

Основные рекомендации и требования по реализации курсовой работы

1. Общие требования

Надо продемонстрировать умение хранить и обрабатывать числовые данные на основе файлов и массивов, причем организованными по определенной структуре - в виде двумерных массивов. Использование библиотечных вариантов для массивов не разрешается.

По собственному выбору, пониманию или озарению можно использовать динамические массивы – их наличие или отсутствие не влияет на оценку, но может облегчить ряд операций по заданию или наоборот усложнить программу – выбор остается за автором курсовой работы.

Минимальное требование: реализовать все на основе разбиения программы на набор функций с ПАРАМЕТРАМИ (никаких глобальных переменных), как уже было рассмотрено на лекциях и практике.

ВСЕ варианты реализации массивов и функции пишутся СОБСТВЕННЫЕ - с самого начального уровня на основе выполненных 5 индивидуальных заданий, естественно с адаптацией под условия и формулировки задания на курсовую работу.

НИКАКИХ стандартных и библиотечных типов и связанных с ними функций в курсовой работе быть НЕ МОЖЕТ !

Иначе курсовая работа не будет соответствовать цели и в таком виде не будет принята, т.е. как итог - неаттестация !!!

В курсовой работе должны быть учтены все составляющие формулировки задания и выполнены указанные действия, причем в качестве результата должен быть предложен не первый попавшийся, а ВСЕ варианты, удовлетворяющие заданию. Частичная или ошибочная реализация вариантов будет оцениваться с понижением оценки.

Программа должна предоставлять возможность неоднократного выполнения действий (например, расчет площадей и выбор среди них наибольших) из перечисленных в формулировке задания - запуск программы только на одно действие (например, только расчет площадей без выявления максимальных) будет считать недостатком и влияет на оценку.

Внимательно следите за всем образующимся в памяти при выполнении всех действий при выборе для реализации динамическими массивами - память не резиновая и не нужные массивы или их фрагменты (отдельные строки в двухмерном массиве) следует своевременно удалять из памяти – отсутствие действий по освобождению памяти влияет на оценку. Аварийное завершение работы программы из-за недостатка памяти после совершения нескольких действий при обработки небольших данных считается ошибочным и влияет на оценку.

Не забывайте о реакции программы на некорректные данные или про обработку недопустимых действий (например, файл не открыт или надо что-то удалить, а оно уже отсутствует или вообще его не было) с информированием о ситуации.

Очевидно многообразие функций и разбиение текста программы на них. Допускается лишь однократный вызов конкретной функций.

НИКАКОГО ИСПОЛЬЗОВАНИЕ ГРАФИКИ НЕ ТРЕБУЕТСЯ – она не изучалась, времени на выполнение отводится немного, а принимается программа по соответствию заданию, все иллюстрации могут быть нарисованы вручную и включены в пояснительную записку !!

2. Особенности разных формулировок и реализации заданий

I. Задания в своей основе требуют обрабатывать исходный набор точек - в явном виде это указано в формулировке, но по сути обрабатываемых геометрических объектов (фигур) может появиться набор линий (прямых, частей окружностей). Все они располагаются на координатной плоскости и имеют **ВЕЩЕСТВЕННЫЕ** характеристики.

Точки задаются парой вещественных координат (x, y). Формально для них нет ограничений.

Возможно образуемые для геометрических объектов линии (прямые) целесообразно задавать тройками коэффициентов (A, B, C) прямой из канонического уравнения $Ax + By + C = 0$ (прямые могут оказаться параллельными оси OX или оси OY), а окружности очевидными центром и радиусом.

Потенциально программа должна обрабатывать произвольный (по количеству) набор точек, особенно при выборе реализации динамических массивов. Однако для больших наборов потребуется много времени на обработку, поэтому подготавливаемые примеры следует ограничивать где-то в районе 100 точек.

