

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

**(ДГТУ)**

Факультет Информатика и вычислительная техника

Кафедра Кибербезопасность информационных систем

**Тема №1. Задача №112984. Гоблины и шаманы**

**Python version 3.9**

Выполнил обучающийся гр. ВКБ32

Якименко Алексей Андреевич

Проверил

Доцент, Савельев Василий Александрович

Ростов-на-Дону

2021

**Условие задачи**

Гоблины Мглистых гор очень любят 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.

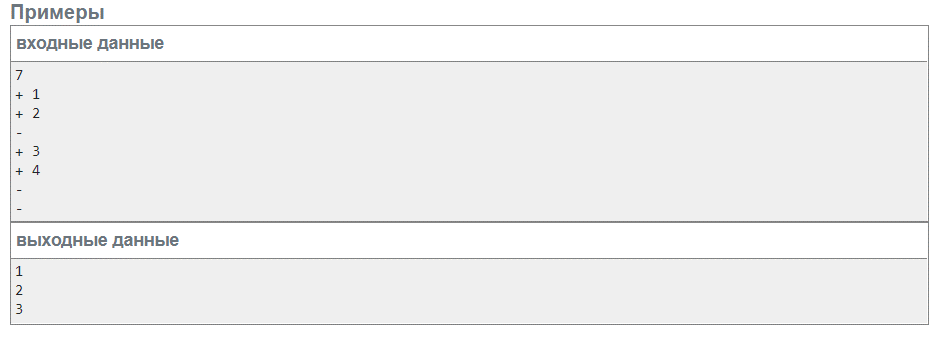
### **Входные данные**

В первой строке входных данный записано число N (1≤N≤10^5) - количество запросов к программе. Следующие N строк содержат описание запросов в формате:

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

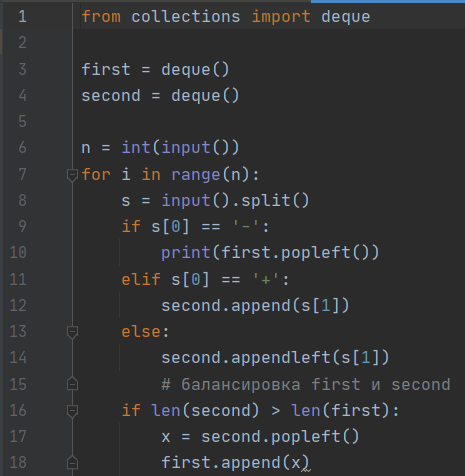
### **Выходные данные**

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



Код программы (Python v3.9):

Скриншот



Текст

from collections import deque  
  
first = deque()  
second = deque()  
  
n = int(input())  
for i in range(n):  
 s = input().split()  
 if s[0] == '-':  
 print(first.popleft())  
 elif s[0] == '+':  
 second.append(s[1])  
 else:  
 second.appendleft(s[1])  
 # балансировка first и second  
 if len(second) > len(first):  
 x = second.popleft()  
 first.append(x)

Ввод (зеленый) и вывод (белый) программы:

