// C++ · Лабораторная 06

Подпрограммы

Пользовательские функции, передача массивов и отделение вычислений от ввода-вывода.

Цель и порядок работы

Освоить создание функций и процедур, передачу данных между подпрограммами и тестирование.

Вынесите вычисления и преобразования в функции с понятными параметрами и результатами. Ввод и вывод остаются в головной программе. Сначала определите контракт каждой функции: какие данные она получает, что изменяет и как сообщает об отсутствии результата.

1. Размеры исходных массивов задавайте именованными константами вне main; передавайте размеры функциям параметрами.
2. Для проверяемых примеров вводите элементы вручную. Случайное заполнение необязательно.
3. Ввод исходных данных и вывод результатов выполняйте в main.
4. Не используйте в подпрограммах глобальные переменные для обмена данными и рабочие массивы. Массивы передавайте вместе с размерами.
5. Сложные промежуточные результаты называйте перед вызовом. Выражения в аргументах допустимы в C++, если их смысл ясен.
6. Составьте схему алгоритма, реализуйте функции и проверьте их на небольших данных с известным ответом.

Требования к коду, заготовка отчёта, тестирование и блок-схемы →

Скачать текущие задания и варианты+ в PDF

Исходные материалы: Методичка №7 · PDF

Перед началом: Лекция 3 · создание функций · Лекция 6 · массивы в параметрах · Лекция 8 · ссылки · Лекция 7 · указатели

> «Процедуре» из методических материалов в C++ обычно соответствует функция с типом результата void. При передаче открытого массива размер передают отдельно: параметр массива не сохраняет число элементов автоматически.

Задания по вариантам

Выполните все задания назначенного варианта.

Общие требования к вариантам

Каждое задание оформите функцией с явным контрактом: входные параметры, изменяемые данные и результат либо признак отсутствия результата. Ввод и вывод выполняйте только в main. Массив и его логическую длину передавайте отдельно.

Размеры задайте именованными константами вне main, но передавайте их функциям параметрами. В функциях не обращайтесь к глобальным переменным для обмена данными. Рабочие массивы не используйте, кроме выходного, явно требуемого условием. Данные вводите вручную; случайное заполнение необязательно.

Размеры от 1 до 10, значения по модулю до 100. Индексы и слои нумеруются с 0. Второй по значению минимум и максимум означают второе различное значение. При равенстве подходящих результатов выбирайте первое в порядке обхода. «Процедура» означает функцию void.

Для нечётной матрицы центральный ромб задаётся манхэттенским расстоянием от центра. Его внешний слой имеет номер 0, следующий — 1; проверяйте допустимость номера. Для локального минимума сравнивайте только существующих соседей по сторонам.

Строковые задания решайте через std::string. Для двоичных строк ограничьте длину 30; для выражений используйте числа от 1 до 1000 и вещественное деление. Параметры могут передаваться указателями; ссылки из лекции 8 допустимы, но не обязательны.

Вариант 1

Задание 1

Создать функцию, определяющую максимальный элемент вектора values, находящийся между вторым по порядку положительным и предпоследним отрицательным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2

Создать процедуру, поворачивающую в квадратном массиве matrix (n – нечетное) средний наибольший ромб элементов на 90 градусов по часовой стрелке.

$$c = (n-1)/2, \quad |i-c| + |j-c| = c - k$$

Вариант 2

Задание 1

Создать функцию, определяющую минимальный элемент вектора values, находящийся между вторым по порядку четным и предпоследним нечетным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2

Создать процедуру, поворачивающую в квадратном массиве matrix (n – нечетное) средний наибольший ромб элементов на 90 градусов против часовой стрелки.

$$c = (n-1)/2, \quad |i-c| + |j-c| = c - k$$

Вариант 3

Задание 1

Создать функцию, определяющую среднее арифметическое значение элементов вектора values, находящихся между вторыми по значению максимальным и минимальным его элементами. Предусмотреть случай, когда таких элементов нет.

Задание 2

Создать процедуру, определяющую в целочисленном векторе values два элемента, наиболее близкие к среднему арифметическому всех элементов вектора. Элементы, расположенные между ними, отсортировать по возрастанию.

Вариант 4

Задание 1

Создать функцию, определяющую максимальный элемент k-го слоя среднего наибольшего ромба элементов в квадратном массиве matrix (n – нечетное). Номер слоя k задать с консоли.

Задание 2

Создать процедуру, вычисляющую в целочисленном векторе values сумму нечетных элементов с четными индексами и сумму четных элементов с нечетными индексами.

$$c = (n-1)/2, \quad |i-c| + |j-c| = c - k$$

Вариант 5

Задание 1

Создать функцию, определяющую в целочисленном прямоугольном массиве matrix наибольший элемент, встречающийся более одного раза. Предусмотреть случай, когда такого элемента нет.

