

Завдання для тренування

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

Вихідні дані: дійсне число -- результат обчислення формули

$$x = \frac{\sqrt{ab}}{e^a * b} + ae^{\frac{2a}{b}}$$

Приклад

Вхідні дані: 0 1

Приклад виклику: `python test.py 0 1`

Результат: 0.0

Вхідні дані: 0.5 10

Приклад виклику: `python test.py 0.5 10`

Результат: 0.688209837593

Практичне завдання №1.1

Вхідні дані: 3 дійсних числа -- аргументи командного рядка.

Вихідні дані: результат обчислення формули

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right)$$

Аргументи передаються в порядку, зазначеному у формулі, назви змінних можуть використовуватися будь-які.

Приклад

Вхідні дані: 1 1 0.25

Приклад виклику: `python lab2_1.py 1 1 0.25`

Результат: 1.59576912161

Практичне завдання №1.2

Вхідні дані: 3 числа x , y та z . x , y -- невід'ємні цілі числа, z дорівнює 0 або 1. x не дорівнює 0. Передаються як аргументи командного рядка.

Вихідні дані: рядок "Everybody sing a song: <текст пісеньки>.", де <текст пісеньки> формується з y куплетів, розділених пробілами. Всі куплети однакові і складаються з x 'la' через дефіс. Якщо z дорівнює одиниці, в кінці ставиться окличний знак, інакше крапка. За відсутності куплетів пробіл перед крапкою/окличним знаком не ставиться.

Підказка: для цього можна застосувати множення рядків.

Приклад

Вхідні дані: 2 3 1

Приклад виклику: `python lab2_2.py 2 3 1`

Результат: Everybody sing a song: la-la la-la la-la!

Вхідні дані: 1 0 0

Приклад виклику: `python lab2_2.py 1 0 0`

Результат: Everybody sing a song:.

ПРАКТИЧНЕ ЗАВДАННЯ №1.3

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.

Наприклад

Вхідні дані: 10 20 30

Приклад виклику: `python lab3_1.py 10 20 30`

Результат: not triangle

Вхідні дані: 1 1 1

Приклад виклику: `python lab3_1.py 1 1 1`

Результат: triangle

Вхідні дані: 5.5 5.5 -2

Приклад виклику: `python lab3_1.py 5.5 5.5 -2`

Результат: not triangle

ПРАКТИЧНЕ ЗАВДАННЯ №1.4

Програма має розраховувати числа послідовності Фібоначчі. Послідовність Фібоначчі -- це послідовність чисел, в якій кожний елемент дорівнює сумі двох попередніх. При цьому нульовий елемент вважається за 0, а перший 1. Отже, сама послідовність виглядає наступним чином:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, ...

Вхідні дані: ціле невід'ємне число n. Передається в програму як аргумент командного рядка.

Результат роботи: значення n-го числа послідовності Фібоначчі.

Будь ласка, не використовуйте рекурсію.

Наприклад

Вхідні дані: 0

Приклад виклику: `python lab3_2.py 0`

Результат: 0

Вхідні дані: 10

Приклад виклику: `python lab3_2.py 10`

Результат: 55

Практичне завдання №2.1

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.

Наприклад

Вхідні дані: 0

Приклад виклику: `python lab4_1.py 0`

Результат: YES

Вхідні дані: puppy

Приклад виклику: `python lab4_1.py puppy`

Результат: NO

Вхідні дані: "mystringlGni rts ym"

Приклад виклику: `python lab4_1.py "mystringlGni rts ym"`

Результат: YES

Практичне завдання №2.2

Вхідні дані: довільна, відмінна від нуля, кількість значень - аргументів командного рядка. Значеннями-аргументами можуть бути числа або рядки, які складаються з цифр та літер латинського алфавіту без пробілів.

Результат роботи: рядок, що складається з отриманих значень в зворотньому порядку, записаних через пробіл. Пробіл в кінці рядка відсутній.

