

**МИНОБРНАУКИ РОССИИ**

**федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

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

**(ФГБОУ ВО «МГТУ «СТАНКИН»)**

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

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

**Отчет по дисциплине «Веб-программирование»**

**по лабораторной работе №2**

| Студент  группы ИДБ-19-05 | Михайлуца Н.А. |
| --- | --- |
|  |  |
| Преподаватель | Кайшев Д.А. |

Ссылка на лабораторную- <https://github.com/Yyrrr/2LR-WEB>

Задание №1.

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.

Код решения представлен ниже.

def func(chislo):

temp = chislo

perevernytoe =0

while (temp!=0):

perevernytoe = (perevernytoe \*10) + (temp % 10)

temp=temp//10

if(chislo==perevernytoe):

return True

else:

return False

x = int(input("введи число: "))

if(func(x)==True):

print("Палиндром")

else:

print("Не палиндром")

Задание №2.

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

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

Код решения представлен ниже.

def fin(list,colvo):

list2 = []

list3 = []

list5 = []

for i in range(0,colvo,1):

if list[i]%2==0:

list2.append(list[i])

if list[i]%3==0:

list3.append(list[i])

if list[i]%5==0:

list5.append(list[i])

return "Делятся на два",list2,"Делятся на три",list3,"Делятся на пять",list5

print("Введите количество элементов:")

colvo= int(input())

list = []

for i in range(0,colvo,1):

print("Введите ", i ," элемент:")

list.append(int(input()))

print(fin(list,colvo))

Задание №3.

Написать функцию, принимающую на вход int, и число, обратное этому int.

Код решения представлен ниже.

def func(chislo):

temp = abs(chislo)

perevernytoe =0

while (temp!=0):

perevernytoe = (perevernytoe \*10) + (temp % 10)

temp=temp//10

if(chislo<0):

perevernytoe=perevernytoe\*-1

return perevernytoe

chislo = int(input("введи число:"))

print(func(chislo))

Задание №4.

Написать функцию, которая будет рассчитывать квадратный корень n-ой степени методом Ньютона.

Код решения представлен ниже.

def func(chislo,stepen):

a=1#начальное число

b=(1/stepen)\*((stepen-1)\*a+(chislo/(a\*\*(stepen-1))))# следующее число

while(abs(b-a)>0.1):

a=b

b=(1/stepen)\*((stepen-1)\*a+(chislo/(a\*\*(stepen-1))))

return b

chislo = int(input("введи число:"))

stepen = int(input("введи степень:"))

print(func(chislo,stepen))

Задание №5.

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

Код решения представлен ниже.

def fin(chislo):

k=0

for i in range(2, (chislo//2)+1, 1):

if (chislo % i == 0):

return False

return True

chislo= int(input("Введите число:"))

if (fin(chislo)==True):

print("Простое")

else:

print("Сложное")

Задание №6.

Написать декоратор, который будет кэшировать результат вызова функции и отдавать его при последующих вызовах данной функции. Усложненный вариант - написать тот же самый декоратор, но с параметром, который будет показывать сколько раз отдавать кэшируемый результат. Если данный счетчик обнуляется, то выполняем функцию и вновь кэшируем ее результат.

Данный код кэширует только одно предыдущее значение.

b=0

с=0

def dec(func):# верх

def wrapper(food):

a=(func(food))

global b

global с

if(с!=0):

print("Старое значение:",a," Новое значение:",b)

else:

print("Функция запущена первый раз, кеш пуст",a)

с=1

b=a

return wrapper

@dec

def fun(chislo):

return(chislo)

fun(5)

fun(6)

fun(7)

fun(0)

fun(7)

Данный код кэширует все значения.

b=[]

def dec(func):# верх

def wrapper(fy):

a=(func(fy))

global b

global с

if(len(b)!=0):

print("Новое значение:",a," Старое значение:",b)

b.append(a)

else:

print("Функция запущена первый раз, кэш пуст",a)

b.append(a)

return wrapper

@dec

def fun(chislo):

return(chislo)

fun(5)

fun(6)

fun(7)

fun(0)

fun(7)

fun(8)

Данный код кэширует данные с параметром.

c = []

counter = []

n = 0

def dec(function):

def wrapper(chislo):

if len(c) != 0 :

if counter[0] == 0 :

c.pop(0)

counter.pop(0)

print (c)

for i in range (len(c)):

counter[i] -= 1

c.append(function(chislo))

counter.append(n)

return wrapper

@dec

def fun(chislo):

return chislo

n = int(input("Как долго хранить кэш?"))

fun(1)

fun(2)

fun(3)

fun(4)

fun(5)

fun(6)

**Выводы:** в данной лабораторной работе мы научились создавать простые функции на языке python, изучили синтаксис данного языка и применили полученные знания на практике.