Задние по структурам данных и алгоритмов самохода Даниила Юрьевича

15.27Д-ПИ05/25б

С++

1)

#include <iostream> // Подключаем библиотеку для ввода/вывода

#include <stack> // Подключаем библиотеку для работы со стеком

using namespace std; // Используем стандартное пространство имен

int main() { // Главная функция программы

stack<int> s; // Создаем стек для целых чисел с именем 's'

// Добавляем числа в стек

s.push(10); // Кладем число 10 на вершину стека

s.push(20); // Кладем число 20 на вершину стека (теперь: 20, 10)

s.push(30); // Кладем число 30 на вершину стека (теперь: 30, 20, 10)

// Выводим и удаляем элементы

while (!s.empty()) { // Пока стек НЕ пустой, выполняем цикл

cout << s.top() << " "; // Выводим верхний элемент стека (30, потом 20, потом 10)

s.pop(); // Удаляем верхний элемент из стека

}

return 0; // Завершаем программу

}

2)

#include <iostream> // Подключаем библиотеку для ввода/вывода

#include <stack> // Подключаем библиотеку для работы со стеком

using namespace std; // Используем стандартное пространство имен

int main() { // Главная функция программы

stack<int> s; // Создаем стек для целых чисел с именем 's'

// Добавляем числа в стек

s.push(10); // Кладем число 10 на вершину стека

s.push(20); // Кладем число 20 на вершину стека (теперь: 20, 10)

s.push(30); // Кладем число 30 на вершину стека (теперь: 30, 20, 10)

// Выводим и удаляем элементы

while (!s.empty()) { // Пока стек не пустой, выполняем цикл

cout << s.top() << " "; // Выводим верхний элемент стека (30, потом 20, потом 10)

s.pop(); // Удаляем верхний элемент из стека

}

return 0; // Завершаем программу

}

JavaScript

1)

#include <iostream> // Подключаем библиотеку для ввода/вывода

#include <stack> // Подключаем библиотеку для работы со стеком

using namespace std; // Используем стандартное пространство имен

int main() { // Главная функция программы

stack<int> s; // Создаем стек для целых чисел с именем 's'

// Добавляем числа в стек

s.push(10); // Кладем число 10 на вершину стека

s.push(20); // Кладем число 20 на вершину стека (теперь: 20, 10)

s.push(30); // Кладем число 30 на вершину стека (теперь: 30, 20, 10)

// Выводим и удаляем элементы

while (!s.empty()) { // Пока стек НЕ пустой, выполняем цикл

cout << s.top() << " "; // Выводим верхний элемент стека (30, потом 20, потом 10)

s.pop(); // Удаляем верхний элемент из стека

}

return 0; // Завершаем программу

}

2)

let word = "racecar"; // Создаем строку с словом для проверки

let stack = []; // Создаем массив как стек

// Заполняем стек символами слова

for (let char of word) { // Для каждого символа в слове

stack.push(char); // Кладем символ в стек

}

// После этого стек: ['r','a','c','e','c','a','r'] (последний элемент 'r')

// Проверяем палиндром

for (let char of word) { // Для каждого символа в оригинальном слове

if (char !== stack.pop()) { // Если символ из слова НЕ равен последнему элементу стека

console.log("Не палиндром!"); // Выводим сообщение

break; // Прерываем цикл

}

}

// Если стек пустой после проверки - значит слово палиндром

if (stack.length === 0) {

console.log("Палиндром!");

}