|  |  |  |
| --- | --- | --- |
| 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 7 | | |
|  | | Работу выполнил  студент группы ИТ-9,10-2025 1 курса  Лежнёв Анатолий Константинович  «08» июня 2025 г. |
| Работу проверил(а)  Анисимова Светлана Игоревна  «\_\_\_» \_\_\_\_\_\_\_\_\_ 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 3](#_Toc200324159)

[Алгоритм решения 4](#_Toc200324160)

[Представление очереди карт: 4](#_Toc200324161)

[Чтение входных данных: 4](#_Toc200324162)

[Основной цикл игры: 4](#_Toc200324163)

[Обработка результатов раунда: 5](#_Toc200324164)

[Пользовательский интерфейс: 5](#_Toc200324165)

[Завершение игры: 5](#_Toc200324166)

[Вывод результата: 6](#_Toc200324167)

[Запуск программы: 6](#_Toc200324168)

[Тестирование 7](#_Toc200324169)

[Вывод меню 7](#_Toc200324170)

[Тестирование №1 7](#_Toc200324171)

[Тестирование №2 7](#_Toc200324172)

[Тестирование №3 7](#_Toc200324173)

[Тестирование №4: 8](#_Toc200324174)

[Тестирование №5: 8](#_Toc200324175)

[Тестирование №6: 9](#_Toc200324176)

[Код программы 10](#_Toc200324177)

# Постановка задачи

Игра в «пьяницу». 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.

# Алгоритм решения

Для решения задачи был разработан следующая структура программы:

## Представление очереди карт:

class Queue:

def \_\_init\_\_(self, items: List[int]) -> None:

self.items = items

def enqueue(self, item: int) -> None:

self.items.append(item)

def dequeue(self) -> int:

if self.is\_empty():

raise IndexError("Очередь пуста")

return self.items.pop(0)

def is\_empty(self) -> bool:

return len(self.items) == 0

 Для хранения карт каждого игрока используется класс Queue, реализующий очередь с методами enqueue, dequeue, is\_empty.

## Чтение входных данных:

Ввод осуществляется через стандартный ввод: две строки с пятью числами для каждого игрока.

## Основной цикл игры:

def compare\_cards(card1: int, card2: int) -> int:

if card1 == 0 and card2 == 9:

return 1

elif card1 == 9 and card2 == 0:

return -1

elif card1 < card2:

return 1

elif card1 > card2:

return -1

else:

return 0

* Игра продолжается, пока обе очереди не опустеют или не достигнут лимит ходов.

На каждом ходе:

* Извлекаются верхние карты каждого игрока (dequeue).

Карты сравниваются по правилам:

* Карта 0 побеждает карту 9.
* Карта 9 проигрывает карте 0.
* В остальных случаях карта с меньшим числом побеждает.

## Обработка результатов раунда:

 Победитель добавляет обе карты в конец своей очереди.

if result == 1:

queue1.enqueue(card1)

queue1.enqueue(card2)

else:

queue2.enqueue(card1)

queue2.enqueue(card2)

## Пользовательский интерфейс:

* Реализация основного цикла программы с интерфейсом, который видит и взаимодействует пользователь

def main():

print("Игра в «пьяницу»")

print("Введите по пять чисел (от 0 до 9) для каждого игрока.")

while True:

try:

input\_line\_1 = input("Карты первого игрока (через пробел): ").strip()

input\_line\_2 = input("Карты второго игрока (через пробел): ").strip()

## Завершение игры:

* Если первая очередь пуста — выигрывает второй (second) с количеством сделанных ходов.
* Если вторая очередь пуста — выигрывает первый (first) с количеством ходов.
* Если лимит ходов достигнут без победителя — объявляется ничья ("botva").

## Вывод результата:

 В зависимости от ситуации выводится строка вида:

