.

'''

Задание 1. Array55. Дан целочисленный массив A размера N (<= 15). Переписать в новый целочисленный

массив B все элементы с нечетными порядковыми номерами (1, 3, ...) и вывести размер

полученного массива B и его содержимое. Условный оператор не использовать.

'''

n = int(input())

a = [int(input()) for \_ in range(n)]

b = [x for x in a if x % 2 == 1]

print(len(b))

print(\*b)

Листинг 13.

'''

Задание 2. Array57. Дан целочисленный массив A размера N. Переписать в новый целочисленный массив B

того же размера вначале все элементы исходного массива с четными номерами,

а затем — с нечетными:

A[0], A[2], A[4], A[6], ..., A[1], A[3], A[5], ... .

Условный оператор не использовать.

'''

n = int(input())

a = [int(input()) for \_ in range(n)]

b = []

for i in range(0,n,2):

b.append(a[i])

for i in range(1,n,2):

b.append(a[i])

print(\*b)

Листинг 14.

'''

Задание 2. Array57. Дан целочисленный массив A размера N. Переписать в новый целочисленный массив B

того же размера вначале все элементы исходного массива с четными номерами,

а затем — с нечетными:

A[0], A[2], A[4], A[6], ..., A[1], A[3], A[5], ... .

Условный оператор не использовать.

'''

n = int(input())

a = [int(input()) for \_ in range(n)]

b = []

for i in range(0,n,2):

b.append(a[i])

for i in range(1,n,2):

b.append(a[i])

print(\*b)

Листинг 14.

'''

Задание 3. Array58. Дан массив A размера N. Сформировать новый массив B того же размера по

следующему правилу: элемент B[K] равен сумме элементов массива A с номерами от 0 до K.

'''

n = int(input())

a = [int(input()) for \_ in range(n)]

b=[]

sums = 0

for i in range(0,len(a)):

sums += a[i]

b.append(sums)

print(\*b)

Листинг 15.

'''

Array114. Дан массив A размера N. Упорядочить

его по возрастанию методом сортировки простыми вставками:

сравнить элементы A0 и A1 и, при необходимости меняя их

местами, добиться того, чтобы они оказались упорядоченными

по возрастанию; затем обратиться к элементу A2 и

переместить его в левую (уже упорядоченную) часть массива,

сохранив ее упорядоченность; повторить этот процесс для

остальных элементов, выводя содержимое массива после

обработки каждого элемента (от 1-го до N-1 го).

'''

'''

10

10

9

8

7

6

5

4

3

2

1

'''

arr = []

size = int(input())

for i in range(size):

arr.append(int(input()))

for i in range(size - 1):

for j in range(size - 1 - i):

MIN, MAX = min(arr[j], arr[j + 1]), max(arr[j], arr[j + 1])

arr[j], arr[j + 1] = MIN, MAX

print(arr)

Листинг 16.

'''

Array114. Дан массив A размера N. Упорядочить

его по возрастанию методом сортировки простыми вставками:

сравнить элементы A0 и A1 и, при необходимости меняя их

местами, добиться того, чтобы они оказались упорядоченными

по возрастанию; затем обратиться к элементу A2 и

переместить его в левую (уже упорядоченную) часть массива,

сохранив ее упорядоченность; повторить этот процесс для

остальных элементов, выводя содержимое массива после

обработки каждого элемента (от 1-го до N-1 го).

'''

'''

10

10

9

8

7

6

5

4

3

2

1

'''

arr = []

size = int(input())

for i in range(size):

arr.append(int(input()))

for i in range(size - 1):

for j in range(size - 1 - i):

MIN, MAX = min(arr[j], arr[j + 1]), max(arr[j], arr[j + 1])

arr[j], arr[j + 1] = MIN, MAX

print(arr)

Листинг 17.

'''

Array112. Дан массив A размера N.

Упорядочить его по возрастанию методом сортировки

простым обменом («пузырьковой» сортировкой):

просматривать массив, сравнивая его соседние элементы

(A0 и A1, A1 и A2 и т. д.) и меняя их местами,

если левый элемент пары больше правого; повторить описанные

действия N 1 раз. Для контроля за выполняемыми действиями

выводить содержимое массива после каждого просмотра.

Учесть, что при каждом просмотре количество анализируемых

пар можно уменьшить на 1.

'''

N = int(input())

array = []

for x in range(N):

array.append(int(input()))

print(\*array)

for i in range(N - 1):

for j in range(N - i - 1):

if array[j] > array[j + 1]:

array[j], array[j + 1] = array[j + 1], array[j]

print(\*array)

print("\n Итог:")

print(\*array)

Листинг 18.

'''

Matrix88. Дана квадратная матрица порядка M. Обнулить элементы матрицы,

лежащие ниже главной диагонали. Условный оператор не использовать.

4

4

1

1

1

1

4

4

4

4

2

2

2

2

3

3

3

3

'''

# в первой строке ввода идёт количество строк массива

m, n = int(input()), int(input())

a = []

for i in range(m):

b = []

for j in range(n):

b.append(int(input()))

a.append(b)

for i in range(m):

for j in range(n - i, n):

a[i][j] = 0

for i in range(m):

print(\*a[i])

Листинг 19.

'''

Задание 4. Matrix3. Даны целые положительные числа M, N и набор из M чисел. Сформировать

матрицу размера M x N, у которой в каждом столбце содержатся все числа из исходного

набора (в том же порядке).

'''

# в первой строке ввода идёт количество строк массива

m = int(input())

b = []

for i in range(m):

b.append(int(input()))

print("NEXT\n")

n = int(input())

a = []

for i in range(m):

a.append([0] \* n)

for i in range(m):

for j in range(n):

a[i][j] = b[i]

print(\*a)

Листинг 20.

'''

Задание 1. (Л.Б.) Для проведения конкурса проектов в ККМТ формируются группы

из 4х участников: coder, writer, tester, designer, программирующих

на одном и том же языке.

Каждый студент может программировать только на одном языке

и занимать только одну позицию.

Дан текстовый файл, содержащий перечень студентов с указанием языка и позиции

(каждый студент с новой строки)

Требуется

1. Получить список студентов с указанием языка и позиции

2. Сформировать список всевозможных команд

3. Вывести список команд с указанием состава и названия команды:

Команда 1

coder: ...

designer: ...

tester: ...

writer: ...

Команда 2

...

Пункты 1 и 2 выполнить с использованием генераторов списка

'''

listOfPosition = ['coder:', 'designer:', 'tester:', 'writer:']

listOf\_FirstTeam = []

count = 0

s = str

i = 0

f = open('file.txt', 'r')

with open('file.txt', 'r') as file:

for line in file:

s = line

s = s.split()

position = s[0]

for i in range(4):

if position == listOfPosition[i]:

listOf\_FirstTeam.append(s)

print(\*listOf\_FirstTeam)

print(count)