

Тематика и требования к решению задач для сдачи экзамена по курсу «Программирование, часть 3» в случае приема экзамена комиссией

Аннотация

Документ содержит требования, выдвигаемые к решению задач на экзамене по курсу «Программирование, часть 3», когда студент сдает экзамен комиссии. Приводится тематика задач, которые студент обязан уметь решать к концу третьего семестра обучения. Решение задач, относящихся только к указанной тематике, является необходимым для получения оценки «удовлетворительно», но не недостаточным для получения более высокой оценки.

Составители:

Старший преподаватель кафедры ИУ7 Ломовской И.В. / _____

Рассмотрено на заседании кафедры ИУ7 от 25.12.2015

Заведующий кафедрой ИУ7 Рудаков И.В. / _____

1. Общие требования к решению задач.

В зависимости от набранного количества баллов в течение семестра студенту может быть выдана не одна, а несколько задач. В такой ситуации не решение хотя бы одной из них автоматически означает получение оценки «неудовлетворительно».

Решение задачи включает в себя выделение и реализацию нескольких осмысленных функций. Оно оформляется как многофайловый проект, для сборки которого используется make-файл.

В make-файле могут не использоваться шаблонные правила, но использование переменных (например, для задания опций компиляции) является обязательным.

В случае использования динамической памяти использование утилиты Dr Memory для демонстрации отсутствия утечек является обязательным. Наличие утечек расценивается как решение задачи, содержащее ошибку.

Статические массивы можно использовать только в редких специально оговоренных случаях.

Модульные тесты желательны.

2. Основная тематика задач.

2.1. Обработка последовательностей

В таких задачах элементы последовательности обрабатываются в том порядке, в котором они поступают из последовательности, при этом каждый элемент последовательности используется не более одного раза. Длина последовательности может задаваться разными способами (указание количества элементов, признак конца и т.п.).

Примеры задач на обработку последовательностей:

- Количество четных элементов последовательности.
- Второй максимум последовательности.
- Максимальная длина монотонного фрагмента.
- Количество локальных максимумов.

2.2. Обработка массивов

Примерами *позлементной обработки массивов* служат следующие задачи:

- ввод/вывод массива;
- нахождение суммы элементов массива, произведения элементов, среднего арифметического и т.п.;
- подсчет количества элементов, отвечающих определенному условию или обладающих некоторыми признаками;

Задачи *выборочной обработки массива* схожи по формулировке с задачами *позлементной обработки*, но операции выполняются лишь над теми элементами, значение индекса которых удовлетворяет определенной закономерности (например, определить количество отрицательных элементов среди элементов одномерного массива целых чисел, стоящих на четных местах).

При решении подобных задач рекомендуется найти закономерность изменения индексов и использовать ее в решении задачи.

К задачам *переформирования массива без изменения его размера* относятся различные перестановки, в том числе сортировки. Примерами задач *переформирования массива с изменением его размера* могут служить задачи удаления или вставки элементов массива, отвечающих определенным условиям или обладающих заданными признаками.

К задачам *одновременной обработки массивов* относятся задачи слияния массивов, переписи элементов одного массива, отвечающих определенному условию или имеющих некоторый признак, в другой, формирование нового массива из элементов исходного, преобразованных по некоторой формуле или подчиняющихся определенному закону.

Примерами *поисковых задач* могут служить поиск первого отрицательного, первого положительного и любого первого элемента, отвечающего некоторому условию, поиск единственного элемента, равного некоторому конкретному значению.

2.3. Обработка матриц

Задачи данной группы аналогичны задачам обработки одномерных массивов. Приведем несколько примеров:

- Дана матрица вещественного типа. Определить максимальный элемент матрицы и его координаты в матрице. (*Позлементная обработка матрицы*)
- Дана целочисленная матрица. Определить среди четных строк, строку, имеющую наибольшее среднее арифметическое ее элементов. (*Выборочная обработка матрицы*)
- Дана целочисленная матрица. Отсортировать ее по возрастанию элементов последнего столбца. (*Переформирование матрицы*)
- Дана матрица a целого типа. Определить сумму элементов каждой строки и записать ее в новый массив s . (*Одновременная обработка матриц и массивов*)

Кроме задач, по приемам похожих на обработку одномерных массивов, есть большая группа задач, связанных с различными вариантами обхода матриц, и задач обработки разных групп элементов. Например: дана целочисленная матрица, определить сумму отрицательных элементов матрицы и их количество, среди элементов, лежащих выше главной диагонали.

Помимо умения решать указанные задачи, необходимо знать по крайней мере два способа представления динамических матриц.

2.4. Обработка строк

Выделяют задачи *посимвольной обработки строк*. В качестве примера подобных задач можно привести реализацию одной из функций стандартной библиотеки для работы со строками: `strcpy`, `strcat`, `strstr`, `strcmp`, `atoi`, `snprintf` и др.

Другую группу задач составляют *задачи обработки текста*, к которым можно отнести, например, выделение очередного слова в строке, форматирование строки (по ширине или по центру), удаление лишних пробелов между словами.

2.5. Обработка линейного односвязного списка

Основными операциями с линейным односвязным списком являются:

- добавление нового элемента к списку (элемент может добавляться в пустой список; перед первым элементом; перед заданным элементом; в конец списка);
- просмотр и обработка элементов списка;
- поиск элемента в списке;
- удаление элемента списка (удаление единственного элемента; первого элемента; элемента, расположенного за данным; последнего элемента).

Часто на основе линейного односвязного списка реализуют такие структуры данных как стек и очередь, которые, в свою очередь, используются для решения разных задач: обращение последовательности, проверка баланса скобок в выражении, разделение последовательности на части с сохранением порядка элементов и др.