Наприклад

Вхідні дані: 1

Приклад виклику: `python lab4_3.py 1`

Результат: 1

Вхідні дані: qwe asd zxc 123

Приклад виклику: `python lab4_3.py qwe asd zxc 123`

Результат: 123 zxc asd qwe

Вхідні дані: padawan young my HAVE MUST YOU PATIENCE

Приклад виклику: `python lab4_3.py padawan young my HAVE MUST YOU PATIENCE`

Результат: PATIENCE YOU MUST HAVE my young padawan

Практичне завдання №2.3

Вхідні дані: рядок, що складається з відкриваючих та закриваючих круглих дужок - аргумент командного рядка. Для передачі в якості рядка послідовність береться в подвійні лапки.

Результат роботи: рядок "YES", якщо вхідний рядок містить правильну дужкову послідовність; або рядок "NO", якщо послідовність є неправильною. Дужкова послідовність вважається правильною, якщо всі дужки можна розбити попарно "відкриваюча"- "закриваюча", при чому в кожній парі закриваюча дужка слідує після відкриваючої.

Пояснення для математиків:

"" (порожній рядок) - правильна дужкова послідовність (ПДП)

"(ПДП)" - також ПДП

"ПДППДП" (дві ПДП записані поряд) - також ПДП

Наприклад

Вхідні дані: ")"("

Приклад виклику: python lab4_3.py ")"("

Результат: NO

Вхідні дані: "(()(())"

Приклад виклику: python lab4_3.py "(()(())"

Результат: NO

Вхідні дані: "(()(())())"

Приклад виклику: python lab4_3.py "(()(())())"

Результат: YES

Вхідні дані: "()()()()()()

Приклад виклику: python lab4_3.py "()()()()()()

Результат: NO

Практичне завдання №2.4

(2/2 балів)

Завдання передбачає пошук "щасливих" квитків. "Щасливим" називається такий тролейбусний квиток, в якому сума перших трьох цифр дорівнює сумі останніх трьох. Наприклад 030111 ($0+3+0 = 1+1+1$), 902326 ($9+0+2 = 3+2+6$), 001100 ($0+0+1 = 1+0+0$).

Вхідні дані: два цілих невід'ємних числа ($0 \leq a1 \leq a2 \leq 999999$) - аргументи командного рядка.

Результат роботи: кількість "щасливих квитків", чії номери лежать на проміжку від $a1$ до $a2$ включно. Якщо число на проміжку має менше 6 розрядів, вважати, що на його початку дописуються нулі у необхідній кількості, як це і відбувається при нумерації квитків. Виводити номери квитків не треба.

Наприклад

Вхідні дані: 0 1000

Приклад виклику: python lab4_4.py 0 1000

Результат: 1

Пояснення: номер 000000

Вхідні дані: 1001 1122

Приклад виклику: python lab4_4.py 1001 1122

Результат: 3

Пояснення: номери 001001, 001010, 001100

Вхідні дані: 222222 222333

Приклад виклику: python lab4_4.py 222222 222333

Результат: 7

Пояснення: номери 222222, 222231, 222240, 222303, 222312, 222321, 222330

Завдання №3.1

Дешифрувати повідомлення, зашифроване шифром Бекона

Для кодування повідомлень Френсіс Бекон запропонував кожен літер тексту замінювати на групу з п'яти символів «А» або «В» (назвемо їх "аб-групами"). Для співставлення літер і кодуємих аб-груп в даному завданні використовується ключ-ланцюжок **aaaaabbbbabbbaabbababbaaabaab**, в якому порядковий номер літери відповідає порядковому номеру початку аб-групи.

Наприклад: літера "a" - перша літера алфавіту; для визначення її коду беремо 5 символів з ключа, починаючи з першого: aaaaa. Літера "c" - третя в алфавіті, отже для визначення її коду беремо 5 символів з ключа, починаючи з третього: aaabb.

Таким чином, оригінальне повідомлення перетворюється на послідовність аб-груп і може далі бути накладене на будь-який текст відповідної довжини: А позначається нижнім регістром, В - верхнім.

Наприклад, початкове повідомлення - Prometheus.

