

**УЗБЕКСКОЕ АГЕНТСТВО ИНФОРМАТИЗАЦИИ И СВЯЗИ
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАТИКИ И
КОМПЬЮТЕРНОЙ ГРАФИКИ**

**Методические указания и задания к выполнению
курсовой работы по курсу «Информатика»**

**УЗБЕКСКОЕ АГЕНТСТВО ИНФОРМАТИЗАЦИИ И СВЯЗИ
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

**КАФЕДРА ИНФОРМАТИКИ И
КОМПЬЮТЕРНОЙ ГРАФИКИ**

**Методические указания и задания к выполнению
курсовой работы по курсу «Информатика»**

ТАШКЕНТ - 2010

1. Общая структура курсовой работы

Курсовая работа должна иметь следующую структуру:

- Оглавление.
- Лист контроля
- Теоретическая часть.
- Математическая формулировка задачи.
- Алгоритм решения задачи.
- Код программы.
- Результаты работы программы.
- Анализ результатов.

2. Порядок выполнения курсовой работы

Руководитель курсовой работы выдаёт студенту задание, с учётом его пожеланий, в котором указаны: тема работы, литература (см. приложение 1); объясняет порядок выполнения работы, сроки сдачи её, порядок оформления работы и дни консультаций. Каждому студенту выдается индивидуальный вариант задания. Студент заполняет лист контроля (см. приложение 2);

Студент имеет право получить консультацию у преподавателя кафедры – руководителя курсовой работы, а также пользоваться компьютерным классом для получения результатов своей самостоятельной работы по теме курсовой работы.

Подбор литературы

Изучение метода решения задачи. Индивидуальное задание к курсовой работе подбирается с учетом знаний по математике, информатике, компьютерной графике, полученных студентами к этому времени (на первых двух семестрах и в школе).

Из рекомендуемой литературы необходимо выбрать ту, которая наиболее полно освещает метод решения – поставленной задачи, провести их сравнительный анализ.

Составление алгоритма решения задачи

На этом этапе проводятся математические преобразования, разрабатывается алгоритм решения поставленной задачи, алгоритм представляется или в виде блок-схемы, или структурограммы (диаграммы Насси), или на псевдокоде.

Составление программы на алгоритмическом языке

Имея тщательно продуманный алгоритм решения задачи, зная методы программирования, можно приступить к написанию программы для ЭВМ.

Программа должна быть написана для решения некоторого класса задач, т.е. для решения задачи в общем виде. Исходные данные к задаче рассматриваются как контрольные. Иными словами программа должна работать исправно при любых изменениях (в допустимых пределах) начальных данных.

Любая программа, время жизни которой выходит за пределы одноразового выполнения, должна быть тщательно спланирована и построена, должна легко читаться.

Программа должна иметь внутреннее и внешнее описание (последнее называют руководством для пользователей.)

Внутренне описание (комментарии) рекомендуется разделять на две части: блок комментариев, располагаемый в верхней части программы, и пошаговые комментарии внутри программы. В блок комментариев обычно выносятся следующая информация:

1. Имя программы и ее более подробное название.
2. Фамилия, имя, группа исполнителя.
3. Сжатое изложение того, что должно быть сделано.
4. Ожидаемые входные данные.
5. Имена и назначение основных переменных.
6. Вызываемые функции (подпрограммы).
7. Условия, которые могут возникнуть при выполнении программы.

Следующие рекомендации способствуют эффективному документированию внутри программы:

1. Нет необходимости описывать каждый оператор отдельным комментарием. Комментарии должны помогать следить за выполнением программы.
2. Используйте комментарии для разделения программы на отдельные подразделы. Такие подразделы содержат обычно от 5 до 10 операторов и имеют одно - целевое назначение.
3. При расположении комментария полезно делать отступ от текста программы.
4. Использование достаточного количества пробелов и отступов в тексте программы, позволяют сделать программу более выразительной.
5. Выбирайте осмысленные имена $X = X + A$ дает гораздо меньше информации, чем $summa = summa + A$.

Во всех программах есть ошибки. Ошибки могут возникнуть от разных источников, например, от непонимания задачи, плохо продуманной программы, набивки операторов с ошибками и т.п. Труднее всего обнаружить семантическую ошибку, которая не может быть обнаружена при трансляции (компиляции) программы.