first <число\_ходов>

second <число\_ходов>

botva

## Запуск программы:

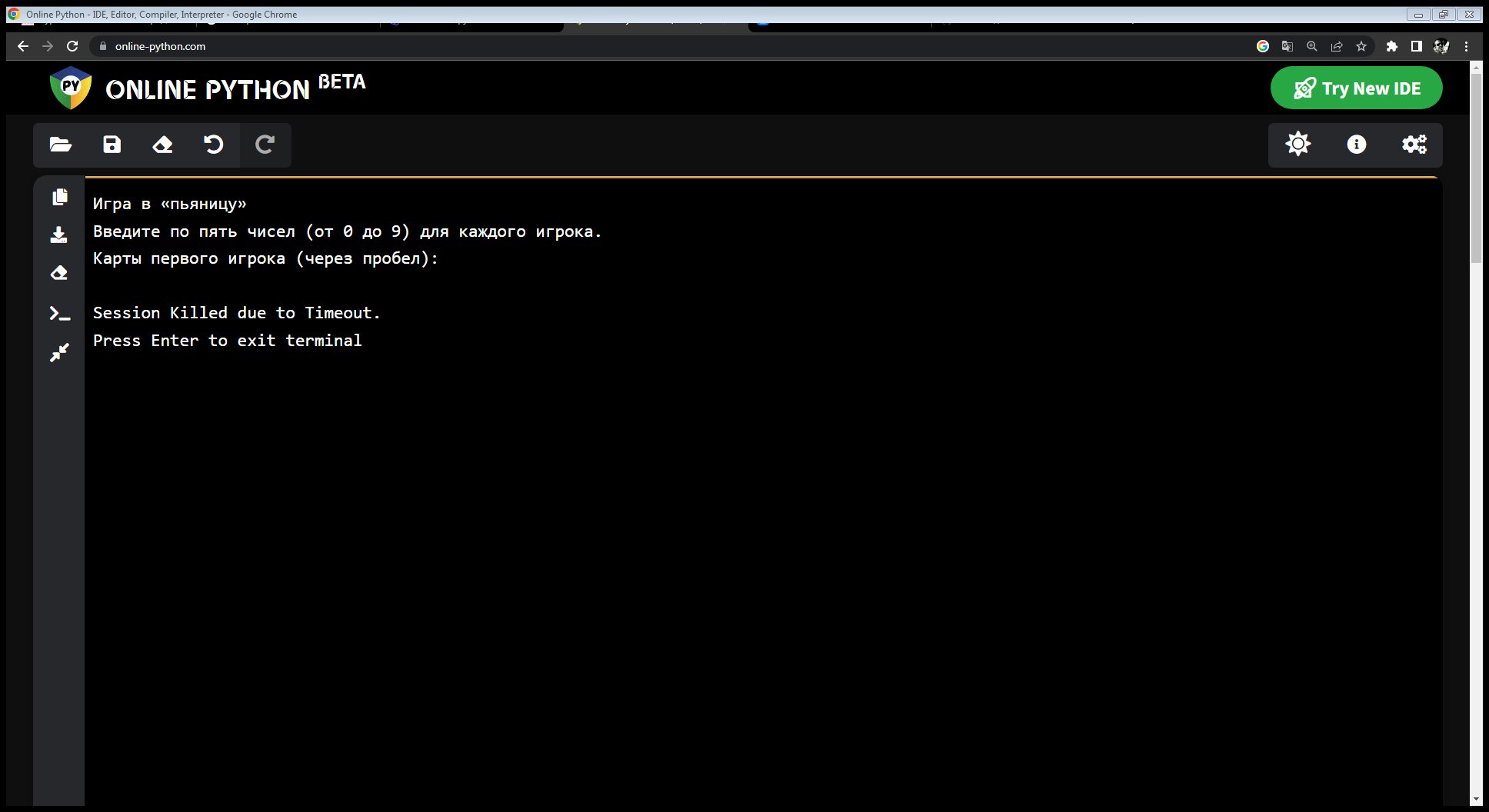
 Стандартная блокировка для запуска основной функции программы

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

main()