II. Обязательно надо выполнять проверку возможности существования обрабатываемых геометрических объектов на плоскости (есть аналогии со 2-м предварительным заданием):

- в части заданий по точкам надо будет формировать фигуру, причем очевидно, что все вершины фигуры не могут располагаться на одной прямой, а в отдельных случаях **ОБЯЗАТЕЛЬНО** важен порядок обхода вершин; также нужно принять решение как поступать при наличии одинаковых точек – считать их одной или разными и, исходя из принятого решения, формировать все фигуры (даже повторяющиеся для неоднократно встречающихся одинаковых по координатам точек), при этом надо обратить внимание на формулировки конкретных заданий

- ориентация геометрических объектов на плоскости может быть произвольной, т.е. надо учитывать не только параллельность координатным осям, но и расположение с любыми углами по отношению к ним

- «вырожденные» геометрические объекты (например, все вершины фигуры совпадают или лежат на одной прямой) не должны рассматриваться, но информация об их наличии должна быть представлена в результатах

III. Для выполнения действий надо предусмотреть как в памяти будут храниться получаемые промежуточный и окончательный результаты:

- по исходным точкам могут появляться фигуры и надо выбрать способ их хранения (заводить отдельные массивы под вершины фигур можно, но они окажутся очень большими и надо определить с привлечением комбинаторных формул число таких фигур и вершин - целесообразнее хранить порядок следования номеров вершин фигур на основе индексного массива)

- возможно для каждой из образуемых фигур будет необходимо высчитывать и хранить дополнительную величину (периметр или площадь, расстояние до точки или прямой, углы наклона и пр.) и опять же для них надо выбрать способ хранения в том же массиве или отдельном массиве

IV. Некоторые задания достаточно изящно решаются после выполнения сортировки (на лекциях пока еще не рассмотрены), например, после вычисления числовой характеристики (площадь/периметр, расстояние и пр.) и упорядочивания по не возрастанию/не убыванию легко найти минимумы/максимумы, которые будут сосредоточены в начале/конце отсортированного по соответствующей величине массива.

3. Организация ввода/вывода и соответствующего внутреннего хранения данных

Все исходные данные хранятся во входных файлах как это предполагает автор, что должно быть отражено в выбранном формате файлов.

I. Формат входных файлов с данными

Выбор набора входных файлов может быть разным и зависит от общего восприятия задания, степени освоения языка и т.п.

Для математических заданий надо хранить отдельные составляющие (координаты точек) в одном файле.

Все находится в одном файле (points.txt), причем числовые значения для конкретной точки записываются в отдельной строке («каждая точка извлекается из отдельной строки»):

222

23.5 -16.4

-136.82

56.23 656.2 78.652

-234.56 23.71 это 4-я точка

В первой строке записывается ожидаемое число точек (аналогично 4-му предварительному заданию), но можно считывать до конца файла с обязательным контролем выбранного размера массива.

ОБРАТИТЕ ВНИМАНИЕ - в отдельной строке может быть:

- только 2 числа и это полное соответствие 2-м ожидаемым координатам
- всего 1 число и это ошибочная ситуация нехватки координат для очередной точки
- чисел больше 2-х и надо считывать только первые 2 числа с пропуском оставшихся до конца строки
- чисел 2 или больше с возможным текстом и надо считывать только первые 2 числа с пропуском всех оставшихся символов до конца строки
- возможны пустые строки

НЕОБХОДИМО предусмотреть обработку этих ситуаций (аналогично рассмотренному для 5-го предварительного задания) путем пропуска «лишних» символов и переключением режима пропуска символов-разделителей. В итоге в программу должно поступить фактическое число точек при игнорировании неправильно заданных - завершение работы программы при обнаружении ошибочных строк не проводится, они просто пропускаются.

II. Формат выходных файлов с данными

Первый файл - файл протокола, куда **ПОДРОБНО** заносится каждый этап работы программы:

- на этапе обработки файлов осуществляется протоколирование некорректных ситуаций с записью соответствующего пояснения по разобранной ситуации
- контрольный вывод введенных данных (фактически заполненных массивов с указанием сформированного в них числа элементов)
- результат выполняемого действия фиксируется по каждому выполненному шагу (формирование очередной фигуры по точкам с указанием их координат, указание на пропуск несуществующей комбинации – невозможности существования фигуры, расчет числовой характеристики и результат выбора среди них (например, максимума, а выбор 0 в качестве минимума не допускается), т.е. все промежуточные результаты фиксируются подробно (желательно с текстовыми пояснениями по разбираемой ситуации), также сообщения о выявленных некорректностях при обработке и выполнении действий
- ... (по усмотрению разработчика программы или указаниям в формулировке задания)

Например:

Считано 20 точек и их координаты:

1-я точка: (2333.12; -32.53)

20-я точка (0; 23.53)

Построено 4 фигуры с точками в вершинах:

1-я фигура: (31; -32.32) (324.12; 0) (0; -1232.4)

4-я фигура: (-3.1; 2.2) (34.12; 0.322342) (-0.42; 32.321)

Их площади: ...

Второй файл - краткое представление входных данных и полученные итоговые результаты после всех выполненных действий по формулировке задания.

