



Лабораторная 07. Указатели, многомерные массивы

Для каждой лабораторной создавайте отдельную папку. Каждое упражнение и дополнение в отдельный файл и не загружайте на GitHub .exe файлы.

Обязательная часть. Сделать с использованием репозитория GitHub, сохраняя промежуточные результаты, в ТУИС РУДН прикрепить ссылку на финальный коммит (все задания разместить в одной отдельной директории):

Упражнение 1. (3 балла)

Напишите программу, которая считывает 10 слов с консоли и хранит их в массиве.

Необходимо реализовать 3 алгоритма лексической сортировки: сортировка выбором (Selection Sort), пузырьком (Bubble Sort), расчёской (Comb sort), и использовать их на вашем массиве. Каждый алгоритм выведет пользователю в консоль как изменяется его массив на каждой итерации алгоритма и в конце выведет сколько итераций в сумме вышло.

Напишите в комментарии в вашей программе (например, до #include) ваши ответы на следующие вопросы:

- Кто из алгоритмов отсортировал один и тот же словарь за наименьшее количество итераций? Для любого ли словаря это выполняется? Аргументируйте каждый свой ответ.
- При использовании функций из библиотеки <time.h> и подсчета времени работы каждой функции за одинаковое ли время выполнились алгоритмы? Аргументируйте свой ответ.

Требования:

- 1) Ввод только латиницы и без заглавных
- 2) Каждая сортировка должна быть оформлена в виде отдельной функции
- 3) Пользователь должен выбирать в консоле, какой алгоритм использовать, (3 опции выбора) или попросить подсчитать всеми тремя (1 опция выбора). В сумме 4 опции выбора.

Примечание: Здесь понадобятся функции strcmp(), strcpy() для работы со строками, и вам крайне поможет выделения в отдельную функцию печать в консоли вашего двумерного массива.

Упражнение 2. (1 балл)

Из предыдущего упражнения возьмите вашу функцию для сортировки расчёской и сделайте рефакторинг с помощью указателей.

Требования:

- 1) Если Вы выполнили упражнение 1, то выполните рефакторинг с дубликатом файла вашего решения для 1-го упражнения
- 2) Если Вы не выполняли упражнение 1, то напишите только функцию сортировки расчёской, а в main() продемонстрировать его работу уже с указателями



Упражнение 3. (1 балл)

Напишите программу, которая будет считать количество согласных и количество гласных в введенном пользователем слове.

Требования:

- 1) Только латиница без заглавных символов
- 2) Пользователь должен вводить только 1 слово
- 3) Работа с символом слова должна быть через указатель

По желанию. Если сделаете, загрузите на Ваш GitHub, на ТУИС РУДН прикрепить GitHub ссылку. Каждая работа будет оценена дополнительными баллами, но срок выполнения на баллы такой же, как и у основных.

Дополнение 1. (1 балл)

Функция `swap_nums()` должна поменять значения чисел `a, b`. А функция `swap_pointers()` должна поменять значения переменных `s1, s2`, но этого не происходит.

Требования:

- 1) Исправьте программу так, чтобы она работала
- 2) Объясните в комментариях рядом в чем была ошибка.

Код:

```
#define _CRT_SECURE_NO_WARNINGS
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void swap_nums(int* x, int* y);
void swap_pointers(char* x, char* y);

int main(int argc, char* argv[]){
    int a, b;
    char* s1 = (char*)malloc(200 * sizeof(char));
    char* s2 = (char*)malloc(200 * sizeof(char));
    a = 3; b = 4;
    swap_nums(&a, &b);
    printf("a is %d\n", a);
    printf("b is %d\n", b);
    strcpy(s1, "I should print second");
    strcpy(s2, "I should print first");
    swap_pointers(s1, s2);
    printf("s1 is %s\n", s1);
    printf("s2 is %s\n", s2);
    return 0;
}

void swap_nums(int* x, int* y){
    int tmp;
    tmp = *x;
    *x = *y;
    *y = tmp;
}

void swap_pointers(char* x, char* y){
    char* tmp = (char*)malloc(200 * sizeof(char));
    tmp = x;
    x = y;
    y = tmp;
}
```



Дополнение 2. (1 балл)

Параметр `const` никогда не позволит изменить значение переменной, к которому он относится. Но один программист со [StackOverflow](https://stackoverflow.com) следующим кодом решил попробовать сломать это правило:

```
#include <stdio.h>

int main()
{
    const int i = 2;
    const int* ptr1 = &i;
    int* ptr2 = (int*)ptr1;
    *ptr2 = 3;

    printf("i = %d\n", i);
    printf("i = ptr2 = %d", *ptr2);

    return 0;
}
```

Требования (на все вопросы ответьте в комментариях):

- 1) Напишите, что выдает Вам эта программа, будь это результат или ошибка. Результат логичен?
- 2) Запустите эту программу также в онлайн компиляторе gsc. Результат тот же?
- 3) Исследуйте и ответьте, что такое Неопределенное поведение (Undefined Behaviour) в программировании?

Дополнение 3. (2 балла)

Необходимо в Упражнении 1 протестировать работу трёх алгоритмов сортировки на массиве из 10 000 элементов. Создать массив можно статически, но как его заполнить словами? Поскольку материалов по обработке файлов для загрузки готового списка мы не проходили на данный момент, ограничимся случайным заполнением символами для каждого слова фиксированного размера (размер слова определяете Вы).

Требования:

- 1) Реализуйте данную программу
- 2) Ответьте снова на вопрос: при использовании функций из библиотеки `<time.h>` и подсчета времени работы каждой функции за одинаковое ли время выполнились алгоритмы? Напишите в комментариях рядом.