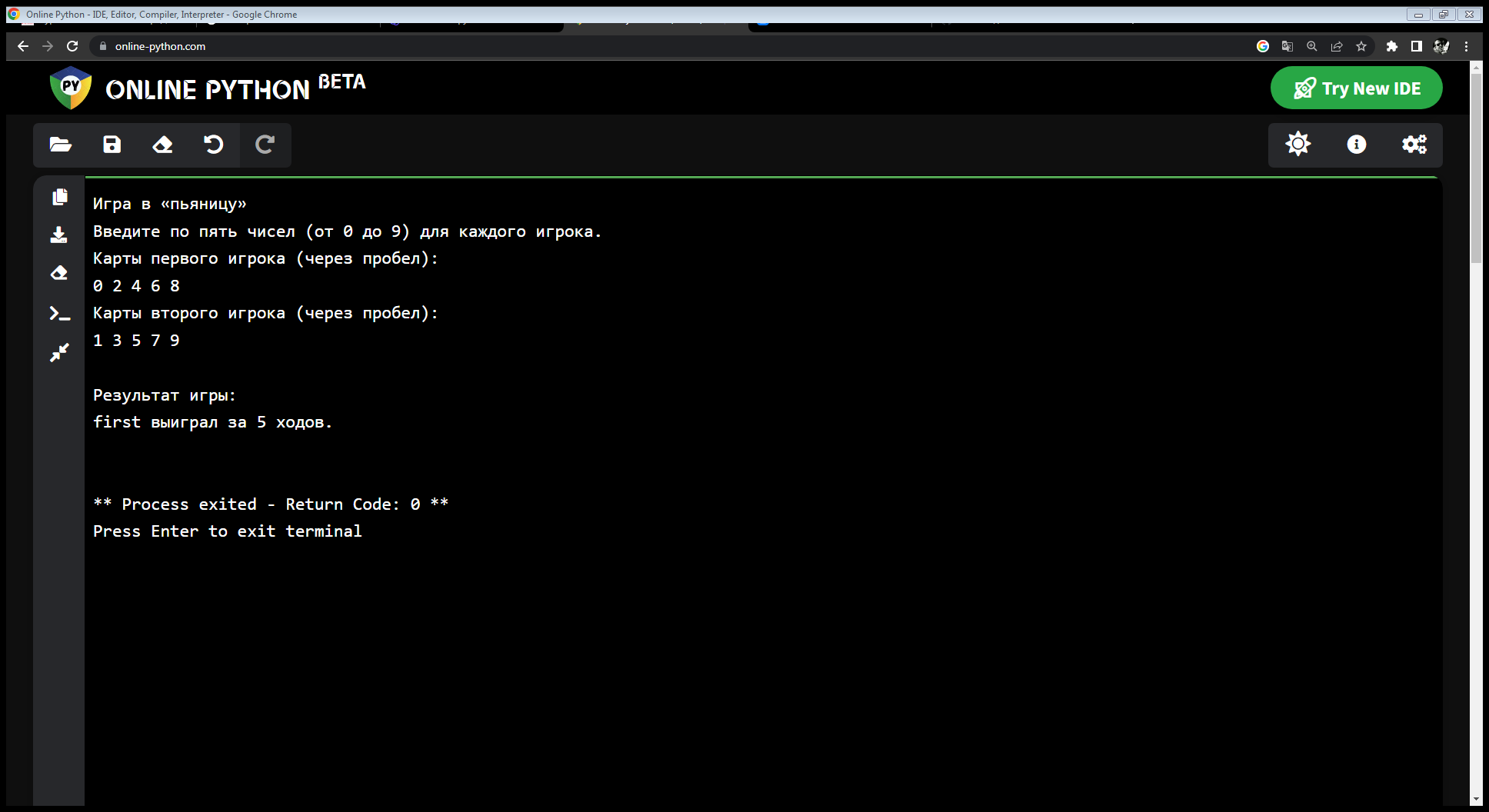
# Тестирование

Тестирование программы включало следующие сценарии:

Вывод меню:

