

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

|  |  |
| --- | --- |
| Факультет | Информатика и вычислительная техника |
|  | (наименование факультета) |
| Кафедра | Кибербезопасность информационных систем |
|  | (наименование кафедры) |

**ОТЧЕТ по практической работе**

**по дисциплине “Методы программирования”**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Автор |  | | |  | Короп А.Д. | | | |
|  | (подпись, дата) | | |  |  | | | |
| Обозначение | 10.05.01.550000.000 О | | Группа | | | | ВКБ33 |
| Направление подготовки | | 10.05.01 Компьютерная безопасность | | | | | |
| Профиль | Компьютерная безопасность | | | | | | |
| Преподаватель |  | | |  | | Савельев В.А. | |
|  | (подпись, дата) | | |  | |  | |

г. Ростов-на-Дону

2021 год

**Лабораторная работа №1**

**Простые структуры данных: массив, список, двунаправленный список, стек, очередь, дек, динамический массив**

Задача 1

Гоблины Мглистых гор очень любых ходить к своим шаманам. Так как гоблинов много, к шаманам часто образуются очень длинные очереди. А поскольку много гоблинов в одном месте быстро образуют шумную толку, которая мешает шаманам проводить сложные медицинские манипуляции, последние решили установить некоторые правила касательно порядка в очереди.

Обычные гоблины при посещении шаманов должны вставать в конец очереди. Привилегированные же гоблины, знающие особый пароль, встают ровно в ее середину, причем при нечетной длине очереди они встают сразу за центром.

Так как гоблины также широко известны своим непочтительным отношением ко всяческим правилам и законам, шаманы попросили вас написать программу, которая бы отслеживала порядок гоблинов в очереди.

Входные данные

В первой строке входных данный записано число NN (1≤N≤1051≤N≤105) - количество запросов к программе. Следующие NN строк содержат описание запросов в формате:

* "+ i" - гоблин с номером ii (1≤i≤N1≤i≤N) встает в конец очереди.
* "\* i" - привилегированный гоблин с номером ii встает в середину очереди.
* "-" - первый гоблин из очереди уходит к шаманам. Гарантируется, что на момент такого запроса очередь не пуста.

Выходные данные

Для каждого запроса типа "-" программа должна вывести номер гоблина, который должен зайти к шаманам.

**Примеры**

|  |  |
| --- | --- |
| входные данные  7  + 1  + 2  -  + 3  + 4  -  - | выходные данные  1  2  3 |

Выполнение работы

from collections import deque

n = int(input())

q = deque()

qt = deque()

res = []

k1 = 0

k2 = 0

for i in range(n):

t1 = input().split()

if "-" in t1:

res.append(q.popleft())

k1 -= 1

elif '+' in t1:

qt.append(t1[-1])

k2 += 1

else :

qt.appendleft(t1[-1])

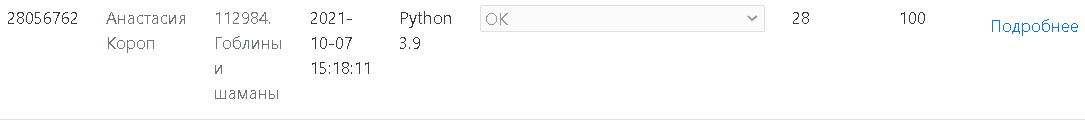
k2 += 1

if k1 < k2:

q.append(qt.popleft())

k2 -= 1

k1 += 1

print(\*res, sep='\n')

Задача 2

Даны числа n и m. Создайте массив A[n][m] и заполните его, как показано на примере.

Входные данные

Программа получает на вход два числа n и m.

Выходные данные

Программа должна вывести полученный массив.

**Примеры**

|  |  |
| --- | --- |
| входные данные  4 10 | выходные данные  0 1 3 6 10 14 18 22 26 30  2 4 7 11 15 19 23 27 31 34  5 8 12 16 20 24 28 32 35 37  9 13 17 21 25 29 33 36 38 39 |

Выполнение работы

#include <iostream>

using namespace std;

int main()

{

const int N = 4;

const int M = 10;

int arr[N][M] =

{

{ 0, 1, 3, 6, 10, 14, 18, 22, 26, 30 },

{ 2, 4, 7, 11, 15, 19, 23, 27, 31, 34 },

{ 5, 8, 12, 16, 20, 24, 28, 32, 35, 37 },

{ 9, 13, 17, 21, 25, 29, 33, 36, 38, 39 }

};

for (size\_t i = 0; i < N; ++i)

{

for (size\_t j = 0; j < M; ++j)

{

cout << arr[i][j] << '\t';

}

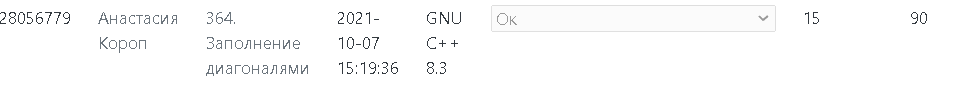
cout << endl;

}

cout << endl;

return 0;

}



Задача 3

Рассмотрим последовательность, состоящую из круглых, квадратных и фигурных скобок. Программа должна определить, является ли данная скобочная последовательность правильной.

