

C Koans

ИТЕРАЦИЯ 4

Разработчики:

- Афанасьев Назар (0310, магистр)
- Бухарин Максим (2382)
- Кочуров Александр (2382)
- Муравин Егор (2382)
- Федоров Михаил (2382)
- Чепасов Дмитрий (2382)

Заказчик: Заславский Марк Маркович

Постановка цели итерации

Сроки итерации: 13.05.2025 - 26.05.2025

Главная цель итерации – подготовить версию 3 (максимально работоспособная версия)

План на итерацию

- Реализовать тему Strings (Строки)
- Добавить русификацию
- Улучшение интерфейса
- Тестирование финального продукта

Пример готовой задачи

По теме String

Заполните пропущенные значения так, чтобы программа завершилась с кодом 0.

Ответ: (штрафной режим: 0 %)

```
1 #include <stdlib.h>
2 #include <string.h>
3 #include <stdio.h>
4
5 /*
6  * Работа с C-строками (массивами символов):
7  * 1. Строка инициализируется литералом "C is the BEST!"
8  * 2. Изменение символов разными способами:
9  *   - Через индекс массива []
10 *   - Через арифметику указателей *
11 *   - Комбинированным способом
12 * 3. Проверка результата после каждого изменения
13 */
14
15 int main() {
16     // Инициализация строки (автоматически добавляется нуль-терминатор)
17     char string[] = "C is the BEST!";
18
19     // Способ 1: изменение через индекс массива
20     string[5] = 'm';
21     if (strcmp(, string) != 0)
22         return 1; // Ошибка после первого изменения
```

Проверить

Пример работы программы (фрагмент)

```
PS D:\Dev\lab\УПРНО\ебз\mse1h2025-koans> docker run --rm koans_generator --method code_tmp cond_task tmp_coderunner --name strings_assignment_task
Код: strings_assignment_task:
#include <stdlib.h>
#include <string.h>
#include <stdio.h>

/*
 * Работа с C-строками (массивами символов):
 * 1. Строка инициализируется литералом "I love C so much!"
 * 2. Изменение символов разными способами:
 *    - Через индекс массива []
 *    - Через арифметику указателей *
 *    - Комбинированным способом
 * 3. Проверка результата после каждого изменения
 */

int main() {
    // Инициализация строки (автоматически добавляется ноль-терминатор)
    char string[] = "I love C so much!";

    // Способ 1: изменение через индекс массива
    string[5] = 'N';
    if (strcmp(string, "I love C so much!") != 0)
        return 1; // Ошибка после первого изменения

    // Способ 2: изменение через арифметику указателей
    *(string + 0) = 'c';
    if (strcmp(string, "I love C so much!") != 0)
        return 2; // Ошибка после второго изменения

    // Способ 3: комбинированный доступ
    (string + 3)[0] = 'q';
    if (strcmp(string, "I love C so much!") != 0)
        return 3; // Ошибка после третьего изменения

    return 0;
}

Условие: strings_assignment_task:
Заполните пропущенные значения так, чтобы программа завершилась с кодом 0.
Coderunner: strings_assignment_task:
import subprocess, sys, re

def find_substrings(text, substrings):
    pattern = re.compile('|'.join(map(re.escape, substrings)))
    return pattern.findall(text)

def check_student_answers_for_substrings(student_answers, substrings):
    for answer in student_answers:
        found_substring = find_substrings(answer, substrings)
        if found_substring:
            print(f"Your answer has {set(found_substring)} in this answer", answer)
            sys.exit()

def replacer(match):
    key = match.group(1)
    return template.get(key, match.group(0))

student_answer = {{ STUDENT_ANSWER }}

error_messages = {'1': 'Неверное состояние строки после первого изменения (индексный доступ)', '2': 'Неверное состояние строки после второго изменения (арифметика указателей)', '3': 'Неверное состояние строки после третьего изменения (комбинированный доступ)'}

banned_words = ['exit', 'return', '[', 'goto', 'malloc', 'free']
check_student_answers_for_substrings(student_answer, banned_words)
```

Вывод всех загруженных задач

```
PS D:\Dev\lab\УРПО\еяз\mse1h2025-koans> docker run --rm koans_generator --list-tasks
=== Хранилище задач ===
* Группа: Array
  - about_array_task: Знакомство с массивом
* Группа: Base
  - basic_task: Базовое задание на сравнение чисел
* Группа: Function
  - function_basics_task: Базовое знакомство с функциями
  - function_prototypes_task: Знакомство с прототипом функций
  - function_scope_and_vars_task: Знакомство с областью видимости переменных
* Группа: Pointer
  - pointers_and_addresses_task: Знакомство с указателями и адресами
  - pointers_arrays_and_arithmetic_task: Указатели, массивы и арифметика указателей
  - pointers_as_function_arguments_task: Указатели как аргументы функций
  - pointers_function_task: Указатели на функции
* Группа: String
  - strings_assignment_task: Работа со строками
  - strings_copy_task: Копирование строк
  - strings_declaration_task: Инициализация строк
  - strings_formatting_task: Форматирование строк
  - strings_function_paramater_task: Строковые параметры
  - strings_reference_characters_task: Виды доступа к символам строки
  - strings_sizeof_strlen_task: Функции sizeof и strlen
  - strings_what_is_string_task: Знакомство со строковым типом данных
```

Результаты

- Была подготовлена версия 3
- Реализована тема Strings (Строки)
- Добавлена русификация
- Добавлена возможность вывода всех загруженных задач
- Проведено ручное тестирование всего функционала