

Работа 2. Типы данных, функции, массивы и динамическая память

Введение

Язык программирования C широко используется для низкоуровневого программирования и разработки высокопроизводительных приложений. В этом задании вы познакомитесь с основными элементами языка C, такими как *типы данных*, *функции*, *массивы* и *динамическое выделение памяти*. Эти концепции являются важной основой для написания эффективных программ на C.

Основные темы

Для выполнения работы необходимо изучить материал по следующим темам:

1. Базовые типы данных и преобразование типов

C предоставляет различные встроенные типы данных, такие как `int`, `float`, `double`, и `char`. Эти типы данных определяют, какие значения может хранить переменная и какой объем памяти она будет занимать.

Например, целые числа обычно представляются с помощью типа `int`, а числа с плавающей точкой — с помощью `float` или `double`. Иногда нужно изменить тип данных переменной, чтобы выполнять корректные математические операции или добиться совместимости с другими частями программы. Это можно сделать с помощью преобразования типов, например:

```
float avg = (float)sum / count;
```

Подробнее с типами данных можно ознакомиться тут: <https://metanit.com/c/tutorial/2.3.php>

2. Функции

Функции в C помогают разбивать программу на более мелкие задачи, что делает код удобнее для чтения, отладки и повторного использования. Функции могут принимать параметры и возвращать значения, а их реализация позволяет избежать дублирования кода. В этом задании функции помогут нам организовать и разделить задачи программы (например, ввод данных, расчет статистики, сортировка).

Прочитать о функциях подробнее можно по этой ссылке:
<https://metanit.com/c/tutorial/4.1.php>

3. Массивы

Массивы — это последовательности данных одного типа, которые хранятся в непрерывной области памяти. В С массивы позволяют работать с группами данных, например, с набором чисел, введенных пользователем. Элементы массива можно обрабатывать по отдельности с помощью индексов, что упрощает выполнение однотипных операций над данными.

Общая информация о работе с массивами доступна по ссылке

<https://metanit.com/c/tutorial/2.13.php> . Однако необходимо также учитывать, что массивы в С построены на основе *указателей* - специальных переменных, которые хранят в себе адрес какой-то другой переменной, и таким образом на неё *указывают*. Имя массива является константным указателем на его начало, и именно указатели используются для доступа к массиву, размещенному в динамически выделяемой области памяти.

Познакомиться с указателями можно по следующей ссылке:

<https://metanit.com/c/tutorial/5.1.php> ; кроме того, ссылка <https://metanit.com/c/tutorial/5.5.php> рассказывает о связи указателей и массивов, понимание которой необходимо для выполнения данной работы.

4. Динамическое выделение памяти

В языке С возможно динамически выделять память для данных во время выполнения программы с помощью стандартных функций, таких как `malloc()` и `free()`.

Динамическое выделение памяти особенно полезно, когда размер данных заранее неизвестен, как в этом задании, где пользователю предлагается выбрать количество чисел.

Познакомиться с особенностями использования динамической памяти в С можно тут:

<https://metanit.com/c/tutorial/5.8.php>

Эти основные концепции помогут вам не только выполнить текущее задание, но и понять, как управлять памятью и структурировать код в языке С, что является важным навыком для эффективного программирования.

Задание

Напишите программу на языке С, которая:

1. Сначала запрашивает у пользователя количество чисел, которые он хочет ввести.
2. Выделяет динамическую память для хранения этих чисел.
3. Считывает числа с клавиатуры.

4. Выводит на экран следующие статистические данные:

- Сумму чисел.
- Среднее арифметическое.
- Минимальное и максимальное число.
- Количество четных и нечетных чисел.

5*. *Реализует сортировку введенных чисел по возрастанию с использованием любого метода сортировки (например, пузырьковой сортировки).*

6*. *Выводит отсортированный массив.*

7. Программа должна использовать функции для:

- Ввода массива.
- Подсчета суммы, минимума, максимума и других статистических данных.
- **Сортировки массива.*
- Вывода результатов на экран.

Требования:

1. Базовые типы данных: Для хранения чисел используйте `int`. Для вычисления среднего значения используйте преобразование типов, чтобы результат был в виде `float`.
2. Функции: Напишите функции для выполнения каждой из задач: ввода данных, подсчета статистики, *сортировки** и вывода.
3. Массивы: Вводимые числа должны быть сохранены в массиве, размер которого определяется пользователем.
4. Динамическое выделение памяти: Массив для чисел выделяется динамически с использованием функции `malloc`.

Пример работы программы:

Введите количество чисел: 5

Введите числа: 3 1 4 1 5

Сумма: 14

Среднее: 2.80

Минимум: 1

Максимум: 5

Четные: 1

Нечетные: 4

Отсортированный массив: 1 1 3 4 5

Это задание охватывает работу с массивами, базовыми типами данных, динамическим выделением памяти и функциями с использованием указателей.