Пустая последовательность явлеятся правильной. Если A – правильная, то последовательности (A), [A], {A} – правильные. Если A и B – правильные последовательности, то последовательность AB – правильная.

Входные данные

В единственной строке записана скобочная последовательность, содержащая не более 100000 скобок.

Выходные данные

Если данная последовательность правильная, то программа должна вывести строку yes, иначе строку no.

**Примеры**

|  |  |
| --- | --- |
| входные данные  ()[] | выходные данные  yes |

Выполнение работы

def checksq(s):

stack = ''

op,cl = '([{ )]}'.split()

for c in s:

if c in op:

stack += c

elif c in cl:

if stack == '':

return False

elif op[cl.index(c)] == stack[-1]:

stack = stack[:-1]

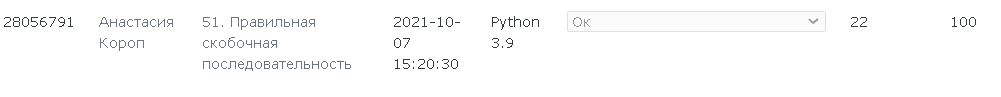
return stack == ''

a = '{}()(){{{{}}[][][]}}'

b = '{}()(){{{{}}[][][]}}())'

print(f'В сроке "{a}" скобки расставлены {"" if checksq(a) else "не "}правильно.')

print(f'В сроке "{b}" скобки расставлены {"" if checksq(b) else "не "}правильно.')



Задача 4

К тупику со стороны пути 1 подъехал поезд. Разрешается отцепить от поезда один или сразу несколько первых вагонов и завезти их в тупик (при желании, можно даже завезти в тупик сразу весь поезд). После этого часть из этих вагонов вывезти в сторону пути 2. После этого можно завезти в тупик еще несколько вагонов и снова часть оказавшихся вагонов вывезти в сторону пути 2. И так далее (так, что каждый вагон может лишь один раз заехать с пути 1 в тупик, а затем один раз выехать из тупика на путь 2). Заезжать в тупик с пути 2 или выезжать из тупика на путь 1 запрещается. Нельзя с пути 1 попасть на путь 2, не заезжая в тупик.

Известно, в каком порядке изначально идут вагоны поезда. Требуется с помощью указанных операций сделать так, чтобы вагоны поезда шли по порядку (сначала первый, потом второй и т.д., считая от головы поезда, едущего по пути 2 в сторону от тупика). Напишите программу, определяющую, можно ли это сделать.

Входные данные

Вводится число N — количество вагонов в поезде (1≤N≤100). Дальше идут номера вагонов в порядке от головы поезда, едущего по пути 1 в сторону тупика. Вагоны пронумерованы натуральными числами от 1 до N, каждое из которых встречается ровно один раз.

Выходные данные

Если сделать так, чтобы вагоны шли в порядке от 1 до N, считая от головы поезда, когда поезд поедет по пути 2 из тупика, можно, выведите сообщение YES, если это сделать нельзя, выведите NO.

**Примеры**

|  |  |  |
| --- | --- | --- |
| Входные данные | Входные данные | Комментарии |
| 3  3 2 1 | YES | Надо весь поезд завезти в тупик, а затем целиком вывезти его на 2-й путь. |
| 4  4 1 3 2 | YES | Сначала надо в тупик завезти два вагона, один из которых оставит в тупике, а второй — вывезти на 2-й путь, после чего завезти в тупик еще два вагона и вывезти 3 вагона, стоящие в тупике, на 2-й путь |
| 3  2 3 1 | NO |  |

l = int(input())

s = list(map(int, input().split()))

tup = [0] # тупик

tr2 = [0] # путь 2

for i in range(l):

while tup[-1] == tr2[-1] + 1:

tr2.append(tup[-1])

tup.pop()

if s[i] == tr2[-1] + 1:

tr2.append(s[i])

else:

tup.append(s[i])

while tup[-1] == tr2[-1] + 1:

tr2.append(tup[-1])

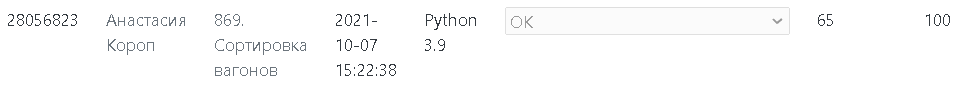
tup.pop()

if tr2[-1] == l:

print('YES')

else:

print('NO')



Задача 5

Шпион Коля зашифровал и послал в центр радиограмму. Он использовал такой способ шифровки: сначала выписал все символы своего сообщения (включая знаки препинания и т.п.), стоявшие на четных местах, в том же порядке, а затем ­– все символы, стоящие на нечетных местах. Напишите программу, которая расшифровывает сообщение.

