

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Московский государственный технологический университет «СТАНКИН»**

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

|  |  |
| --- | --- |
| **Институт**  **информационных**  **систем и технологий** | **Кафедра**  **информационных систем** |

**Основная образовательная программа 09.03.02  
«Информационные системы и технологии»**

**Отчёт**

Выполнила:

Плотникова Д. А.

Проверил:

Кайшев Д.А.

Задания:

Написать функцию, которая на вход принимает int и возвращает true или false в зависимости является ли это число палиндром. Число является палиндромом, если оно читается справа налево и слева направо одинаково (25)

Написать функцию, которая принимает на вход список из положительных целочисленных элементов и возвращает три списка: (25)

* 1. в первом - числа, которые делятся на 2
  2. во втором - числа, которые делятся на 3
  3. с третьем - числа, которые делятся на 5

Написать функцию, принимающую на вход int, и число, обратное этому int (25)

Написать функцию, которая будет расчитывать квадратный корень n-ой степени методом Ньютона ([https://ru.wikipedia.org/wiki/Алгоритм\_нахождения\_корня\_n-ной\_степени](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC_%D0%BD%D0%B0%D1%85%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F_%D0%BA%D0%BE%D1%80%D0%BD%D1%8F_n-%D0%BD%D0%BE%D0%B9_%D1%81%D1%82%D0%B5%D0%BF%D0%B5%D0%BD%D0%B8)) (35)

Написать функцию, принимающую 1 аргумент — число от 0 до 100000, и возвращающую true, если оно простое, false если нет. (35)

На 45 нужно решить все задачи на 25 и 35

Написать декоратор, который будет кэшировать результат вызова функции и отдавать его при последующих вызовах данной функции (для тех, кто был на семинаре, но не обязательно - можете посмотреть как работают декораторы, 50).

Решение:

import functools

import time

def cache(func): #кэширующий декоратор

@functools.wraps(func) #обертка функции

def wrapper(\*args, \*\*kwargs):#функция проверки и добавления в кэш результата работы(обертка)

cache\_key = args + tuple(kwargs.items())#как ключ к кэшу мы записываем аргументы функции

if cache\_key not in wrapper.cache:#если ключа нет в кэше

wrapper.cache[cache\_key] = func(\*args, \*\*kwargs)#записываем результат работы функции в кэш

return wrapper.cache[cache\_key]#как результат обертки вызываем значение функции

wrapper.cache = dict()#кэш обертки - словарь

return wrapper# результат работы декоратора - результат работы обертки

@cache#прописываем декоратор для функции

def newton(a, x, n, accuracy=0.0000000000000001):#число, приближение, степень, точность

xn = 1/n \* ((n-1)\*x + a/(pow(x, (n-1))))# формула Ньютона для вычисления нового приближения

if ((x-xn) <= accuracy):#проверка на точность

return xn# если проверка прошла - заканчиваем работу

else:#

return newton(a, xn, n)#если нет вызываем рекурсию

def IsPrime(n):#проверка на простоту

d = 2#

while n % d != 0:#делим исходное число на все до него

d += 1#инкримент если есть остаток

return d == n#если числа равны - число простое, если нет, то нет

def Reverse(n):# воспроизводим число наоборот

x = str(abs(n))#переводим модуль числа в строку

s = ""#

for a in range(len(x)):#разбиваем на символы

s += str(x[((len(x)-1)-a)])#воспроиводим наоборот

if (n < 0):# если исходное число отрицательное, добавляем минус

return -int(s)#

return int(s)#возвращаем число

def palindrom(n):#

return n==Reverse(n)#используем предыдущую функцию чтобы проверить палиндром

def lists(x = list()):#передаем список

s = list()#назначаем списки

a = list()#

v = list()#

for i in range(len(x)):#проходим по исходному и по проверке делимости распределяем между списками

if ((x[i] % 2) == 0):#

s.append(x[i])#

if ((x[i] % 3) == 0):#

a.append(x[i])#

if ((x[i] % 5) == 0):#

v.append(x[i])#

return s, a, v#возвращаем списки

#выводим результаты

print(newton(8,8,3))

print(IsPrime(99331))

print(Reverse(-3314434))

print(palindrom(123))

print(lists(list([3,4,5,6,7,8, 787, 555])))

print(newton(8,8,3))

start = time.perf\_counter() #проверки скорости работы с кэшем и без

print('Time run:', time.perf\_counter() - start)