

Контрольная работа №2 / Вариант 1

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 2

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 3

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 4

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 5

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 6

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 7

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива С-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 8

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива С-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 9

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива С-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 10

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 11

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 12

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 13

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива С-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 14

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива С-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 15

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива С-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 16

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 17

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 18

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 19

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)

Контрольная работа №2 / Вариант 20

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Complex** комплексного числа, хранящую два поля типа **double**: вещественную и мнимую часть. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadComplexArray(n)**, которая возвращает адрес массива **Complex** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Complex** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом выбора. (Описание алгоритма можно посмотреть в английском разделе Википедии: Selection sort.)

Контрольная работа №2 / Вариант 21

Помните, что отсутствие кода освобождения динамической памяти является грубой ошибкой. Работоспособность решений всех заданий необходимо демонстрировать в основной программе. Решения размещаются в различные файлы, как на практических занятиях.

1. Создайте структуру **Fraction** обыкновенной дроби, хранящую два целых поля: числитель и знаменатель. Создайте функции печати и ввода с клавиатуры для неё.
2. Создайте функцию **ReadFractionArray(n)**, которая возвращает адрес массива **Fraction** размера **n**, размещённого в динамической памяти, данные для него вводятся с клавиатуры.
3. Создайте структуру для представления узла односвязного списка, хранящего комплексные числа, и функции работы с таким списком (добавление в голову, удаление, печать).
4. Создайте функцию **toList**, принимающую массив экземпляров **Fraction** и возвращающую список, заполненный копиями элементов этого массива.
5. Создайте функцию сортировки массива C-строк методом пузырька. (Описание алгоритма можно посмотреть в Википедии.)