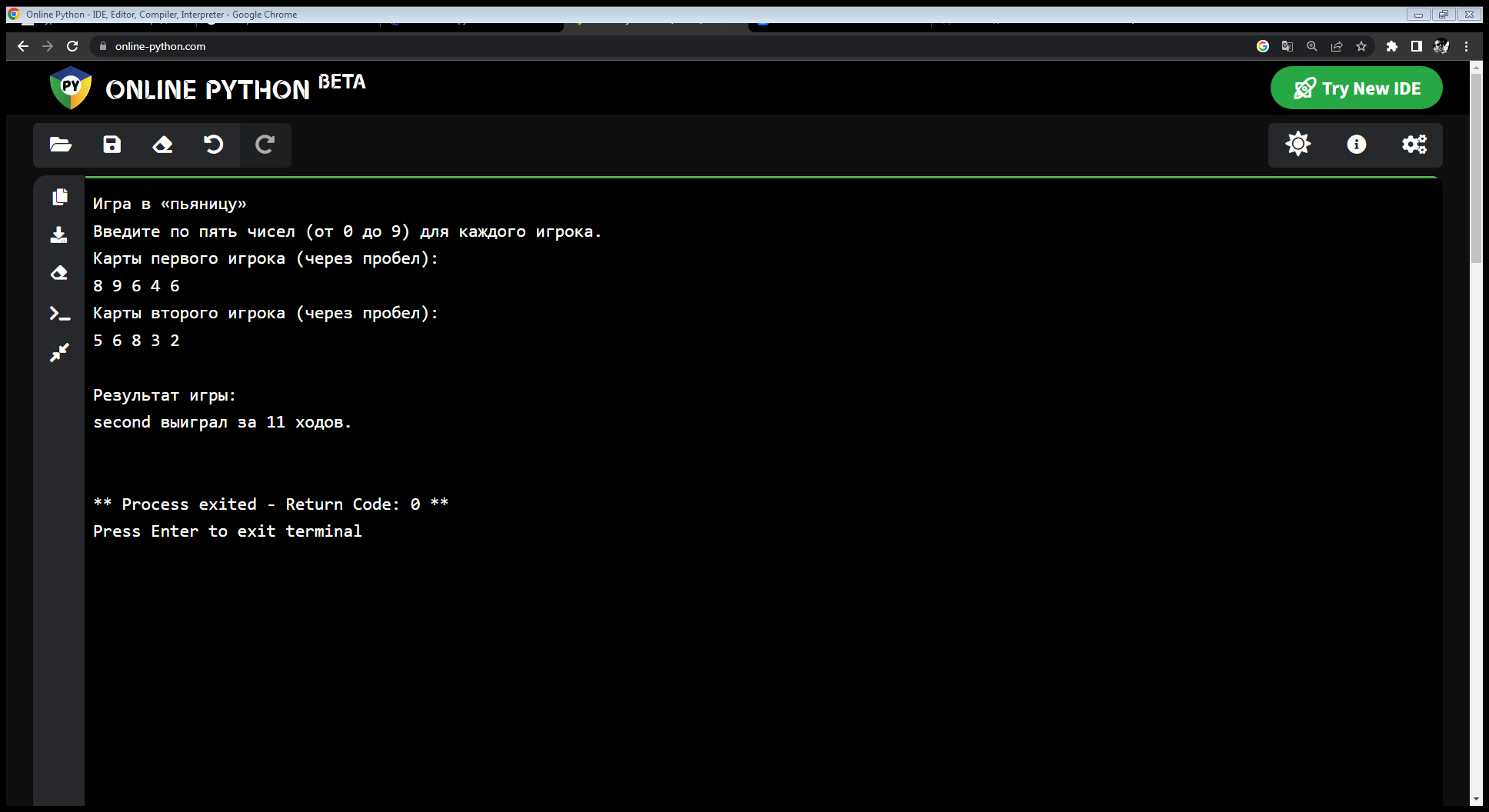
Тестирование №1:

* Ввод, при котором выигрывает первый игрок



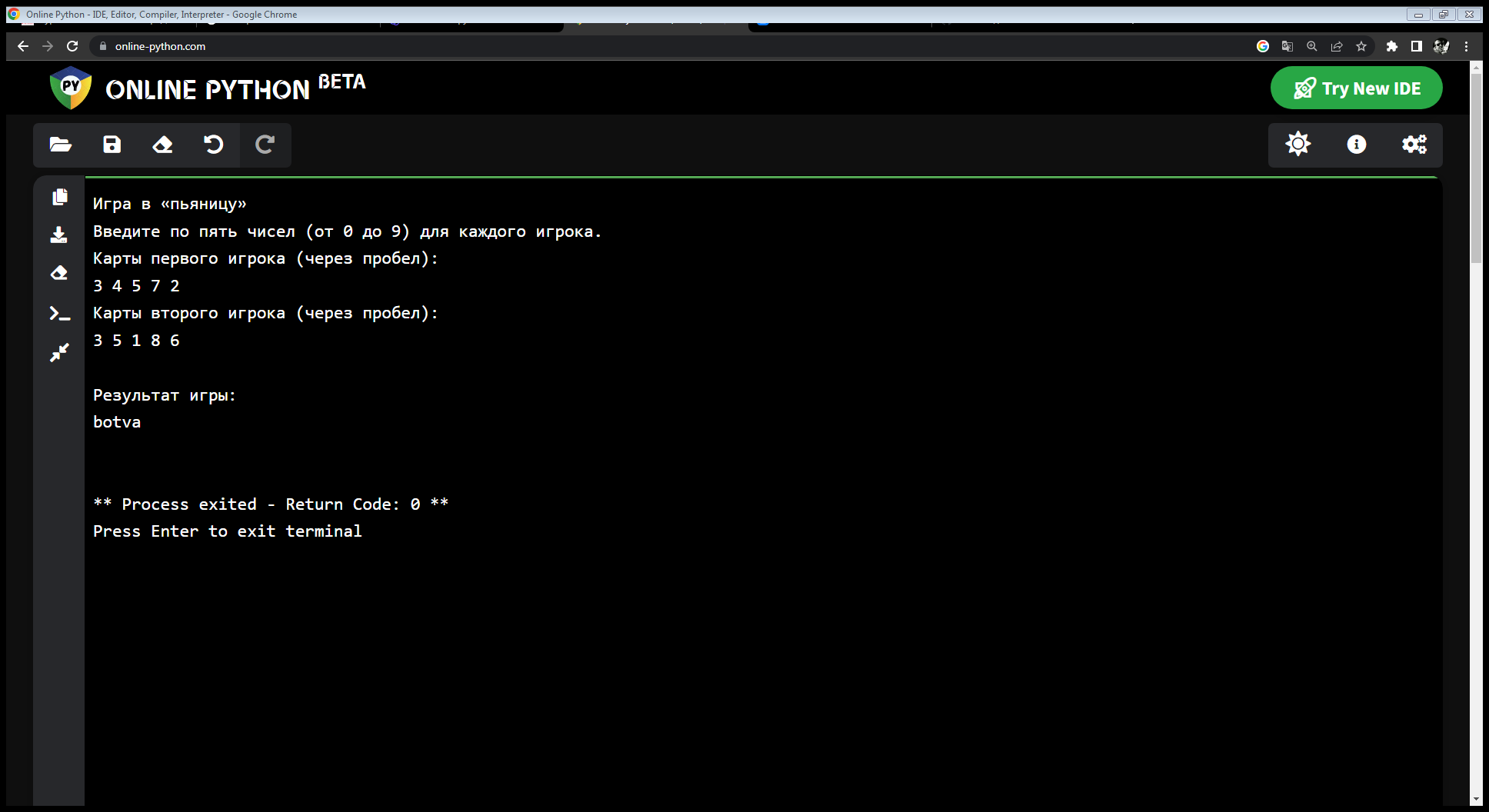
Тестирование №2:

* Ввод, при котором выигрывает второй игрок



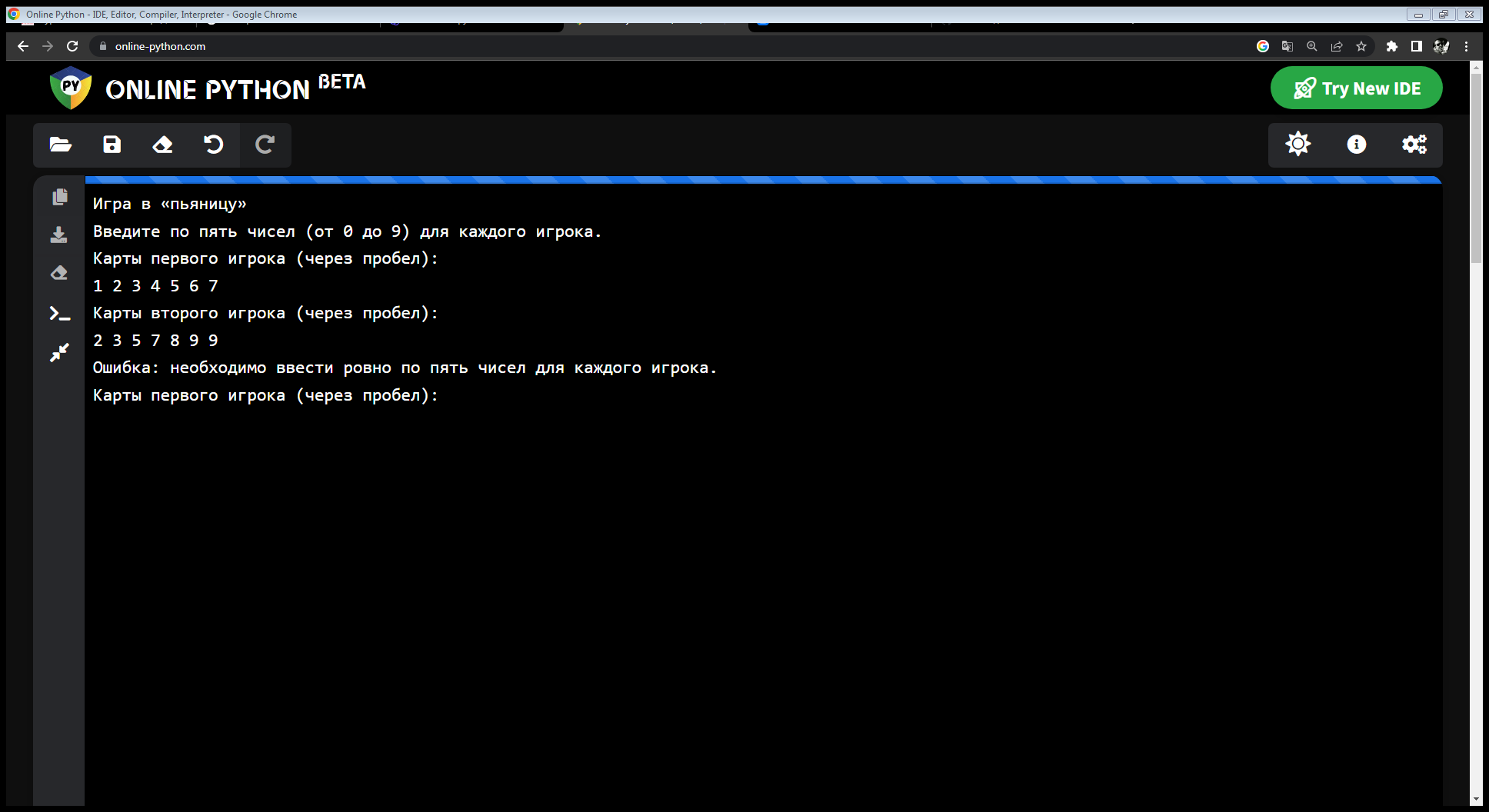
Тестирование №3:

* Ввод, при котором игра зацикливается (при котором достигнут лимит ходов)



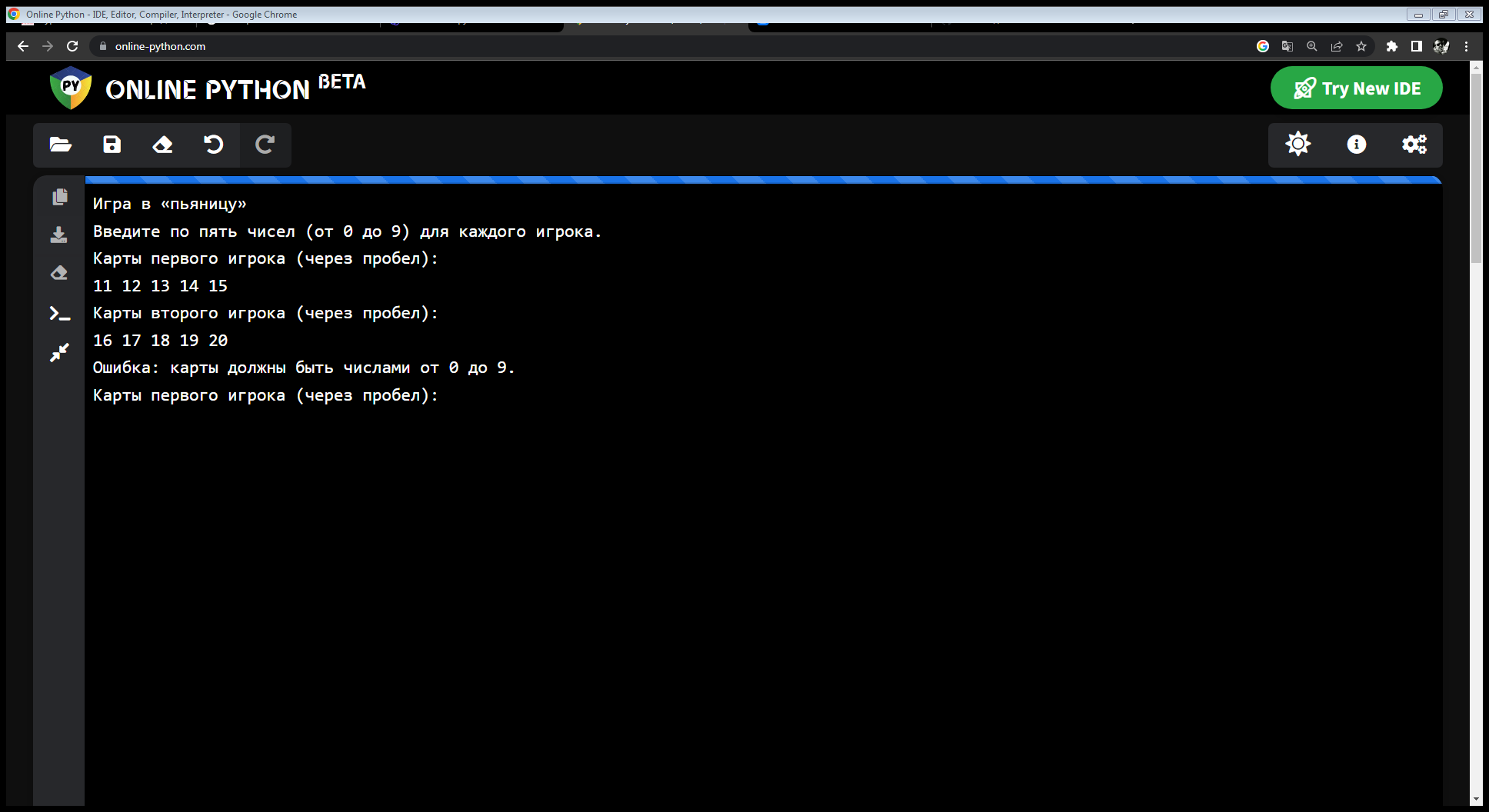
## Тестирование №4:

* Ввод, при котором программа выдает ошибку для корректного ввода определенного количества чисел, удовлетворяющих условию задачи.



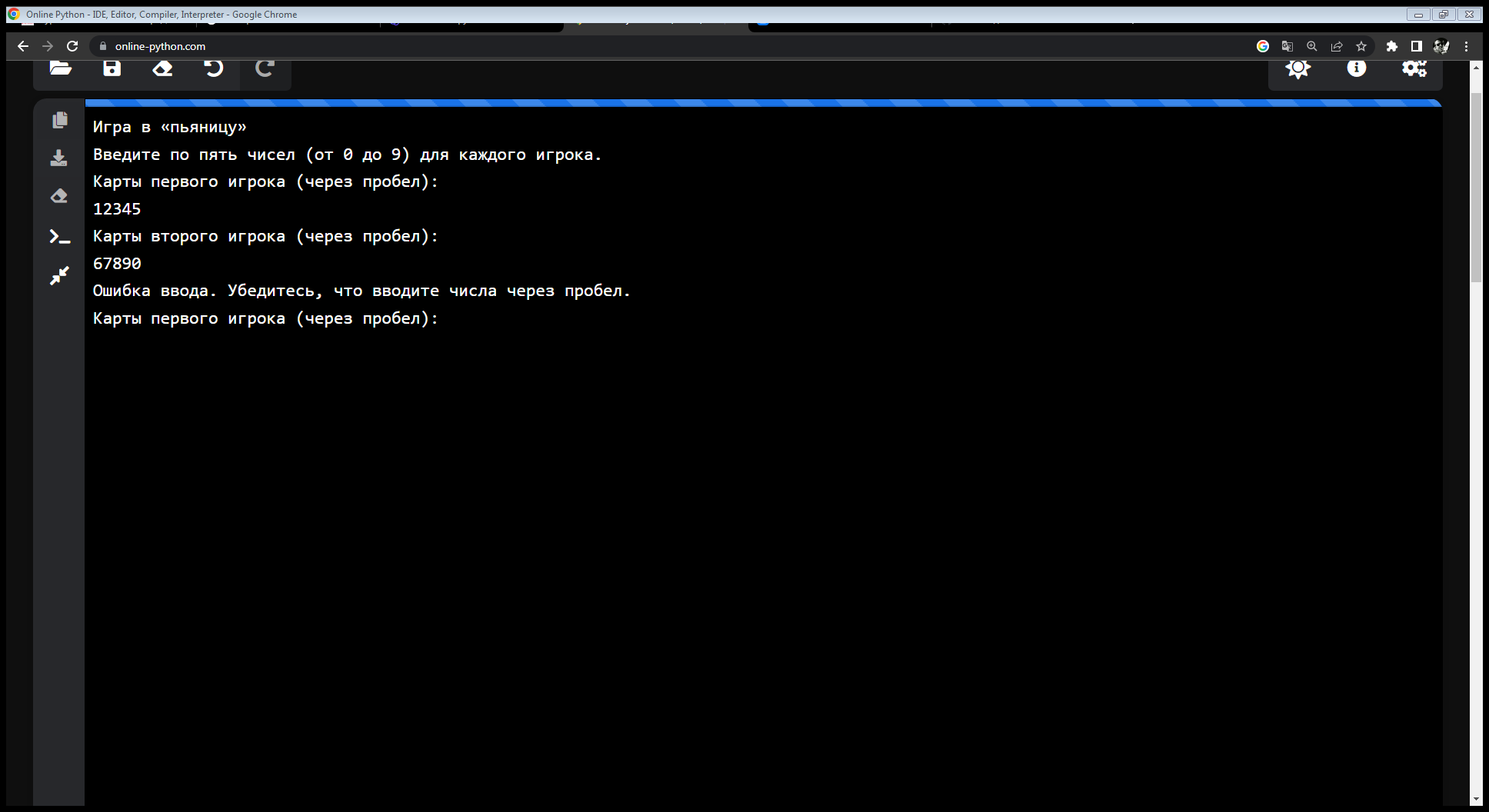
## Тестирование №5:

