



Урок 30. Поиск экстремальных элементов

Часть 1. Материалы ученика

Программирование на Python и C++ · 8 класс

Урок 30 из 33 · 45 минут · Модуль «Марафон по строкам и массивам»

Уровень: начинающий программист; Python 3 и C++17.

Программа: <https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

В серии [-4,-1,-1,-7] максимум встречается дважды. Как сообщить обе позиции и не получить выдуманный максимум 0?

Результат урока: Найти максимум и все его индексы, обработать отрицательные значения, повторы и пустой набор в двух языках.

01. Проблемная ситуация

0-5 минут. Для [-4,-1,-1,-7] запишите максимум и индексы с нуля. Что должна вернуть программа для []? Объясните, почему начальное `best=0` не подходит.

02. Понятия и разобранный пример

5-14 минут: 4 минуты на понятия, 3 на пример, 2 на объяснение способа партнёру.

Первый элемент задаёт реальную границу

В непустой последовательности максимум обязательно равен одному из элементов. Поэтому начальное значение берём из `a[0]`, предварительно проверив пустоту. Ноль как универсальное начальное значение ошибочен для полностью отрицательных данных.

После обработки префикса `best` равен его максимуму, а `positions` содержит все индексы этого максимума. Если `x` больше `best`, прежние позиции больше не подходят: начинаем новый список `[i]`. При равенстве добавляем `i`. При меньшем `x` ничего не меняем.

Контракт: не более 100 целых от -1000 до 1000. Для пустого входа нет максимума. Python возвращает `None`, C++ печатает `empty`. Для непустого входа сохраняем порядок индексов; исходную последовательность не сортируем и не меняем. Одинаковый смысл ответа важнее разного внешнего формата.

Отступы набирайте обычными пробелами: 4 на уровень. При копировании с веб-страницы замените неразрывные пробелы обычными; полный код с обычными пробелами есть в PDF.

[-4,-1,-1,-7]: после индекса 0 состояние (-4,[0]); после 1 — (-1,[1]); после 2 — (-1,[1,2]); после 3 без изменения. При новом максимуме список очищается, при повторе дополняется.

```
def extrema(a):
    if not a:
        return None
    best = a[0]
    positions = []
    for i, x in enumerate(a):
        if x > best:
            best = x
            positions = [i]
        elif x == best:
            positions.append(i)
    return best, positions

for a in [[-4,-1,-1,-7],[],[5],[2,2],[1,3,2,3]]:
    print(a, extrema(a))
```



Индекс	0 1 2 3
Значение	-4 -1 -1 -7
Лучшее	-4 -1 -1 -1
Позиции	[0] [1] [1,2] [1,2]

Состояние после каждого шага: новая вершина сбрасывает позиции, равная добавляет индекс.

03. Практика 1. Проверяем материал

Повтор не заменяет предыдущие позиции

14–25 минут. Для шести входов сначала заполните журнал состояний, затем запустите Python. Зафиксируйте и значение, и все позиции.

Вход	Максимум; позиции
[-4,-1,-1,-7]	-1; [1,2]
[1]; [5]	None; (5,[0])
[2,2]; [1,3,2,3]	(2,[0,1]); (3,[1,3])
[0,-1,0]	0; [0,2]

2 минуты — прогнозы; 4 — запуски; 3 — замените $x > \text{best}$ на $x \geq \text{best}$ и найдите различающий тест; 2 — восстановите код. Критерии: пустота, отрицательные значения, повторы, объяснение сброса списка.

04. Практика 2. Перенос способа

Тот же контракт на C++

25–33 минуты. Перенесите контракт в C++: пустой вектор выводит `empty`, иначе максимум и индексы через пробел. За 2 минуты сопоставьте типы, за 4 завершите перенос по полному образцу ответа, за 2 проверьте [], [-5], [3,1,3] и [-2,-2]. Критерии: те же позиции, проверка `empty` до `a[0]`, неизменный вход.

05. Обсуждение и границы вывода

33–40 минут. За 2 минуты сравните поиск всех позиций с сортировкой, за 3 объясните, что станет с исходными индексами, за 2 предложите проверку полноты ответа. Достаточно ли найти хотя бы один максимум?

06. Индивидуальная проверка

40–45 минут. Для [4,1,4,4] укажите максимум, позиции, число вхождений и результат для пустого набора. Четыре пункта с обоснованием — четыре балла.



