

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 08.07.2025 14:09:03
Уникальный программный ключ:
e3a68f3eaa1e62674b54f4998099d3d6bdfcf836

Оценочные материалы для промежуточной аттестации по дисциплине

«Программирование»

Код, направление подготовки	09.03.02 ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ
Направленность (профиль)	Информационные системы и технологии
Форма обучения	Очная
Кафедра-разработчик	Информатики и вычислительной техники
Выпускающая кафедра	Информатики и вычислительной техники

Типовые задания для контрольной работы (1 семестр):

Отчет должен быть выполнен в текстовом процессоре Word. Листинг программ должен быть на языке C++ с комментариями.

В отчете обязательно должны быть отражены: структура алгоритма (где необходимо по заданию); результаты тестирования программы (с промежуточными результатами).

Варианты заданий в п. II- IV - согласно списка в журнале.

I) Письменно ответьте на вопросы:

Что такое массив?

Почему массив является структурированным типом данных?

Что такое размерность массива? Существуют ли ограничения на размерность массива?

Какого типа могут быть элементы массива?

Какого типа могут быть индексы элементов массива?

Какие простые типы данных относятся к порядковым?

Какими способами может быть заполнен массив? Приведите примеры.

Как определить минимальный объем памяти, отводимой под массив?

Какие действия выполняют обычно над элементами массива?

Может ли массив быть элементом массива?

В каком случае массивы совместны по присваиванию?

Пусть элементами массива A (a[1], a[2], a[3], a[4]) являются соответственно x, -x, x2, -x2. Чему будет равно значение выражения: a[-a[a[3]-2]]+a[-a[a[3]]] при x=2?

Можно ли выполнять обход двумерного массива, организовав внешний цикл по столбцам, а внутренний — по строкам?

Используются ли вложенные циклы, если совершается обход только главной диагонали квадратной матрицы?

II) Составьте алгоритм и программу. Протестируйте программу для массива, состоящего из 10 элементов. При составлении программу предусмотрите возможность просмотра, изменение переменных циклов.

1. Создайте одномерный массив натуральных чисел. Заполните его числами с клавиатуры. Выведите на экран массив и его же в обратном порядке. Найдите сумму четных элементов массива. Подсчитайте их количество. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

3. Создайте одномерный массив. Заполните его по формуле (каждый элемент массива должен быть больше его порядкового номера на 2). Выведите на экран массив и его же в обратном порядке. Найдите среднее арифметическое тех элементов массива, индексы которых являются степенями двойки. Подсчитайте их количество. Вывод массивов и чисел должен сопровождаться соответствующими надписями.

5. Создайте одномерный массив. Заполните его по формуле (каждый элемент массива должен быть больше его порядкового номера в $2i$ раза). Выведите на экран массив. Найдите произведение элементов кратных 3. Подсчитайте их количество. Вывод массива и чисел должен сопровождаться соответствующей надписью.

7. Заполнить случайными числами из диапазона $[0,1]$ вещественный линейный массив из N чисел. Найти максимальное значение и его индекс (первый, если таких значений несколько). Вывод массива и чисел должен сопровождаться соответствующей надписью.

9. Создайте массив натуральных чисел. Числа введите с клавиатуры. Найдите сумму элементов кратных 5. Подсчитайте их количество. Вывод массива и чисел должен сопровождаться соответствующей надписью.

10. Создайте одномерный массив. Заполните его случайными числами от -20 до 20. Подсчитайте сумму отрицательных и положительных элементов массива. Подсчитайте их количество. Вывод массива и чисел должен сопровождаться соответствующей надписью.