В программе следует по возможности предусмотреть защиту от ошибок, руководствуясь следующими соображениями:

1. Минимизируйте ограничения на ввод. Следите за допустимыми значениями входных данных, во многих случаях существует четкий диапазон входных значений. Например, число рабочих часов служащего в неделю должно находиться в диапазоне между 0 и $7 \cdot 24$ часа, а вес человека – между 1 и 200 кг. В этом случае программа должна проверять каждое входное значение.
2. Нужно предвидеть возможные ошибки. При планировании программы определите места, где могут возникнуть ошибки. Например, при выполнении операции деления А на В необходимо проверить, не имеет ли В нулевого значения, аргумент функции выходит за пределы ее области определения и т.д.
3. Всякий раз, когда программа сталкивается с неприемлемой ситуацией любого вида, следует печатать сообщение об ошибке и прекращать ее выполнение.

2.4. Отладка программы

Отладка – это кропотливая исследовательская, экспериментальная работа, часто требующая много времени и терпения.

В процессе обучения программированию на занятиях студент должен научиться понимать диагностические сообщения, выдаваемые компилятором. Часто компилятор (транслятор) указывает на ошибку, которая является следствием другой ошибки, компилятор может и не заметить ошибку, например, семантическую ошибку.

Следующие советы могут успешно провести этот этап выполнения курсовой работы.

1. Выделите ключевые точки внутри программы, когда вы первый раз пишете программу, включите операторы вывода в точки перехода от одной фазы к другой. Если в программе произошла ошибка, вы будете иметь представление, в каком месте ее искать. После отладки программы вспомогательные операции вывода всегда можно будет изъять из программы.
2. Избегайте многократного вставления и выбрасывания отладочных операторов. Легче отлаживать правильный алгоритм, чем неправильный. Гораздо легче переделать или даже вновь написать программу, чем заново переработать ее алгоритмы.
3. Для отладки программ не было найдено пока ничего лучше, чем «ручное» моделирование процесса выполнения программ, «играть роль машины». Вы записываете значения переменных и прослеживаете за ходом работы программы. При этом сравниваете каждое полученное при трассировке значение со значением, которое та или иная переменная должна иметь на соответствующем шаге алгоритма.

4. Выводите на печать столько сведений, но не настолько много, чтобы утонуть в бесполезной информации.
5. Полученные результаты проанализируйте.

2.5. Контрольный пример

Описывается в полном объеме работа программы для конкретных данных, приводится результат и его анализ.

2.6. Защита курсовой работы

Выполненную курсовую работу до начала сессии студент сдает на проверку, готовит текст доклада, и презентацию на 3-5 минут к защите курсовой в назначенный срок. При защите могут быть заданы вопросы к содержанию его сообщения, а также вопросы общего теоретического характера, рейтинговая карта приведена в приложении 3.

3. Требования к оформлению, подготовка к защите работы

- Работа выполняется по теме, заданию и вариантам, выданными преподавателем. Пример задания приводится в приложении 4. Список утвержденных на кафедре тем приводится в приложении 5.
- Работа оформляется в текстовом редакторе MS Word, размер шрифта-14, поля: слева – 3 см., справа – 2 см., сверху и снизу по 2.5 см., межстрочный интервал – полуторный.
- Курсовая работа распечатывается на белой бумаге формата А4, материал размещают только на лицевой стороне каждого листа.
- Первой страницей является титульный лист, который заполняют по установленной форме (см. приложение 6.).
- Все страницы нумеруются начиная с-2 (т.к. титульный лист-1)
- В общую нумерацию страниц не входит список литературы и приложения.
- Каждый раздел начинается с новой страницы.
- Работа подписывается автором и ставится дата.
- За 20 дней до начала сессии работа, сшитая в скоросшивателе, сдается для проверки преподавателю и в назначенный день защищается.

К защите курсовой работы студент готовит доклад, рассчитанный на 3-5 минут.