Задание 2

Создать процедуру, определяющую в целочисленном векторе values два любых одинаковых элемента. Элементы вектора, расположенные между ними, отсортировать по возрастанию. Предусмотреть случай, когда таких элементов нет.

Вариант 6

Задание 1

Создать функцию, определяющую в целочисленном прямоугольном массиве matrix сумму всех элементов, соприкасающихся углами с его максимальным элементом.

Задание 2

Создать процедуру, переворачивающую квадратный массив matrix на 90 градусов по часовой стрелке.

Вариант 7

Задание 1

Создать функцию, определяющую в прямоугольном целочисленном массиве matrix количество всех его седловых точек. Седловой точкой матрицы называется ее элемент, одновременно наименьший в своей строке и наибольший в своем столбце.

Задание 2

Создать процедуру, сдвигающую элементы k-ого слоя целочисленного прямоугольного массива matrix на p шагов вправо. Номер слоя k и количество шагов p задать с консоли.

Вариант 8

Задание 1

Создать функцию, определяющую в квадратном целочисленном массиве matrix сумму элементов, находящихся выше и ниже главной и побочной диагоналей (“песочные часы”). Элементы диагоналей в сумму не включать.

Задание 2

Создать процедуру, формирующую вектор result из максимальных элементов диагоналей целочисленного квадратного массива matrix , параллельных его главной диагонали.

Вариант 9

Задание 1

Создать функцию, определяющую в целочисленном квадратном массиве matrix сумму элементов, находящихся левее и правее главной и побочной диагоналей (“бабочка”). Элементы диагоналей в сумму не включать.

Задание 2

Создать процедуру, перемножающую две матрицы по правилам матричного умножения. Предусмотреть случай, когда размеры матриц заданы некорректно.

$$C_{ij} = \sum_{k=0}^{p-1} A_{ik} B_{kj}$$

Вариант 10

Задание 1

Создать функцию, определяющую в квадратном целочисленном массиве matrix (n- нечетное) сумму элементов k-го слоя среднего наибольшего ромба элементов. Номер слоя k задать с консоли.

Задание 2

Создать процедуру, приводящую с помощью допустимых преобразований заданную матрицу коэффициентов системы линейных алгебраических уравнений к треугольному виду.

$$c = (n-1)/2, \quad |i-c| + |j-c| = c - k$$

Для метода Гаусса выбирайте ненулевой ведущий элемент с перестановкой строк. Если такого элемента нет, сообщите о вырожденной системе вместо деления на ноль. Храните преобразуемые коэффициенты в double; допуск 10^-9. Решение задачи с несовместной или неоднозначной системой этим сообщением завершается.

Вариант 11

Задание 1

Создать функцию, определяющую в целочисленном прямоугольном массиве matrix сумму элементов, соприкасающихся сторонами с его максимальным элементом.

Задание 2

Создать процедуру, решающую систему линейных алгебраических уравнений вида Ann* Xn = Bn Где Ann - матрица коэффициентов системы, Bn – вектор правых частей системы, Xn – вектор решения.

Для метода Гаусса выбирайте ненулевой ведущий элемент с перестановкой строк. Если такого элемента нет, сообщите о вырожденной системе вместо деления на ноль. Храните преобразуемые коэффициенты в double; допуск 10^-9. Решение задачи с несовместной или неоднозначной системой этим сообщением завершается.

$$AX = B$$

Вариант 12

Задание 1

Создать функцию, удаляющую из заданной строки все последовательности символов, находящиеся между корректно записанными круглыми скобками, вместе с этими скобками. Скобки могут быть вложенными.

Задание 2

Создать процедуру, формирующую вектор result из максимальных элементов диагоналей целочисленного квадратного массива matrix , параллельных его побочной диагонали.

Вариант 13

Задание 1

Строка – это правильное арифметическое выражение, содержащее только натуральные числа и знаки арифметических операций между ними (+, -, *, /). Создать функцию, подсчитывающую результат при выполнении арифметических операции в порядке их следования.

Задание 2

Создать процедуру, сдвигающую элементы k-ого слоя прямоугольного целочисленного массива matrix на p шагов влево. Номер слоя k и количество шагов p задать с консоли.

Вариант 14

Задание 1

Ввести две строки, состоящие только из нулей и единиц. Считая их изображениями двоичных чисел, создать функцию, складывающую их по правилам двоичной арифметики. Сумму преобразовать в десятичное число.