Например, надо найти максимальный периметр и этому может соответствовать несколько фигур (все они были обозначены в файле протокола и теперь представляются компактно):

2 фигуры имеют минимальный периметр (равен 31.532):

1-я фигура: (31; -32.32) (324.12; 0) (0; -1232.4)

4-я фигура: (-3.1; 2.2) (34.12; 0.322342) (-0.42; 32.321)

III. Формат внутреннего хранения данных

Как отмечалось в начале все данные должны храниться в массивах. Автор выбирает каким способом организуются эти массивы:

- могут быть с предопределенным размером (по константе) или динамические

- могут быть двумерные по виду данных (float POINTS[N][2], где 2 обеспечивает хранение координат x и y)

- дополнительно рассчитываемые числовые значения могут храниться в отдельных массивах, а могут быть приписаны в исходные двумерные массивы (float POINTS[N][3], где по 0 и 1 обеспечивается хранение координат x и y, а по 2 – рассчитываемое расстояние от точки до начала координат)

- могут быть собственные варианты ...

Обратите внимание, что размер массива под исходные точки существенно меньше размера под образуемые из них фигуры (расчет по комбинаторным формулам).

При этом никто не заставляет хранить для фигур координаты точек – можно хранить номера этих точек, формируя индексный массив.

Пояснительная записка

Ее содержание практически ничем не отличается от требований к полному отчету по 5-му предварительному заданию. Главное отличие по шаблону с сайта вуза – титульный лит и первые 3 содержательные страницы (задание на курсовую работу, аннотация/реферат на русском и английском языках, содержание – по разделам).

Обозначено примерное содержание пояснительной записки с последовательностью разделов, можно взять их как основу и в дополнение обязательно использовать таблицу описания функций (как в отчете к 5-му предварительному заданию):

- | |
|---|
| <ol style="list-style-type: none">1. Исходная формулировка задания2. Анализ задания и выполнение контрольного примера3. Математическая постановка задачи:<ol style="list-style-type: none">3.1. Исходные данные3.2. Ограничения на исходные данные3.3. Результирующие (выходные) данные3.4. Связь выходных данных с исходными данными4. Особенности решения задания на компьютере (указание известных ограничений, форм представления данных и т. п.)5. Форматы представления данных на внешних носителях при использовании файлов6. Внутренний формат представления данных7. Разбиение основной задачи на подзадачи и подготовка спецификации соответствующих им функций (описание назначения функций, их параметров, особенностей реализации) с табличным видом описания: |
|---|

Имя функции	Назначение	Параметры				Внешние эффекты
		входные	выходные	модифицируемые	транзитные	

8. Описание метода решения отдельных подзадач и/или выбор среди ранее подготовленных с адаптацией к выявленным особенностям задания (использование стандартным методов поиска/сортировки и т.п.).

9. Оформление описания алгоритма в виде блок-схемы (для каждой функции, должна быть представлена логика действий – ветвления и циклы, общие действия могут быть укрупнены).

10. Текст программы

11. Тестовые примеры (файлы), реализующие проверку правильности выполнения программы во всем диапазоне допустимых значений исходных данных, реакцию на некорректные данные, вывод промежуточных результатов, проверяющих правильность программной реализации предложенного алгоритма

12. Выводы по проделанной работе с анализом полученных результатов

Разделы «Анализ задания и выполнение контрольного примера» и «Тестовые примеры (файлы)» должны сопровождаться рисунками, поясняющими расположение точек и образуемых геометрических объектов (включая прямые, окружности и т.п.) и разнообразные ситуации по формулировке задания.

Итоговая оценка за курсовую работу

Оценка (от неудовлетворительно до отлично при возможной неаттестованно в случае полного несоответствия заданию) выставляется на основе следующих составляющих:

- работающей программы, выполняющей все составляющие задания или только их часть (в отличии от предварительных заданий программа курсовой работы может иметь некоторые недоделки, частично нереализованные действия, что просто влияет на итоговую оценку)
- полноты подобранных примеров (набора файлов), обеспечивающих полноту возможных ситуаций при обработке данных согласно задания
- оформления пояснительной записки и заполнения разделов (в том числе правильного оформления блок-схем и полноценного содержания всех разделов)

Программа будет рассматриваться ТОЛЬКО при наличии:

- набора файлов с исходными данными на разные ситуации
- фрагментов пояснительной записки, поясняющих предложенные наборы файлов (разделы 2, 3, 5, 11)