Задача № 1/5

Задание № 1.

Напишите программу, которая считывает два целых числа *a* и *b* и выводит наибольшее значение из них. Числа — целые от 1 до 1000.

При решении задачи можно пользоваться только целочисленными арифметическими операциями +, -, \*, //, %, =. Нельзя пользоваться нелинейными конструкциями: ветвлениями, циклами, функциями вычисления модуля, извлечения квадратного корня.

a = int(input('первое число? '))

b = int(input('второе число? '))

print('Наибольшее число ',(a\*(a//b) + b\*(b//a))/(b//a+a//b))

Задание № 2.

Подсчитать значение числа a в степени b с логарифмической сложностью.

n = int(input('число? '))

p = int(input('степень? '))

b = 1

while (p > 0):

  b \*= n

  p -= 1

print('результат:', b)

O(N)

Задание № 3.

Написать алгоритм (функцию) определения четности целого числа, который будет аналогичен нижеприведенному по функциональности, но отличен по своей сути.

Объяснить плюсы и минусы обеих реализаций.

C/C++ example:

bool isEven(int value){return value%2==0;}

Решения по модулю 2 с %2 хороши, но для этого требуется деление и вычитание. Поскольку компьютеры используют двоичную арифметику, гораздо более эффективным решением является:

value = int(input('Введите число? '))

is\_odd = (value & 1) == 1

print(is\_odd)

Задание № 4.

Написать минимум по 2 реализации циклических буферов FIFO и LIFO.

Объяснить плюсы и минусы каждой реализации.

Fifo

import queue

q1 = queue.Queue()

#Addingitems to the queue

q1.put(11)

q1.put(5)

q1.put(4)

q1.put(21)

q1.put(3)

q1.put(10)

#using bubble sort on the queue

n =  q1.qsize()

for i in range(n):

    x = q1.get() # the element is removed

    for j in range(n-1):

        y = q1.get() # the element is removed

        if x > y :

            q1.put(y)   #the smaller one is put at the start of the queue

        else:

            q1.put(x)  # the smaller one is put at the start of the queue

            x = y     # the greater one is replaced with x and compared again with nextelement

    q1.put(x)

while (q1.empty() == False):

    print(q1.queue[0], end = " ")

    q1.get()

Lifo

import queue

q1 = queue.LifoQueue()

for i in range(20):

    q1.put(i)

import queue

q1 = queue.LifoQueue()

for i in range(20):

    q1.put(i)

while not q1.empty():

  print("The value is ", q1.get())

Lifo – Fifo

отличаются способом **удаления элементов**:

В случае с очередью вы удаляете объект, который был недавно добавлен (по принципу первый вошел – первым вышел first-in, first-out или **FIFO**), в случае сто стеком вы удаляете последний добавленный элемент (последним зашел – первым вышел, last-in, first-out or **LIFO**).

Задание № 5.

Написать реализацию бинарного поиска.

from random import randint

a = []

for i in range(10):

    a.append(randint(1, 50))

a.sort()

print(a)

value = int(input())

mid = len(a) // 2

low = 0

high = len(a) - 1

while a[mid] != value and low <= high:

    if value > a[mid]:

        low = mid + 1

    else:

        high = mid - 1

    mid = (low + high) // 2

if low > high:

    print("No value")

else:

    print("ID =", mid)