

Список вопросов к экзамену (3 семестр)

1. Переменные-указатели в C (C++). Основные операции, инициализация.
2. Константный указатель и указатель на константу. Операторы **new** и **delete**.
3. Встроенные массивы в C (C++). Структура двумерного встроенного массива. Объявление, инициализация, доступ к элементам. Связь массива и указателя.
4. Динамические массивы и указатели. Операторы **new[]** и **delete[]**.
5. Структура расположения двумерного динамического массива в свободной памяти (два способа). Операторы **new[]** и **delete[]**.
6. Арифметика указателей в C (C++).
7. Ссылки в C++. Константные ссылки.
8. Способы передачи параметров в подпрограмму: по значению, по ссылке.
9. Возвращаемые значения функций в C (C++) (правила, небезопасные операции).
10. Основные способы изменения переменной в вызывающем контексте с помощью функции. Примеры.
11. Влияние модификатора **const** при передаче параметров в функцию в C (C++) на примере константного объекта, константного указателя и указателя на константу.
12. Рекурсия и определения: простая и косвенная рекурсия, рабочая и терминальная ветвь рекурсивного алгоритма, база рекурсии и шаг рекурсии. Пояснить примером.
13. Рекурсия и определения: прямой и обратный ход рекурсии, префиксная и постфиксная запись рекурсии, индикатор завершения рекурсивных вызовов и декомпозиция. Пояснить примером.
14. Рекурсия и определения: полное дерево рекурсивных вызовов, глубина и объем рекурсии, линейная и каскадная рекурсия. Пояснить примером.
15. Рекуррентное соотношение. Итерация и рекурсия. Связь рекурсии и рекуррентных соотношений. Пример.
16. Статическая и динамическая типизация. Основные данные, связываемые с объектом в Python (счетчик ссылок, тип, значение, идентификатор). Функции **type()** и **isinstance()**, назначение, пример использования.
17. Понятие хеш (хеш-функция) и уникальный идентификатор. Функции **hash()** и **id()**, назначение, пример использования.
18. Подсчет ссылок и сборка мусора в Python, функция **getrefcount()**.
19. Элементы функционального программирования в Python: функции **map()**, **reduce()**, **filter()** и **zip()**.
20. Элементы функционального программирования в Python: генераторы списков.
21. Изменяемые и неизменяемые типы данных в Python. Примеры, сравнение. Оператор **is**.
22. Сравнение переменных в C и Python.
23. Генераторы списков в сравнении с **map()** и **filter()**.
24. Генераторные выражения.
25. Генераторы коллекций, синтаксис (для списков, множеств, словарей).
26. Интернированные объекты в Python.

27. Взаимодействие программ с вызывающим контекстом в Python (вызов по соиспользованию). Примеры.
28. Понятие итератора и итерируемого объекта.
29. Пример создания собственного итератора на базе класса с реализацией **next()** и **iter()**.
30. Типы генераторов. Генератор-функция.
31. Передача функции в качестве параметра другой функции в Python. Привести пример.
32. Указатель на функцию в C. Указатель на функцию в качестве параметра другой функции.
33. Концепция ООП, основные понятия. Понятие класса и объекта класса в общей методологии.
34. Синтаксис определения класса в Python. Объект-класс и объект-экземпляр.
35. Объекты-методы класса в Python и **self**. Особенности видимости и переопределений атрибутов-данных и атрибутов-методов. Возможные конфликты и рекомендации по именованию в классах.
36. Динамические структуры. Основные типы.
37. Динамические структуры. Связные линейные списки (одно и двунаправленные).
38. Динамические структуры. Связные линейные списки (стек, очередь, дек).
39. Динамические структуры. Связные списки (деревья). Виды обхода деревьев.
40. Динамические структуры. Деревья и способы их заполнения (формирования) из входного потока (два способа).
41. Файлы текстовые и бинарные. Сравнение. Атрибуты открытия файлов. Переопределение потоков.
42. Работа с текстовыми файлами: чтение, запись, запись по частям.
43. Работа с бинарными файлами. Модуль **Pickle**.
44. Оператор **with**. Файлы в формате CSV (устройство и пример работы).
45. Основные понятия реляционной СУБД. Что такое DB API 2.0.
46. Модуль расширения работы с БД. Объект-соединение, объект-курсор, объекты-типы.

Примеры задач:

1. Написать программу на языке C для решения рекурсивной задачи в динамическом массиве. Использовать функции для выделения (освобождения) памяти под массив, ввода массива, вывода массива на консоль, рекурсивную функцию вычислений. Задача:

Для заданного одномерного массива X из N элементов с помощью рекурсивной функции найти сумму выражений, вычисляемых по формуле X_i^3 . В рекурсивной функции каждый раз отделять левую половину от правой половины рассматриваемой части массива и применять эту же функцию к обеим частям. Рекурсивные вызовы заканчивать, когда останется только один элемент в рассматриваемой части массива.

Например, для $N = 12$: $\sum_{i=1}^{12}(X_i^3) = \sum_{i=1}^6(X_i^3) + \sum_{i=7}^{12}(X_i^3)$.

2. Написать программу на языке C для решения задачи с динамическими массивами. Использовать функции для выделения (освобождения) памяти под

массив, ввода массива, вывода массива на консоль, не менее одной функции для вычисления в массивах. Задача:

На основании заданной целочисленной квадратной матрицы $S(n,n)$ сформировать одномерный массив V , состоящий из положительных значений элементов матрицы, расположенных в ее левой половине. Обход матрицы выполнять по столбцам. Выделить альтернативу.

3. Написать программу на языке Python для решения задачи с применением подходящих функций для обработки последовательностей (`map`, `zip`, `reduce`, `filter`, `lambda`, генераторы коллекций). Задача:

Задан текстовый файл, в каждом каждой строка имеет информацию о товаре: производитель, страна, название товара, количество на складе. Необходимо отфильтровать информацию по производителю и составить словарь, где ключом будет «производитель», а значением набор список кортежей вида (страна, название товара, количество на складе). Записать полученный словарь в бинарный файл.

Замечание. Для решения задач данного типа может потребоваться: (`map`, `zip`, `reduce`, `filter`, `lambda`, генераторы коллекций), работа с текстовыми и бинарными файлами, сортировка, знание методов строк, например поиск подстроки в строке.

4. Написать программу на языке Python для решения задачи, где одна функция используется в качестве параметра другой функции. Задача:

Составить подпрограммы ($F1$ и $F2$), выполняющие вычисления в одномерных массивах по правилу: $F1$ — суммирование элементов, $F2$ — произведение элементов.

Составить подпрограмму ($F3$), принимающую на вход двумерный массив A из n строк и m столбцов, параметр in равный '0' или '1', подпрограмму (F_i) для вычислений на элементах массива, и выполняет следующие действия: если $in = 0$, формирует одномерный массив из значений, вычисленных на строках матрицы с помощью заданной подпрограммы (F_i); если $in = 1$ — действия выполняются на столбцах. Возвращает полученный одномерный массив.

Применить $F3$ последовательно к следующим наборам заданных параметров: $[A(k,d), 0, F1]$, $[B(s,s), 1, F2]$.