Входные данные

Вводится одна непустая строка длиной не более 250 символов – зашифрованное сообщение. Строка может состоять из любых символов, кроме пробельных.

Выходные данные

Выведите одну строку – расшифрованное сообщение.

**Примеры**

|  |  |
| --- | --- |
| входные данные  rvt!Pie! | выходные данные  Privet!! |

s = "rvt!Pie! "

lst = ['\_' for \_ in range(len(s))]

for i,x in enumerate(s[:len(s)//2]):

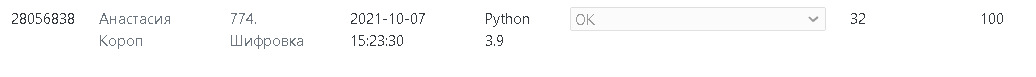
lst[i\*2+1] = x

for i,x in enumerate(s[len(s)//2:]):

lst[i\*2] = x

print(s)

print(''.join(lst))



Задача 6

Формат входных данных

В каждой строке сначала записан номер класса (число, равное 9, 10 или 11), затем (через пробел) – фамилия ученика. Общее число строк в файле не превосходит 100000. Длина каждой фамилии не превосходит 50 символов.

Формат выходных данных

Необходимо вывести список школьников по классам: сначала всех учеников 9 класса, затем – 10, затем – 11. Внутри одного класса порядок вывода фамилий должен быть таким же, как на входе.

Пример

|  |  |
| --- | --- |
| Входные данные | Выходные данные |
| 9 Иванов 10 Петров 11 Сидоров 9 Григорьев 9 Сергеев 10 Яковлев | 9 Иванов 9 Григорьев 9 Сергеев 10 Петров 10 Яковлев 11 Сидоров |

with open("C:\\Users\\admin\\Desktop\\name.txt", "r", encoding="utf-8") as a:

b={}

for i in a.read().split("\n"):

c=i.split(" ")

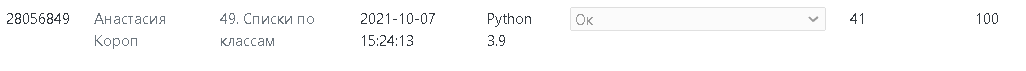
if int(c[0]) not in b:

b[int(c[0])]=[]

b[int(c[0])].append(c[1])

for i in sorted(b):

print(b[i])



Задача 7

В игре в пьяницу карточная колода раздается поровну двум игрокам. Далее они вскрывают по одной верхней карте, и тот, чья карта старше, забирает себе обе вскрытые карты, которые кладутся под низ его колоды. Тот, кто остается без карт – проигрывает.

Для простоты будем считать, что все карты различны по номиналу, а также, что самая младшая карта побеждает самую старшую карту ("шестерка берет туза").

Игрок, который забирает себе карты, сначала кладет под низ своей колоды карту первого игрока, затем карту второго игрока (то есть карта второго игрока оказывается внизу колоды).

Напишите программу, которая моделирует игру в пьяницу и определяет, кто выигрывает. В игре участвует 10 карт, имеющих значения от 0 до 9, большая карта побеждает меньшую, карта со значением 0 побеждает карту 9.

Входные данные

Программа получает на вход две строки: первая строка содержит 5 чисел, разделенных пробелами — номера карт первого игрока, вторая – аналогично 5 карт второго игрока. Карты перечислены сверху вниз, то есть каждая строка начинается с той карты, которая будет открыта первой.

Выходные данные

Программа должна определить, кто выигрывает при данной раздаче, и вывести слово first или second, после чего вывести количество ходов, сделанных до выигрыша. Если на протяжении 106 ходов игра не заканчивается, программа должна вывести слово botva.

**Примеры**

|  |  |
| --- | --- |
| входные данные  1 3 5 7 9  2 4 6 8 0 | выходные данные  second 5 |

first = input().split()

second = input().split()

for n in range(10\*\*6):

try:

a = first.pop(0)

b = second.pop(0)

except IndexError:

print("first" if first else "second", n)

break

winner = first

if (a == '9' and b == '0') or a < b:

winner = second

winner.extend([a, b])

else:

print("botva")