* Ввод, при котором программа выдает ошибку для корректного ввода чисел в диапазоне от 0 до 9



## Тестирование №6:

* Ввод, при котором программа выдаёт ошибку для корректного ввода чисел через пробел



Все тесты подтвердили, что программа работает корректно.

# Код программы

"""

Игра в «пьяницу». 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 10^6 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

"""

from typing import List

class Queue:

"""Класс реализует очередь для хранения карт."""

def \_\_init\_\_(self, items: List[int]) -> None:

self.items = items

def enqueue(self, item: int) -> None:

self.items.append(item)

def dequeue(self) -> int:

if self.is\_empty():

raise IndexError("Очередь пуста")

return self.items.pop(0)

def is\_empty(self) -> bool:

return len(self.items) == 0

def size(self) -> int:

return len(self.items)

def compare\_cards(card1: int, card2: int) -> int:

if card1 == 0 and card2 == 9:

return 1

elif card1 == 9 and card2 == 0:

return -1

elif card1 < card2:

return 1

elif card1 > card2:

return -1

else:

return 0

def play\_game(player1\_cards: List[int], player2\_cards: List[int]) -> str:

"""

Моделирует игру и возвращает результат.

"""

queue1 = Queue(player1\_cards)

queue2 = Queue(player2\_cards)

max\_moves = 10\*\*6

moves = 0

while not queue1.is\_empty() and not queue2.is\_empty() and moves < max\_moves:

moves += 1

card1 = queue1.dequeue()

card2 = queue2.dequeue()

result = compare\_cards(card1, card2)

if result == 1:

queue1.enqueue(card1)

queue1.enqueue(card2)

else:

queue2.enqueue(card1)

queue2.enqueue(card2)

if queue1.is\_empty():

return f"second {moves}"

elif queue2.is\_empty():

return f"first {moves}"

else:

return "botva"

def main():

print("Игра в «пьяницу»")

print("Введите по пять чисел (от 0 до 9) для каждого игрока.")

while True:

try:

input\_line\_1 = input("Карты первого игрока (через пробел): ").strip()

input\_line\_2 = input("Карты второго игрока (через пробел): ").strip()

cards\_player\_1 = list(map(int, input\_line\_1.split()))

cards\_player\_2 = list(map(int, input\_line\_2.split()))

if len(cards\_player\_1) !=5 or len(cards\_player\_2)!=5:

print("Ошибка: необходимо ввести ровно по пять чисел для каждого игрока.")

continue

if any(c <0 or c >9 for c in cards\_player\_1 + cards\_player\_2):

print("Ошибка: карты должны быть числами от 0 до 9.")

continue

break

except ValueError:

print("Ошибка ввода. Убедитесь, что вводите числа через пробел.")

result = play\_game(cards\_player\_1, cards\_player\_2)

print("\nРезультат игры:")

if result == "botva":

print("botva")

else:

winner, moves\_count = result.split()

print(f"{winner} выиграл за {moves\_count} ходов.")

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

main()