| ДИСЦИПЛИНА | Технологии индустриального программирования |
| --- | --- |
| ИНСТИТУТ | ИПТИП |
| КАФЕДРА | Индустриального программирования |
| ВИД УЧЕБНОГО МАТЕРИАЛА | Контрольная работа |
| ПРЕПОДАВАТЕЛЬ | Астафьев Рустам Уралович |
| СЕМЕСТР | 1 семестр, 2024/2025 уч. год |

# Контрольная работа №5

## Формат контрольной

Каждому студенту назначаются 2 задачи по вариантам, которые необходимо решить из заданного набора. В случае, если в наборе есть совпадение, то решайте по номеру варианта +1 и так, пока не получите необходимый уникальный набор.

Например, если ваш вариант "1 - 1", то сначала к одной из единиц +1, получается "1 - 2".

## Условия заданий

ВО ВСЕХ ЗАДАЧАХ МОЖНО ИСПОЛЬЗОВАТЬ ВСТРОЕННЫЕ СТРУКТУРЫ ДАННЫХ.

1. Реализуйте структуру данных для двусвязного списка и напишите функции для добавления, удаления и поиска элементов.

2. Напишите программу, которая определяет, является ли строка палиндромом, используя стек.

3. Реализуйте очередь с помощью двух стеков. Напишите функции для добавления и удаления элементов.

4. Создайте множество, которое хранит уникальные элементы, и реализуйте функции для добавления, удаления и проверки наличия элемента.

5. Напишите программу, которая находит пересечение двух множеств.

6. Реализуйте стек с поддержкой минимального элемента. Добавьте функции для получения минимального элемента без удаления его из стека.

7. Создайте очередь с приоритетом и реализуйте функции для добавления элементов с приоритетом и извлечения элемента с наивысшим приоритетом.

8. Напишите программу, которая проверяет правильность расстановки скобок в выражении, используя стек.

9. Реализуйте функцию, которая объединяет два отсортированных множества в одно отсортированное множество.

10. Напишите программу, которая находит количество уникальных элементов в массиве, используя множество.

11. Реализуйте стек, который поддерживает операции push, pop и получение текущего размера стека.

12. Создайте очередь, которая поддерживает операцию получения среднего значения всех элементов в очереди.

13. Напишите программу, которая определяет, содержится ли одна строка в другой, используя map для хранения символов.

14. Реализуйте структуру данных "двусторонняя очередь" и напишите функции для добавления и удаления элементов с обеих сторон.

15. Напишите программу, которая находит количество повторяющихся элементов в массиве с использованием map.

16. Реализуйте стек, который может выполнять операции реверса (разворота) своих элементов.

17. Создайте очередь фиксированного размера с функцией проверки переполнения и опустошения.

18. Напишите программу, которая генерирует все подмножества множества.

19. Реализуйте функцию, которая находит максимальное значение в массиве, используя стек для хранения промежуточных значений.

20. Напишите программу, которая определяет, является ли строка анаграммой другой строки, используя map для подсчета символов.

21. Реализуйте структуру данных "множество", которая поддерживает операции объединения и пересечения множеств.

22. Создайте стек с функцией проверки на наличие дубликатов перед добавлением элемента.

23. Напишите программу, которая реализует очередь с фиксированной длиной и выводит элементы в порядке их добавления.

24. Реализуйте функцию, которая возвращает количество уникальных слов в тексте, используя map для подсчета слов.

25. Напишите программу, которая проверяет наличие циклов в односвязном списке.

26. Реализуйте стек с возможностью возврата второго по величине элемента без его удаления.

27. Создайте очередь с приоритетом и реализуйте функцию для изменения приоритета существующего элемента.

28. Напишите программу, которая находит все уникальные пары чисел в массиве, сумма которых равна заданному числу.

29. Реализуйте структуру данных "множество" с функцией получения случайного элемента.

30. Напишите программу, которая реализует функцию "выталкивания" элемента из стека без его удаления.

31. Создайте очередь, которая поддерживает операцию "peek" для получения первого элемента без его удаления.

32. Реализуйте функцию для поиска второго по величине элемента в массиве с использованием стека.

33. Напишите программу, которая находит все элементы первого множества, которые отсутствуют во втором множестве.

34. Реализуйте структуру данных "очередь", которая поддерживает возможность добавления элементов в конец и начало очереди.

35. Напишите программу, которая проверяет симметричность двусвязного списка.

36. Создайте стек с функцией "очистки" всех элементов за один вызов.

37. Реализуйте функцию для нахождения количества различных символов в строке с использованием map.

38. Напишите программу, которая реализует функцию "реверс" для очереди.

39. Создайте множество и реализуйте функцию для получения всех его подмножеств.

40. Реализуйте стек с возможностью извлечения максимального элемента за O(1).

41. Напишите программу, которая проверяет наличие дубликатов в массиве с использованием множества.

42. Создайте очередь с возможностью извлечения минимального элемента за O(1).

43. Реализуйте структуру данных "двусвязный список" с функцией поиска по значению и индексу.

44. Напишите программу, которая находит количество пар скобок в выражении с использованием стека.

45. Создайте множество и реализуйте функцию для проверки его подмножества по отношению к другому множеству.

46. Реализуйте очередь с функцией "выталкивания" самого старого элемента без его удаления из памяти.

47. Напишите программу, которая определяет количество различных комбинаций символов в строке с использованием map.

48. Создайте стек и реализуйте функцию для перемещения верхнего элемента на нижнюю позицию стека.

49. Реализуйте структуру данных "очередь" с возможностью получения всех элементов в обратном порядке.

50. Напишите программу, которая сравнивает два множества на равенство и выводит результат сравнения.