Часть 2. Ответы и материалы учителя

Практика 1: разбор и критерии

При $\geq$ каждый равный максимум заново создаёт [i]: для [2,2] останется только [1], для [-4,-1,-1,-7] только [2]. Правильно сохранять [0,1] и [1,2]. В пустом списке a[0] не читается. Все шесть ожидаемых ответов приведены в таблице.

Практика 2: разбор и критерии

Выводы: empty; -5 0; 3 0 2; -2 0 1. Вектор позиций хранит std::size_t. При равенстве используется отдельная ветка; очистка допустима только при строго большем значении. Полная программа:

```
#include <iostream>
#include <vector>
void show(const std::vector<int>& a) {
    if (a.empty()) { std::cout << "empty\n"; return; }
    int best = a[0];
    std::vector<std::size_t> positions;
    for (std::size_t i = 0; i < a.size(); ++i) {
        if (a[i] > best) { best=a[i]; positions.clear(); }
        if (a[i] == best) { positions.push_back(i); }
    }
    std::cout << best;
    for (auto i : positions) { std::cout << " " << i; }
    std::cout << "\n";
}
int main() {
    for (auto a : std::vector<std::vector<int>>
        {{},{-5},{3,1,3},{-2,-2}}) { show(a); }
}
```

Обсуждение: возможный вывод

Сортировка переставляет значения и теряет прежние позиции, если их отдельно не хранить. Проверка ответа: каждый указанный индекс содержит best; нигде нет значения больше best; ни одна позиция best не пропущена. Одного максимума недостаточно по контракту.

Индивидуальная проверка: ответ и критерии

4; [0,2,3]; 3; None в Python или empty в нашем C++-выводе. По одному баллу за пункт; последнее означает отсутствие значения, а не максимум 0.

Учителю: подготовка и план на 45 минут

Подготовьте компьютер, редактор, папку с правом записи, распечатку ученической части PDF, бумагу и ручку. Python 3.11+ и IDLE или редактор; запуск python lesson.py (либо python3 lesson.py). Внешние библиотеки не нужны. Для второй практики также подготовьте G++: g++ -std=c++17 -Wall -Wextra -pedantic main.cpp -o lesson; запуск ./lesson. Установку и проверку запуска учитель выполняет заранее. Все программы, данные и ожидаемые ответы даны; регистрация детей и интернет на занятии не нужны.

Уровень — начинающий программист, 8 класс. Предпосылки: списки и vector, индексы, циклы, сравнения. В паре автор прогнозирует и вводит код, проверяющий ведёт журнал, затем роли меняются. Без компьютера выполните трассировку полного кода на бумаге и сопоставьте с готовыми результатами; учитель демонстрирует подготовленный запуск. Бумажный вариант не считается фактическим запуском ученика. Перед практикой учитель выдаёт готовый файл разобранного примера, чтобы дети тратили время на изменение и проверку, а не длинное перепечатывание. Матрица программы предназначена учителю.



Минуты	Работа	Результат
0–5	Проблема	Вопрос и первое решение
5–14	Понятия и пример	Способ проверки
14–25	Практика 1	Готовый продукт и проверка
25–33	Практика 2	Перенос способа
33–40	Обсуждение	Аргумент и ограничение
40–45	Индивидуальная проверка	Самостоятельный ответ

07. Соответствие исходной программе

Пункт программы	Действие ученика	Проверяемый результат
Марафон по строкам и массивам	Найти максимум и все его индексы, обработать отрицательные значения, повторы и пустой набор в двух языках.	Исходник, журнал тестов и объяснение результата
Повторение и обобщение; строки, условия и коллекции	Тот же контракт на C++	Самостоятельная доработка, ключ и индивидуальная проверка

Источники и права на изображения

Исходная программа курса

<https://schoolotzyv.ru/blog/python-cpp-8-klass-programma>

Python: структуры данных

<https://docs.python.org/3/tutorial/datastructures.html>

C++: контейнер vector

<https://eel.is/c++draft/vector.overview>

Тексты, учебные данные, реплики, программы и схемы подготовлены для уроков портала «Школы России». Числовые серии — учебные модели, не результаты проведённых классом опытов. Схемы авторские, сторонние фотографии не используются. Методические ссылки приведены для учителя; ученик получает все материалы здесь. Проверка первичных источников: 18.10.2026.