**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**

**“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ”**

##### ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ

# кафедра фізико–технічних засобів захисту інформації

# Програмування 4

# Лабораторна робота № 5

# Використання функцій

#### Виконав студент гр. ФЕ-81

#### Макарцов А. О.

Київ,

2020

**Мета роботи:** Оволодіння практичними навичками у використанні функцій

Завдання:

А)17) Дано парне число ; перевірити для цього гіпотезу Гольбаха. Ця гіпотеза (на сьогоднішній день не спростована та повністю не доведена) полягає в тому, що кожне парне число , що більше 2, можна представити у вигляді суми двох простих чисел. (Визначити функцію, що дозволяє розпізнавати прості числа.)

Б)Піднести до додатного цілого степеня дійсне ненульове число.

Код реалізації:

import math

################################################

# Part A

###############################################

def create\_primes\_list(num,lst, pos):

if lst[pos] > math.sqrt(num) + 1: #All prime divisors less than sqrt(num)

return lst

else:

lst = list(n for n in lst if (n % lst[pos] != 0 or n == lst[pos]))

pos += 1

return create\_primes\_list(num,lst,pos)

def list\_in\_file(lst, fl): #File is exists and was opened

for i in lst:

fl.write(str(i) + ',')

def file\_to\_list(fl): # returns int`s list

for line in fl:

temp = line.split(',')

return list(int(n) for n in temp if bool(n) == True)

def get\_primes(num):

try:

primes\_file = open('primes','r')

primes = file\_to\_list(primes\_file)

primes\_file.close()

return primes

except FileNotFoundError:

print('Creating primes list...Wait!\nNext time it is must be faster')

primes = create\_primes\_list(num, list(range(2,num)), 0)

primes\_file = open('primes', 'w')

list\_in\_file(primes, primes\_file)

primes\_file.close()

return primes

def get\_answer(num, primes\_list):

base\_prime = primes\_list[0]

currnet\_index = 0

while base\_prime < num:

poss\_prime = num - base\_prime

if poss\_prime in primes\_list:

return(base\_prime, poss\_prime)

else:

currnet\_index += 1

base\_prime = primes\_list[currnet\_index]

def valid\_n(n):

try:

int(n)

except ValueError:

print('Bad input !')

return False

if int(n) <= 2 or int(n) > 1e+6:

print('n is in bad domain !\n(2 < n < 1000000)')

return False

if int(n) % 2 != 0:

print('n must be even !')

return False

return True

################################################

# Part B

###############################################

def valid\_power(power):

try:

int(power)

except ValueError:

print('Bad input !')

return False

if int(power) < 0:

print('Power must be >= 0')

return False

if int(power) > 200:

print('Power must be < 200 !')

return True

def valid\_num(num):

try:

float(num)

except ValueError:

print('Bad input !')

return False

if float(num) == 0:

print('Num = 0 !')

return False

if float(num) < -10000 or float(num) > 10000:

print('Num is in bad domain !')

return False

return True

def pow\_(num, power, base):

if power == 1:

return num

elif power == 0:

return 1

else:

return pow\_(num \* base, power - 1, base)

def main():

print("###########\nPart A\n###########")

primes = get\_primes(1000000)

n = input('n = ')

if valid\_n(n) == True:

n = int(n)

a,b = get\_answer(n, primes)

print('%i = %i + %i' %(n, a, b))

print("###########\nPart B\n###########")

num = input('Num = ')

power = input('Power = ')

if valid\_num(num) == True and valid\_power(power) == True:

num = float(num)

power = int(power)

try:

print('%f^%i = %f' % (num, power, pow\_(num,power,num)))

except:

print('Something went wrong, try again.')

if \_\_name\_\_ == '\_\_main\_\_':

main()

Приклад виконання програми:

