Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

**МЕХАНІКО-ТЕХНОЛОГІЧНИЙ ФАКУЛЬТЕТ**

Кафедра кібербезпеки та програмного забезпечення

**Звіт**

з виконаної лабораторної роботи № 1

Дисципліна: Комп`ютерна логiка

Виконав :

студент академічної групи КІ-15

Аннаєв А. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Перевірив :

Викладач

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Кропивницкий- 2018

**Лабараторна работа № 1**

Завдання №1

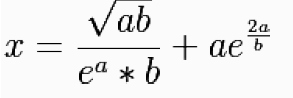
Ваша програма має бути записана в окремому файлі, який може виконуватися консольною командою python <ім'я файлу>.py. Результат обчислень має виводитися на екран командою print.

Кожна програма має сприймати довільні значення, які задовольняють описаному в завданні формату, і передаються у вигляді аргументів командного рядка при виклику програми: python <ім'я файлу>.py arg1 arg2 arg3

Формат введення даних та виведення результатів має ТОЧНО збігатися із вказаним.

Складіть код, перевірте його на різних прикладах. Зверніть увагу, що тестові приклади до завдань лише демонструють логіку роботи та показують формат даних.

**1)Вхідні дані**: 2 невід'ємних дійсних числа a та b -- аргументи командного рядка. b не дорівнює 0.

**Вихідні дані**: дійсне число -- результат обчислення формули

Task\_1 :

def task\_1(a, b):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + str(a) + ' ' + str(b));

# расчет по форму Задания -1

W = (math.sqrt((a\*b))/((math.pow(math.e, a))\*b) + a \*(math.pow(math.e,((2\*a)/b))));

return W;

Task\_2 :

def task\_2(x,y,z):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + str(x) + ' ' + str(y) + ' ' + str(z));

# для читабельности формула разделена на две части

first\_part = ( 1 / (z \*(math.sqrt (2 \* math.pi ) )));

second\_part = ( math.exp (- (pow((x - y),2))/(2\*pow(z,2))));

return first\_part \* second\_part;

Task\_3 :

def task\_3(x,y,z):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + str(x) + ' ' + str(y) + ' ' + str(z));

song = "Everybody sing a song: ";

first\_arg = None;

second\_arg = '';

result = None;

if x > 1:

first\_arg = "la-" \* ( x - 1);

first\_arg = first\_arg + "la ";

else:

first\_arg = "la ";

if y > 0:

second\_arg = first\_arg \* y;

else:

second\_arg = ' ';

index = len(second\_arg) - 1;

if z == 1:

result = second\_arg[:index] + "!";

else:

result = second\_arg[:index] + ".";

return song + result;

Test :

arg\_count = len(sys.argv) - 1; # определение количества аргументов не учитывая имени файла

result = None; # результат выполнения алгоритма

if arg\_count == 2: # если введены два аргумента значит у нас первое задание

a = float(sys.argv[1]); # получение первого аргумента

b = float(sys.argv[2]); # получение второго аргумента

if a >= 0 and b > 0: # проверка по условию задания

result = task\_1(a,b);

else:

print("Некоррекные входные данные");

elif arg\_count == 3: # если введены три аргумента значит второе или третье задание

x = float(sys.argv[1]);

y = float(sys.argv[2]);

z = float(sys.argv[3]);

if x >= 0 and y >= 0 and (z == 0 or z == 1):

result = task\_3(int(x),int(y),int(z));

else:

result = task\_2(x,y,z);

print("Результат : " + str(result));

Завдання №2

**1)Вхідні дані**: 3 дійсних числа a, b, c. Передаються в програму як аргументи командного рядка.

**Результат роботи**: рядок "triangle", якщо можуть існувати відрізки з такою довжиною та з них можна скласти трикутник, або "not triangle" -- якщо ні.

