|  |  |  |
| --- | --- | --- |
| 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 «Язык программирования Python»  Вариант 12 | | |
|  | | Работу выполнил  студент группы ИТ-7,8-2024 1 курса  Титова А.А.  «12» июня 2025 г. |
| Работу проверил  Рубцова М.Б.  «16» июня 2025 г. |
| Пермь 2025 | | |

СОДЕРЖАНИЕ

[Постановка задачи 2](#_Toc1589271356)

[Алгоритм решения 3](#_Toc2121350064)

[Тестирование 4](#_Toc327769277)

[1. Проверка корректной работы программы 5](#_Toc1273470464)

[2. Проверка "защиты от дурака" 5](#_Toc1982405868)

[Код программы 5](#_Toc1470428837)

# Постановка задачи

Шарики. В одной компьютерной игре игрок выставляет в линию шарики разных цветов. Когда образуется непрерывная цепочка из трех и более шариков одного цвета, она удаляется из линии. Все шарики при этом сдвигаются друг к другу, и ситуация может повториться. Напишите программу, которая по данной ситуации определяет, сколько шариков будет сейчас уничтожено. Естественно, непрерывных цепочек из трех и более одноцветных шаров в начальный момент может быть не более одной.

Входные данные: даны количество шариков в цепочке (не более 105) и цвета шариков (от 0 до 9, каждому цвету соответствует свое целое число).

Выходные данные: требуется вывести количество шариков, которое будет уничтожено.

Примеры:

(входные данные)

5 1 3 3 3 2

(выходные данные)

3

(входные данные)

10 3 3 2 1 1 1 2 2 3 3

(выходные данные)

10

# Алгоритм решения

1. Запрашивается ввод данных от пользователя: количество шариков и их цвета, разделенные пробелами.  
 2. Введенные данные преобразуются в список целых чисел.  
 3. Выполняется проверка введенных данных на корректность:  
 \* Проверяется наличие введенных чисел.  
 \* Проверяется, что количество шариков находится в допустимом диапазоне.  
 \* Проверяется, что количество цветов шариков соответствует заявленному количеству шариков.  
 \* Проверяется, что цвета шариков находятся в допустимом диапазоне.  
 4. Если данные не корректны, пользователю выводится сообщение об ошибке, и запрос повторяется.  
 5. Если данные корректны, список цветов шариков сохраняется для дальнейшей обработки.

6. Создаётся объект, представляющий линию шариков, используя полученный от пользователя список цветов шариков.

7. Инициализируется счетчик уничтоженных шариков значением 0.

8. Выполняется итеративный процесс, пока в линии шариков остаются цепочки для удаления:

a. Поиск цепочек:

Выполняется поиск всех непрерывных цепочек из трех или более шариков одинакового цвета. Каждая цепочка представляется как пара индексов: начало и конец цепочки.

b. Удаление цепочек:

Если цепочки найдены:

Удаление цепочек происходит в обратном порядке, чтобы избежать смещения индексов. Из линии шариков удаляются шарики, составляющие цепочку. Счетчик уничтоженных шариков увеличивается на количество удаленных шариков.

Если цепочки не найдены:

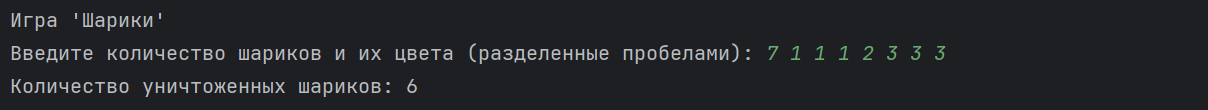
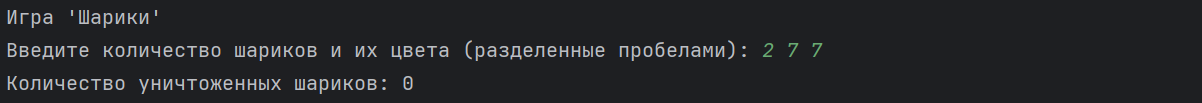
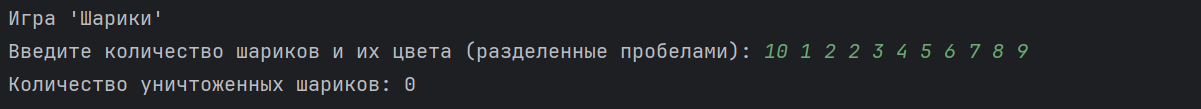
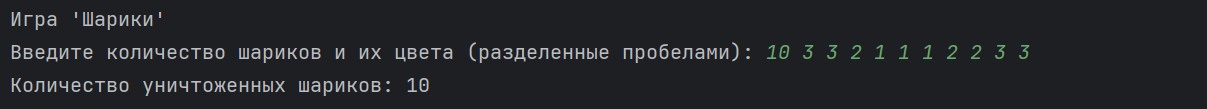
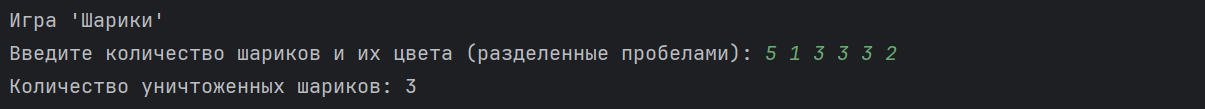
Процесс обработки линии шариков завершается.

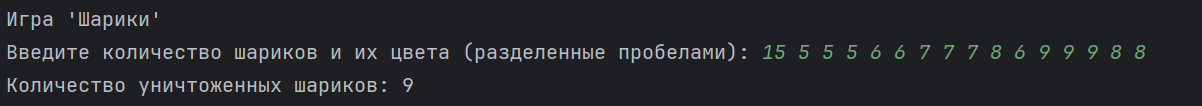
9. Выводится сообщение о количестве уничтоженных шариков.

10. Программа завершается.

# Тестирование

## 1. Проверка корректной работы программы

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

## 2. Проверка "защиты от дурака"

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

# Код программы

<https://github.com/titovalina/12-?tab=readme-ov-file#readme>