Задание 2

Создать процедуру, определяющую все локальные минимумы в целочисленном прямоугольном массиве matrix Найденные минимумы занести в вектор result Передайте вместимость выходного массива не меньше числа элементов матрицы и верните фактическое число найденных минимумов. Если минимумов нет, верните count=0; фиктивные нули в результат не добавляйте. Локальным минимумом называется элемент массива, который меньше своих соседних элементов, соприкасающихся с ним сторонами.

Вариант 15

Задание 1

Строка состоит из нескольких слов, разделенных пробелом. Создать функцию, определяющую минимальное количество первых символов слов, по которым можно различить слова в строке.

Задание 2

Создать процедуру, которая упорядочивает строки прямоугольного массива matrix по возрастанию количества одинаковых элементов в каждой строке.

Показатель строки во втором задании — максимальная кратность одного значения в ней: для [2,2,3,3,3] он равен 3, для строки различных значений — 1. Сортируйте по этому показателю по возрастанию; строки с одинаковым показателем сохраняют исходный порядок.

Вариант 16+ · Контракты поиска и преобразования

Функции

Напишите функцию поиска двух крайних вхождений заданного значения с отдельным признаком успеха и двумя индексами через выходные параметры. Вторая функция разворачивает участок строго между найденными индексами.

Дополнение+

Проверьте функции раздельно и вместе. Для пустого массива, отсутствия значения и одного вхождения преобразование ничего не меняет. Глобальные данные и ввод-вывод внутри функций запрещены.

Вариант 17+ · Статистика диагоналей

Функции

Напишите функцию суммы заданной диагонали квадратной матрицы, параллельной главной. Диагональ задаётся разностью номера столбца и строки. Неверный номер отклоняйте через признак успеха.

Дополнение+

Другой функцией заполните выходной массив сумм всех диагоналей; для каждой вызывайте первую. Отдельная функция находит максимальную сумму и номер её первого появления. Размеры

передавайте явно.

Вариант 18+ · Параметрический сдвиг слоя

Подпрограммы

Разделите задачу циклического сдвига периметра прямоугольной матрицы на функции: проверка слоя, длина периметра, сдвиг на один шаг, сдвиг на k шагов.

Дополнение+

Поддержите оба направления, одну строку, один столбец и центральный элемент. После сдвига вперёд и обратно матрица должна восстановиться. Ввод и вывод остаются в main.

Вариант 19+ · Композиция текстовых функций

Подпрограммы

Напишите функции: проверка круглых скобок, удаление корректных скобочных фрагментов, нормализация пробелов. Работайте со std::string; функции возвращают результат, не печатают его.

Дополнение+

Соберите конвейер в main. При некорректных скобках преобразования не запускаются. Для корректного входа выведите промежуточные строки и проверьте, что повторная нормализация уже не меняет результат.

Вариант 20+ · Умножение матриц по функциям

Подпрограммы

Напишите отдельную проверку согласованности размеров, функцию вычисления одного элемента произведения и функцию заполнения всей выходной матрицы. Последняя использует функцию элемента.

$$c_{ij} = \sum_{k=0}^{p-1} a_{ik} b_{kj}$$

Дополнение+

Добавьте проверку умножения на единичную матрицу и на нулевую. Вводите числа по модулю до 100, размеры от 1 до 10, результат храните в long long. В подпрограммах не обращайтесь к глобальным изменяемым данным.

Возможные вопросы к защите

Попробуйте ответить своими словами и показать соответствующее место в своей программе. Под вопросом можно открыть ориентир для ответа и перейти к нужной теме лекции.

▼ Из каких частей состоит определение функции?

Назовите тип результата, имя, параметры и тело; отличите определение от вызова.

Повторить: Лекция 3 · создание функций

▼ Чем функция с возвращаемым значением отличается от void?

Покажите, где результат нужен вызывающей программе и как изменяется переданный массив.

Повторить: Лекция 3 · создание функций

▼ Почему массиву-параметру нужен отдельный размер?

Объясните преобразование массива в указатель и потерю информации о длине.

Повторить: Лекция 6 · параметры-массивы

▼ Чем передача по значению отличается от передачи по ссылке?

Объясните, какие изменения видит вызывающий код, и сравните с передачей указателя.

Повторить: Лекция 8 · ссылки · Лекция 7 · указатели

▼ Как функция должна сообщить, что нужного элемента нет?

Опишите контракт результата, не смешивающий корректное значение с ошибкой поиска.

Повторить: Лекция 3 · создание функций · Лекция 5 · проверки

▼ Почему нельзя вернуть указатель или ссылку на обычную локальную переменную?

Свяжите область видимости, время жизни и висячий адрес.

Повторить: Лекция 7 · время жизни · Лекция 8 · время жизни ссылок
