|  |  |  |
| --- | --- | --- |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное автономное  образовательное учреждение высшего образования  «Пермский государственный национальный  исследовательский университет» | | |
|  | Институт компьютерных наук и технологий | |
| **ОТЧЁТ**  по индивидуальной работе №2  по дисциплине «Языки программирования»  Вариант 1 | | |
|  | | Работу выполнил  студент группы ПМИ-10 - 2024 1 курса  Лысков Никита Анатольевич  «16» 06.2024 г. |
| Работу проверил  \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Фамилия И.О.  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2024 г. |
| Пермь 2024 | | |

СОДЕРЖАНИЕ

[Постановка задачи 3](#_Toc169519103)

[Алгоритм решения 4](#_Toc169519106)

[Тестирование 4](#_Toc169519107)

[Код программы 5](#_Toc169519108)

# Постановка задачи

Баржа. На барже располагается K грузовых отсеков. В каждый отсек можно поместить некоторое количество бочек с одним из 10 000 видов топлива. Причём извлечь бочку из отсека можно лишь в случае, если все бочки, помещённые в этот отсек после неё, уже были извлечены. Таким образом, в каждый момент времени в каждом непустом отсеке имеется ровно одна бочка, которую можно извлечь, не трогая остальных. Будем называть такие бочки крайними.

Изначально баржа пуста. Затем она последовательно проплывает через N доков, причём в каждом доке на баржу либо погружается бочка с некоторым видом топлива в некоторый отсек, либо выгружается крайняя бочка из некоторого отсека. Однако, если указанный отсек пуст, либо если выгруженная бочка содержит не тот вид топлива, который ожидалось, следует зафиксировать ошибку. Если на баржу оказывается погружено более P бочек или если после прохождения всех доков она не стала пуста, следует также зафиксировать ошибку. От вас требуется либо указать максимальное количество бочек, которые одновременно пребывали на барже либо зафиксировать ошибку.

Входные данные: в первой строке три целых числа N, K и P (1 ≤ N, K, P ≤ 100 000). Далее следует N строк с описанием действия, выполняемого в очередном доке. Если в нём происходит погрузка, то строка имеет вид «+ A B», где A — номер отсека, в который помещается бочка, а B — номер вида топлива в ней. Если же док занимается разгрузкой, то строка имеет вид «- A B», где A — номер отсека, из которого извлекается бочка, а B — номер ожидаемого вида топлива. Выходные данные: вывести либо одно число, равное искомому максимуму в случае безошибочного прохождения баржой маршрута, либо вывести слово «Error» в противном случае.

# Пример: (входные данные) 6 1 2 + 1 1 + 1 2 - 1 2 - 1 1 + 1 3 - 1 3

# (выходные данные) 2

# Алгоритм решения

1. Инициализируем переменные и структуры данных:
   * Стек для каждого из K отсеков.
   * Переменные для отслеживания текущего количества бочек на барже и максимального количества бочек.
2. Чтение входных данных:
   * В первой строке считываем три числа: N, K, P.
   * В следующих N строках считываем описание операций.
3. Обработка каждой операции:
   * Для операции погрузки (+ A B):
     + Проверяем, не превышает ли новое количество бочек P.
     + Добавляем бочку в указанный отсек и обновляем текущие и максимальные значения.
   * Для операции выгрузки (- A B):
     + Проверяем, есть ли в отсеке бочки и соответствует ли вид топлива ожидаемому.
     + Извлекаем бочку и обновляем текущее количество бочек.
4. Проверка после всех операций:
   * Если баржа не пуста, фиксируем ошибку.
   * Выводим максимальное количество бочек на барже.

# Тестирование

Входные данные:

6 1 2

+ 1 1

+ 1 2

- 1 2

- 1 1

+ 1 3

- 1 3

Выходные данные:

2

Входные данные:

5 2 3

+ 1 1

+ 2 2

- 1 1

- 2 2

+ 1 1

Выходные данные:

Error

# Код программы

def process\_barge\_operations():

import sys

try:

# Чтение данных

print("Введите количество операций (N), количество отсеков (K) и максимальное количество бочек (P) через пробел:")

first\_line = input().strip()

N, K, P = map(int, first\_line.split())

operations = []

print(f"Введите {N} операций (например, + 1 1 или - 1 1):")

for \_ in range(N):

operations.append(input().strip())

# Проверка корректности ввода

if len(operations) != N:

print("Error")

return

# Инициализируем структуры данных

stacks = [[] for \_ in range(K)]

current\_bottles = 0

max\_bottles = 0

for op in operations:

parts = op.split()

if len(parts) != 3:

print("Error")

return

operation, A, B = parts

A = int(A) - 1

B = int(B)

if operation == '+':

if current\_bottles + 1 > P:

print("Error")

return

if 0 <= A < K:

stacks[A].append(B)

current\_bottles += 1

if current\_bottles > max\_bottles:

max\_bottles = current\_bottles

else:

print("Error")

return

elif operation == '-':

if 0 <= A < K and stacks[A]:

expected\_bottle = stacks[A].pop()

if expected\_bottle != B:

print("Error")

return

current\_bottles -= 1

else:

print("Error")

return

else:

print("Error")

return

if current\_bottles != 0:

print("Error")

else:

print(max\_bottles)

except Exception as e:

print("Error")

if \_\_name\_\_ == "\_\_main\_\_":

process\_barge\_operations()