**Типовые темы курсовых работ по курсу «Информатика»
для студентов 1-го курса всех направлений**

№ темы	НАИМЕНОВАНИЕ ТЕМЫ
1	Алгоритмы и программы обработки массивов в СИ++, Турбо Паскаль.
2	Программирование игр в различных средах.
3	Алгоритмы и программы реализации численных методов в среде Си ++, Турбо Паскаль. Вывод результатов в виде графиков и таблиц.
4	Углубленное изучение отдельных разделов программирования на С++, Турбо Паскаль.
5	Классы стандартной библиотеки С++ (описание и возможности конкретного класса)
6	Разработка мультимедийных документов по заданной тематике (Power Point, HTML).
7	Алгоритмы и программы сортировки последовательностей (Методы Хоара, прямой вставки. Шелла, пирамидальная сортировка и т.д.)
8	MS EXCEL, алгоритмы реализации численных методов в среде (нелинейные уравнения, аппроксимация, оптимизация)
9	Объектно - ориентированное программирование в среде Дельфи
10	Практика программирования в среде Турбо Паскаль, С ++ (Обработка результатов психологического опроса, работа с файлами, Записи, Структуры, Строки)
11	Создание Web страницы, посвященной ученым Узбекистана, работающим в области информатики
12	Создание сборника тестов для промежуточных контролей по темам «Программирование на С++, Turbo Pascal»
13	Алгоритмы и программы изображения различных примитивов в компьютерной графике
14	MS ACCESS.Проектирование и создание баз данных по теме «Абитуриент», «Студенты».
15	Delphi, создание пользовательского модуля «Сортировка»
16	Темы по выбору.

ОБРАЗЕЦ ЗАДАНИЯ

УТВЕРЖДАЮ
Зав. кафедрой
«Информатика и КГ»

Задания к курсовой работе по информатике

Ф.И.О. преподавателя доц. Идиатулина Э.М.

Для студентов факультета АТ – гр.223, 224, 225, 226-252, 283, 293-10.

ТЕМА №1

СИ. Алгоритмы и программы обработки массивов

ЗАДАНИЯ:

Переписать три задачи из приложения 4 каждого уровня, соответственно номеру своего варианта и выполнить согласно изложенным выше требованиям (раздел 1), причем:

Задача 1. Использовать изученные алгоритмы для конкретного размера массива, в программе предусмотреть ввод и печать исходных данных и результата с поясняющим текстом, привести список использованных переменных, подготовить тесты, проверить работу программы.

Задача 2. Требуется применения рассмотренных на лекциях алгоритмов и их сочетаний. Программу необходимо составить в общем виде так, чтобы она могла быть использована для обработки массивов различных размеров, привести список использованных переменных, подготовить тесты.

В программе предусмотреть ввод и печать исходных данных и результата с поясняющим текстом, проверить работу программы.

Задача 3. Требуется, кроме всего прочего, также некоторого творческого подхода. Вывод осуществить по формату массивов.

Перечень задач и методические указания к ним имеются у преподавателя.

Литература

1. Методические указания к выполнению курсовой работы по информатике – электронная библиотека ТУИТ.
2. Конспект лекций по информатике – алгоритмический язык СИ.
3. С/С++. Программирование на языке высокого уровня /Павловская Т.А. –СПб.: Питер, 2003.

ТЕМА №2

С++. Решение задач с использованием массивов

Решение большинства реальных задач на ЭВМ требует организации данных в виде массивов. В решении задач данного задания использовать принципы модульного программирования.

ЗАДАНИЯ:

Переписать три задачи по одной из каждого уровня, соответственно номеру своего варианта.

Задача 1. Выбрать способ представления данных и составить список используемых переменных. Сформулировать задачу математически. Составить алгоритм и программу. Подготовить тесты.

Задача 2. Выбрать целесообразный способ представления данных. Сформулировать задачу математически. Составить алгоритм и программу, особое внимание обратить на вывод результатов в наглядной форме.

Задача 3. Это задача требует использования отдельных функций в том числе и для пунктов 2 – задания.

Перечень задач и методические указания к ним прилагаются (приложение 4).

Литература

1. Методические указания к выполнению курсовой работы по информатике - электронная библиотека ТУИТ.
2. Конспект лекций по информатике – алгоритмический язык СИ.
3. С/С++. Программирование на языке высокого уровня /Павловская Т.А. -СПб.: Питер, 2003.
4. Лабораторные работы.

ТЕМА № 3

Программирование игр

Разработка игровых программ дает большой эффект при обучении программированию, так как вызывает интерес обучаемых и требует использования разнообразных приемов программирования и средств языка.