1) $\begin{array}{cccccc} 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ 1 & 2 & 3 & \dots & n \\ n & n-1 & n-2 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ n & n-1 & n-2 & \dots & 1 \end{array}$ (n-четное)	2) $\begin{array}{cccccc} 0 & 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{array}$
--	--

3) $\begin{matrix} n & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & n-1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & n-2 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & 2 & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & 1 \end{matrix}$	4) $\begin{matrix} 1*2 & 0 & 0 & \dots & 0 & 0 & 0 \\ 0 & 2*3 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 3*4 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 0 & (n-1)*n & 0 \\ 0 & 0 & 0 & \dots & 0 & 0 & n*(n+1) \end{matrix}$
5) $\begin{matrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \end{matrix}$	6) $\begin{matrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 2 & 2 & 2 & \dots & 2 & 2 & 0 \\ 3 & 3 & 3 & \dots & 3 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n-1 & n-1 & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{matrix}$
7) $\begin{matrix} 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ 0 & 1 & 1 & \dots & 1 & 1 & 0 \\ 0 & 0 & 1 & \dots & 1 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 1 & 2 & 3 & \dots & n-2 & n-1 & n \end{matrix}$	8) $\begin{matrix} 1 & 0 & 0 & \dots & 0 & 0 & 1 \\ 1 & 1 & 0 & \dots & 0 & 1 & 1 \\ 1 & 1 & 1 & \dots & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{matrix}$
9) $\begin{matrix} n & 0 & 0 & \dots & 0 & 0 & 0 \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n-2 & n-1 & n & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 1 & 2 & 3 & \dots & n-2 & n-1 & n \end{matrix}$	10) $\begin{matrix} 1 & 2 & 3 & \dots & n-2 & n-1 & n \\ 2 & 3 & 4 & \dots & n-1 & n & 0 \\ 3 & 4 & 5 & \dots & n & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ n-1 & n & 0 & \dots & 0 & 0 & 0 \\ n & 0 & 0 & \dots & 0 & 0 & 0 \end{matrix}$

IV) Составьте программу. Протестируйте программу. При составлении программу предусмотрите возможность просмотра изменение переменных циклов.

1.Создайте двумерный массив (число строк и столбцов задает пользователь). Заполните его произвольными числами с клавиатуры, выведите ваш массив на экран. Найдите сумму элементов главной диагонали Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

2. Создайте двумерный массив 3x3. Заполните ее целыми числами с клавиатуры. Выведите исходную матрицу на экран. Умножьте каждый элемент матрицы на 3, вновь выведите матрицу на экран. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

3. Создайте двумерный массив 3*5. Заполните его случайными числами от -20 до 20 , выведите ваш массив на экран. Подсчитайте для каждой строки количество отрицательных элементов Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

4. Создайте матрицу 5x5, значение каждого элемента которой равно сумме номера строки и номера столбца, на пересечении которых он находится, и вычислите сумму элементов каждой строки. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

5. Создайте двумерный массив 5*5. Заполните его случайными числами, выведите ваш массив на экран, подсчитайте сумму элементов в строках. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

6. Создайте двумерный массив 5*5. Заполните его по формуле (каждый элемент должен быть больше суммы его индексов на 1), выведите ваш массив на экран, подсчитайте произведение элементов в строках. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

7. Создайте двумерный массив 5*5. Элементы массива должны быть числами от 10 до 90. Заполните массив числами с клавиатуры. Если пользователь вводит число, не удовлетворяющее условию – программа должна сообщать ему об этом и не принимать введенное число как элемент массива. Выведите на экран массив. Подсчитайте сумму элементов в столбцах. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

8. Создайте двумерный массив 3*5. Элементы массива должны быть числами от 10 до 900(через функцию формирования случайного числа). Подсчитайте сумму элементов в столбцах. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

9. Создайте двумерный массив (число строк и столбцов произвольно). Заполните по формуле (например, $(5*i+j)$). Найдите сумму элементов в строках. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

10. Создайте матрицу 10x10, заполните ее случайными числами. Выведите матрицу на экран. Вычислите сумму всех элементов. Ввод чисел, вывод массивов должен сопровождаться соответствующими надписями.

Типовые вопросы к экзамену (1 семестр):

Модуль 1: Основы программирования на C++

1. Что такое переменная в C++? Какие типы данных вы знаете? Приведите примеры.
2. Как объявить и инициализировать переменную в C++? В чем разница между объявлением и определением переменной?
3. Какие правила именования переменных существуют в C++? Что такое область видимости переменной?
4. Объясните назначение операторов `cin` и `cout`. Как использовать их для ввода и вывода данных?
5. Какие арифметические операторы доступны в C++? Каков приоритет их выполнения?
6. Какие логические операторы доступны в C++? Приведите примеры их использования.
7. Какие операторы отношений доступны в C++? Как они используются для сравнения значений?
8. Опишите структуру простой программы на C++. Что такое функция `main`?
9. Что такое директивы препроцессора? Приведите примеры использования.
10. Что такое комментарии в C++? Зачем они нужны?

Модуль 2: Управляющие конструкции и алгоритмизация

11. Какие типы условных операторов существуют в C++? Опишите синтаксис и особенности каждого из них (`if`, `if-else`, `switch-case`).
12. Как использовать условные операторы для ветвления алгоритмов? Приведите примеры.
13. Что такое логическое выражение в условном операторе? Какие операторы могут быть использованы в логическом выражении?
14. Какие типы циклических конструкций существуют в C++? Опишите синтаксис и особенности каждой из них (`for`, `while`, `do-while`).
15. Как использовать циклические конструкции для организации повторяющихся операций? Приведите примеры.
16. В чем разница между циклами `while` и `do-while`? Когда какой цикл лучше использовать?

17. Какие операторы управления циклом доступны в C++ (break, continue)? Как они работают?
18. Как организовать вложенные и комбинированные управляющие конструкции? Приведите примеры.

19. Как обрабатывать ошибки при вводе данных? Зачем нужна валидация данных?

Модуль 3: Массивы и обработка данных

20. Что такое массив в C++? Как объявить и инициализировать одномерный массив?

21. Как объявить и инициализировать двумерный массив? Как хранятся данные в двумерном массиве?

22. Как получить доступ к элементам массива? Что такое индекс массива?

23. Как перебрать элементы массива при помощи цикла? Приведите примеры.

24. Напишите программу, которая находит минимальный элемент в массиве.

25. Напишите программу, которая вычисляет сумму элементов массива.

26. Напишите программу, которая вычисляет среднее арифметическое элементов массива.

27. Как модифицировать, удалить и вставить элементы в массив?

28. Какие типичные ошибки могут возникнуть при работе с массивами? Как их избежать?

Модуль 4: Алгоритмы сортировки

29. Что такое алгоритм сортировки? Зачем нужна сортировка данных?

30. Опишите алгоритм сортировки пузырьком. Каковы его достоинства и недостатки?

31. Опишите алгоритм быстрой сортировки (Quick Sort). Каковы его достоинства и недостатки?

32. В чем разница между сортировкой по возрастанию и сортировкой по убыванию?

33. Как использовать стандартные функции сортировки в C++ (например, std::sort)?

34. Как оценить трудоемкость алгоритма сортировки?

35. Как отладить программу, реализующую алгоритм сортировки?

Модуль 5: Алгоритмы поиска

36. Что такое алгоритм поиска? Зачем нужен поиск данных?

37. Опишите алгоритм линейного поиска. Каковы его достоинства и недостатки?

38. Опишите алгоритм бинарного поиска. Каковы его достоинства и недостатки?

39. В каких случаях можно использовать бинарный поиск?

40. Как оценить эффективность алгоритма поиска?

41. Как обрабатывать случай, когда искомое значение отсутствует в массиве?

Модуль 6: Стандартная библиотека шаблонов C++ (STL)

42. Что такое Стандартная библиотека шаблонов (STL)? Каковы ее основные компоненты?

43. Что такое контейнеры в STL? Приведите примеры.

44. Что такое алгоритмы в STL? Приведите примеры.

45. Что такое итераторы в STL? Как они используются для работы с контейнерами?