1. Кодуємо його через аб-послідовності:

p = abbab
r = babab
o = aabba
m = bbaab
e = abbbb
t = babba
h = bbbab
e = abbbb
u = abbaa
s = ababb

Результат: abbab babab aabba bbaab abbbb babba bbbab abbbb abbaa ababb

2. Підбираємо текст приблизно такої ж довжини, в якому сховаємо наше повідомлення:
Welcome to the Hotel California Such a lovely place Such a lovely place

3. Для зручності розбиваємо його на групи по 5 символів і відкидаємо зайву частину (щоб при декодуванні не отримувалися зайві п'ятірки). Співставимо аб-рядок і текст-сховище для порівняння:

abba b babab aabba bbaab abbbb babba bbbab abbbb abbaa ababb

Welco metot heHot elCal iforn iaSuc halov elypl aceSu chalo vely

4. Змінюємо регістр символів, кодуючи А та В:

abba b babab aabba bbaab abbbb babba bbbab abbbb abbaa ababb

wELcO MeToT heHOt ELcaL iFORN IaSUc HALoV eLYPL aCEsu cHaLO vely

5. Повертаємо початкові пробіли на місце:

wELcOMe To The HOtEL caLiFORNiA SuCh A LoVeLY PLaCE suCh a LOvely

Для дешифрування повідомлення необхідно виконати зворотні дії.

Вхідні дані: рядок, передається в програму як аргумент командного рядка. Може містити пробіли та літери латинського алфавіту в будь-якому регістрі. Для передачі в якості одного аргументу рядок береться в подвійні лапки.

Результат роботи: рядок - дешифроване повідомлення.

Наприклад

1. Вхідні дані: I canT DAnCE i CANt TAlK Hey
 2. видаляємо пробіли, розбиваємо на групи по 5 символів: IcanT DAnCE iCANt TAlKH ey
 3. ey відкдається
 4. символи нижнього регістру перетворюються в а, верхнього - в b: baaab bbabb abbba bbabb
 5. декодуємо, використовуючи ключ:
baaab = w
bbabb = i
abbba = k
bbabb = i
 6. Результат: wiki
-
1. Вхідні дані: Hot sUn BEATIng dOWN bURNING mY FEet JuSt WalKIng arOUnd HOt suN mAKiNG me SWeat
 2. видаляємо пробіли, розбиваємо на групи по 5 символів: HotsU nBEAT IngdO WNBUR NIngm YFEet JuStW alKIn garOU nDHOt suNmA kiNGm eSWea t
 3. t відкдається
 4. символи нижнього регістру перетворюються в а, верхнього - в b: baaab abbbb baaab bbabb bbbaa bbbaa babab aabba aaabb abbba aabab aabba abbaa
 5. декодуємо, використовуючи ключ:
baaab = w
abbbb = e
baaab = w
bbabb = i
bbbaa = l
bbbaa = l
babab = r
aabba = o
aaabb = c
abbba = k
aabab = y
aabba = o
abbaa = u
 6. Результат: wewillrockyou

Завдання №3.2

Задача «Шоколадка»

Шоколадка имеет вид прямоугольника, разделенного на $n \times m$ долек. Шоколадку можно один раз разломить по прямой на две части. Определите, можно ли таким образом отломить от шоколадки часть, состоящую ровно из k долек. Программа получает на вход три числа: n , m , k и должна вывести YES или NO.

Входные данные	Выходные данные
4 2 6	YES
2 10 7	NO
5 7 1	NO
7 4 21	YES
5 12 100	NO

Завдання №3.3

Для настольной игры используются карточки с номерами от 1 до N. Одна карточка потерялась. Найдите ее, зная номера оставшихся карточек.

Дано число N, далее N – 1 номер оставшихся карточек (различные числа от 1 до N).

Программа должна вывести номер потерянной карточки.

Входные данные	Выходные данные
5 1 2 3 4	5
5 3 5 2 1	4
4 3 2 4	1

Завдання №3.4

По данному натуральному $n \leq 9$ выведите лесенку из n ступенек, i -я ступенька состоит из чисел от 1 до i без пробелов.

Входные данные	Выходные данные
3	1 12 123

Завдання №3.5

Вклад в банке составляет x грн. Ежегодно он увеличивается на p процентов, после чего дробная часть копеек отбрасывается. Определите, через сколько лет вклад составит не менее y грн.

Выражение «дробная часть копеек отбрасывается» означает, что если у вас оказалось 123.4567 грн, т. е. 123 грн. и 45.67 копеек, то после округления у вас получится 123 рубля и 45 копеек, т.е. 123.45 грн.

Программа получает на вход три натуральных числа: x , p , y и должна вывести одно целое число.

Входные данные	Выходные данные
100 10 200	8
100 100 200	1