Task\_1 :

def task\_1(a,b,c):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + str(a) + ' ' + str(b) + ' ' + str(c));

# проверка аргументов на соответсвиее требованию

if (a + b) > c and (a + c) > b and (b + c) > a:

return "triangle";

else:

return "not triangle"

Task\_2 :

def task\_2(n):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + str(x));

# динамический перерасчет

a,b = 0,1;

for i in range(n):

a,b = b, a + b;

return a;

Test :

arg\_count = len(sys.argv) - 1; # определение количества аргументов не учитывая имя файла

result = None; # результат выполнения алгоритма

if arg\_count == 1: # если один аргумент то задание -2

n = int(sys.argv[1]); # получаем целое число

result = task\_2(n); # вычисляем результат

elif arg\_count == 3: # если три аргумента то задание -1

a = float(sys.argv[1]); # получаем первое действительное число

b = float(sys.argv[2]); # получаем второе действительное число

c = float(sys.argv[3]); # получаем третье действительное число

result = task\_1(a,b,c);

else:

print("Не корректные входные данные");

print("Результат: " + str(result));

**Завдання №3**

1) **Вхідні дані**: рядок, передається в програму як аргумент командного рядка. Може містити пробіли, літери латинського алфавіту в будь-якому регістрі та цифри. Для передачі одним значенням рядок, що містить пробіли, береться в подвійні лапки.

**Результат роботи**: рядок "YES", якщо отриманий рядок є паліндромом, або "NO" - якщо ні. Рядок вважається паліндромом, якщо він однаково читається як зліва направо, так і справа наліво. Відмінністю регістрів та пробілами знехтувати

Task\_1:

def task\_1(string):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + string);

# перевод в нижний регистр так как регистр не учитывается

string = string.lower();

# удаляем пробелы так как пробелы не учитываются

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

# определяем новую строку представленную в обратном порядке

new\_string = string[::-1];

# сравнение

if string == new\_string:

print( "YES");

else:

print ("NO");

Task\_2

def task\_2(string):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + string);

lists = []

# определяем слова учитывая разделитель "пробел"

lists = string.split(" ");

lists.reverse();

print(' '.join(lists));

Task\_3

def task\_3(string):

# отображение полученных аргументов

print("Вхідні дані:" + '' + string);

# определяем список для хранения всех скобок

backets = [];

# определяем счетчики для левой и правой скобки

open\_count = 0;

close\_count = 0;

# проход по строке и выявленые скобки добавляем в список

for symbol in string:

if symbol == '(' :

open\_count += 1;

backets.append(symbol);

elif symbol == ')' :

close\_count += 1;

backets.append(symbol);

# минимальная проверка (оптимизация по большие строки)

if len(backets)%2 != 0 or open\_count != close\_count or backets[0] == ')' or backets[len(backets)-1] == '(':

print("NO")

return;

# счетчик для проверки

open\_count = 0;

for b in backets: # проход по списку

if b == '(' :

open\_count += 1;

elif b == ')' and open\_count > 0 :

open\_count -= 1;

elif b == ')' and open\_count == 0: # если нашли правую скобку а ей нет пары то

print("NO");

return;

print("YES");

Task\_4 :

def task\_4(a,b):

# отображение полученных аргументов

print("Вхідні дані:" + ' ' + str(a) + ' ' + str(b));

counter = 0; # cчетчик совпадений

for i in range(a,b):

argument = str(i);

arg\_len = len(argument);

if arg\_len < 6:

need = '0'\* (6 - arg\_len);

argument = need + argument;

first\_part = str(argument[:3]);

second\_part = str(argument[3:]);

f = 0;

r = 0;

for s in first\_part:

f = f + int(s);

for s in second\_part:

r = r + int(s);

if r == f:

counter =counter + 1;

print("Результат: " + str(counter));

Test :

arg\_count = len(sys.argv); # определение количества аргументов не учитывая имя файла

task = 0; # номер задания для выполнения идет вторым аргументом после имени файла

if arg\_count >=3: # минимальное необходимое количество аргументов

task = int(sys.argv[1]); # определенияем задание

else:

print("Не достаточно аргументов");

exit(1)

arguments = [] # список для аргументов

for index in range(2, arg\_count): # получаем все аргументы кроме имени файла и номера я

arguments.append(sys.argv[index]);

if task == 1 : # если задание номер 1

task\_1(''.join(arguments));

elif task == 2: # если задание номер 2

task\_2(''.join(arguments));

elif task == 3: # если задание номер 3

task\_3(''.join(arguments));

elif task == 4: # если задание номер 4

a = int(sys.argv[2]); # определение необходимых аргументов

b = int(sys.argv[3]); # определение необходимых аргументов

if a >= 0 and a <= b and b <= 999999:

task\_4(a, b);

else:

print("Не корректные аргументы ");

else:

print("Не корректные аргументы ");