



**МИНОБРНАУКИ РОССИИ**  
**федеральное государственное бюджетное образовательное учреждение**  
**высшего образования**  
**«Московский государственный технологический университет «СТАНКИН»**  
**(ФГБОУ ВО МГТУ «СТАНКИН»)**

---

**Институт**  
информационных технологий

**Кафедра**  
информационных систем

**Отчет по лабораторной работе № 2**  
**по дисциплине «Веб-программирование»**

**Студентка**  
группа ИДБ–19–05

**Ле Тхи Хань Хуен**

**Преподаватель**

---

подпись

**Кайшев Д.А.**

---

подпись

Москва 2021 г.

## 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:

```
#Main function handle requirement
i=0
def func(inValue, len):
    slicedString=str(inValue)[len::-1]
    if int(slicedString)==inValue:
        print("TRUE!")
    else:
        print("FALSE!")
#Enter Number
while i<10:
    while True:
        try:
            valInput=int(input("Enter your number(>9): "))
            if len(str(valInput))<=1 or valInput <1:
                print("Your input value have to be a number greater than 9! Please enter
again!")
            i=0
        else:
            i=20
            break
    except ValueError:
        print ("Your input value is not int! Please enter again!")
        i=0
func(valInput,len(str(valInput)))
```

### Результат:

```
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2> cd PythonApplication1
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe1.py
Enter your number(>9): 123
FALSE!
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe1.py
Enter your number(>9): 353
TRUE!
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> █
```

## ЗАДАНИЕ 2

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

- а) в первом - числа, которые делятся на 2
- б) во втором - числа, которые делятся на 3
- с) с третьем - числа, которые делятся на 5

### Код:

```
firstList = []
secondList = []
thirdList = []
inputList=[1,2,3,4,5,6]
#Main function hadle requirement
def func(lIn):
    for x in lIn:
        if x%2 == 0:
            firstList.append(x)
        elif x%3 == 0 :
            secondList.append(x)
        elif x%5 == 0:
            thirdList.append(x)
func(inputList)
print("First List : ", firstList)
print("Second List : ", secondList)
print("Third List : ", thirdList)
```

### Результат:

```
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2> cd PythonApplication1
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe2.py
First List : [2, 4, 6]
Second List : [3]
Third List : [5]
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> █
```

## ЗАДАНИЕ 3

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

### Код:

```

#Main function handle requirement
def func(valInputFunc):
    slicedString=str(valInput)[len(str(valInput)):-1]
    output=""
    if valInput < 0:
        slicedString = slicedString[:-1]
        output = int("-"+slicedString[:1]+slicedString[1:])
    else:
        output = int(slicedString)
    print("output: ",output)
#Enter number
i=0
valInput=0
while i<10:
    while True:
        try:
            valInput=int(input("Enter your number: "))
            i=10
            break
        except ValueError:
            print ("Your input value is not int! Please enter again!")
            i=0
func(valInput)

```

## Результат:

```

PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe3.py
Enter your number: 0
output: 0
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe3.py
Enter your number: 15
output: 51
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe3.py
Enter your number: 420
output: 24
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe3.py
Enter your number: -247
output: -742
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe3.py
Enter your number: 138
output: 831
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1>

```

#### ЗАДАНИЕ 4

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

**Код:**

```
import math
#Handle input data, make sure data input is number
def enter_data(nameData):
    i=0
    while i<10:
        while True:
            try:
                Input=float(input("Enter "+nameData+" = "))
                i=20
                break
            except ValueError:
                print ("Your input value is not number! Please enter again!")
                i=0
        return Input
#Main function
#Method Newton
def findx_with_method_Newton(nIn, AIn,EpsIn):
    x=1
    while True:
        xk=(1/nIn)*((nIn-1)*x+(AIn/math.pow(x,nIn-1)))
        if abs(x-xk)<EpsIn:
            break
        x=xk
    return x
#Enter data
allowEnterA=False
n=0
A=0
eps=math.pow(10,-10)
#Enter n
while True:
    n=enter_data("n (n>=0)")
    if n<0:
        print("n have to be greater than 0. Please enter n again!")
```

```

elif n==0:
    print("while n=0, A always is 1, can not find x!")
    quit()
else:
    allowEnterA = True
    break
#Enter A
if(allowEnterA):
    while True:
        if n%2==0:
            A = enter_data("A (A=0 or A>1)")
            if A==0 or A>1:
                break
            else:
                print("While n = ",n," A have to be greater than or equal by 0. Please
enter A again!")
        else:
            A = enter_data("A (A>1)")
            if A>1:
                break
            else:
                print("While n = ",n," A have to be greater than 1. Please enter A
again!")
#Output
print("x = ",round(findx_with_method_Newton(n, A,eps),3))

```

## Результат:

```

PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2> cd PythonApplication1
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe4.py
Enter n (n>=0) = 2
Enter A (A=0 or A>1) = 9
x = 3.0
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe4.py
Enter n (n>=0) = 4
Enter A (A=0 or A>1) = 256
x = 4.0
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe4.py
Enter n (n>=0) = 7
Enter A (A>1) = 450
x = 2.393
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe4.py
Enter n (n>=0) = 0
while n=0, A always is 1, can not find x!
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> █

```

## ЗАДАНИЕ 5

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

### Код:

```
#Main function
def func(valForCheckPrime):
    if valForCheckPrime<2:
        return False
    else:
        for i in range(2, valForCheckPrime):
            if valForCheckPrime%i==0:
                return False
        return True
#Enter Number
i=0
valInput=0
while i<10:
    while True:
        try:
            valInput=int(input("Enter your number(from 0 to 100000): "))
            if valInput<0 or valInput >100000:
                print("Your input value have to be a number in range [0 ; 100000]!
Please enter again!")
                i=0
            else:
                i=20
                break
        except ValueError:
            print ("Your input value is not int! Please enter again!")
            i=0
    print("Number is prime! : ", func(valInput))
```

### Результат:

```
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2> cd PythonApplication1
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe5.py
Enter your number(from 0 to 100000): 269
Number is prime! : True
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe5.py
Enter your number(from 0 to 100000): 1105
Number is prime! : False
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> █
```

## ЗАДАНИЕ 7

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 Усложненный вариант - написать тот же самый декоратор, но с параметром, который будет показывать сколько раз отдавать кешируемый результат. Если данный счетчик обнуляется, то выполняем функцию и вновь кешируем ее результат.

### Код:

```
import datetime
import random
#Class save data on cache
class MyCache:
    def __init__(self):
        #constructor
        self.cache={}
        self.max_cache_size=10

    def update(self, key, value):
        """
        Update the cache dictionary and optionally remove the oldest item
        """
        if key not in self.cache and len(self.cache)>=self.max_cache_size:
            self.remove_oldest()

        self.cache[key] = {'date_accessed' : datetime.datetime.now(), 'value' : value}

    def remove_oldest(self):
        """
        Remove the entry that has the oldest accessed date
        """
        oldest_entry=None
        for key in self.cache:
            if oldest_entry==None:
                oldest_entry=key
```



```

        elif
self.cache[key]['date_accessed']<self.cache[oldest_entry]['date_accessed']:
        oldest_entry=key

        self.cache.pop(oldest_entry)

def show_keys_Values(self):

    for k,v in self.cache.items():
        print("=>" +k,v)

@property
def size(self):
    return str(len(self.cache))

#Decorator function
def decor_func_adding_name(func):
    def wrapper(info, name):
        return "Added : " + name + " - " +format(func(info))
    return wrapper

@decor_func_adding_name
def show_group(group):
    return "group : " + format(group)

#Enter Data
cache=MyCache()
while True:
    print("1. Add information \n2. Quit")
    choose=input("Enter number you choose : ")
    if choose=='1':
        name_input = input("Enter your name : ")
        group_input = input("Enter your group : ")
        print("Cache size : ", cache.size)
        print("===== Data in cache : =====")
        cache.show_keys_Values()
        print (show_group(group_input, name_input))
        cache.update(name_input, group_input)
    elif choose=='2':
        break

```

## Результат:

```
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2> cd PythonApplication1
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> py Exe7.py
1. Add information
2. Quit
Enter number you choose : 1
Enter your name : Huyen
Enter your group : IDB-19-05
Cache size : 0
=====Data in cache : =====
Added : Huyen - group : IDB-19-05
1. Add information
2. Quit
Enter number you choose : 1
Enter your name : Anna
Enter your group : ADB-19-01
Cache size : 1
=====Data in cache : =====
>Huyen {'date_accessed': datetime.datetime(2021, 10, 11, 18, 47, 40, 248543), 'value': 'IDB-19-05'}
Added : Anna - group : ADB-19-01
1. Add information
2. Quit
Enter number you choose : 1
Enter your name : Masha
Enter your group : IDB-19-03
Cache size : 2
=====Data in cache : =====
>Huyen {'date_accessed': datetime.datetime(2021, 10, 11, 18, 47, 40, 248543), 'value': 'IDB-19-05'}
>Anna {'date_accessed': datetime.datetime(2021, 10, 11, 18, 48, 9, 693398), 'value': 'ADB-19-01'}
Added : Masha - group : IDB-19-03
1. Add information
2. Quit
Enter number you choose : 2
PS E:\STANKIN\semester 5\Веб-программирование\Lab2\Project 2\PythonApplication1> █
```