ЗАДАНИЯ:

1. Переписать три задачи по одной из каждого уровня, соответственно номеру своего варианта.
2. Изучить различные способы получения случайных чисел, способы программирования простых игр, методы пошаговой детализации.
3. **Задача 1.** Разработать алгоритм и программу для соответствующего способа получения случайных чисел.

Задача 2. Разработать алгоритм и программу для указанной простой игры или фрагмента, используя программы 1-задачи. В программе

предусмотреть вывод соответствующих сообщений для организации диалога с компьютером.

Задача 3. Составить законченную программу, используя программы 1 и 2 – задач.

4. Отладить программу, получить тестовые решения.
5. Приложить код программы.

Перечень задач и методические указания к ним имеются у преподавателя.

Литература

1. Методические указания к выполнению курсовой работы по информатике - электронная библиотека ТУИТ.
2. Конспект лекций по информатике - алгоритмический язык СИ.
3. С/С++. Программирование на языке высокого уровня./ Павловская Т.А. -СПб.: Питер, 2003.

ТЕМА №4

***Алгоритмы и программы решения нелинейных уравнений.
Вывод результатов в виде графиков и таблиц***

ЗАДАНИЯ:

1. Изучить методы численного решения нелинейных уравнений.
2. Переписать конкретные уравнения из приложения 4.
3. Составить алгоритмы к заданным методам, предполагая их оформление в виде отдельных функций.
4. К каждой функции написать инструкцию, содержащей: назначение, используемый метод, переменные, используемые функцией.
5. Требования к вызывающей программе.
6. Составить программу на алгоритмическом языке.
7. Отладить программу, используя подготовленные тесты.
8. Получить решения для заданных уравнений, проанализировать результат.

Литература

1. Демидович Б.П., Марон И. А. Основы вычислительной математики М.Наука,1990.
2. Конспект лекций по информатике - алгоритмический язык СИ.
3. С/С++. Программирование на языке высокого уровня./ Павловская Т.А. -СПб.: Питер, 2003.
4. Воробьева Г. Н., Данилова А. Н. Практикум по вычислительной математике. М.: Высшая школа, 1980.

ЛИСТ КОНТРОЛЯ ВЫПОЛНЕНИЯ КУРСОВОЙ РАБОТЫ ПО ИНФОРМАТИКЕ

РУКОВОДИТЕЛЬ
СТУДЕНТ
ГРУППА
ТЕМА

ДАТА ПОЛУЧЕНИЯ

ПОДПИСЬ

СРОКИ ВОПЛНЕНИЯ

	ФЕВРАЛЬ	МАРТ	АПРЕЛЬ	МАЙ
ПЛАН %	до 20%	до 50%	до 90%	100%
ДНИ ОТЧЁТА				
ВЫПОЛНЕНИЕ				
ПОДПИСЬ				

ЗАМЕЧАНИЯ ПРЕПОДАВАТЕЛЯ

Таблица распределения рейтинговых баллов по видам выполнения курсовой работы (разработана доцентами кафедры)

<i>Виды работ</i>	<i>Сроки</i>			
	<i>февраль</i>	<i>март</i>	<i>апрель</i>	<i>май</i>
1. Выполнение курсовой работы по этапам	До 20% 06-10 бал	До 50% 0- 30 бал	До 90% 0-45 бал	сумма 30- 45 баллов
2. Оформление курсовой работы	-	-	5 -10 баллов	
Итого				35-55 баллов
Защита курсовой работы	20 - 45 баллов			
Четкое и грамотное изложение проделанной работы, содержания работы. Ответы на вопросы	10 -25 баллов			
Использование в программе подпрограмм, функций, модулей. файлов, наличие комментариев. рациональность.	10-20 баллов			
Всего по курсовой работе	55- 100 баллов			

**УЗБЕКСКОЕ АГЕНТСТВО ИНФОРМАТИЗАЦИИ И СВЯЗИ
ТАШКЕНТСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Кафедра информатики и компьютерной графики

**КУРСОВАЯ РАБОТА
по курсу «Информатика»**

Выполнил (а) студент (ка)
группы _____
Ф.И.О.
Преподаватель
Ф.И.О.

Ташкент-2011

Варианты к теме 2
C++. Решение задач с использованием массивов.

ЗАДАНИЕ 1 уровня.

1. Выбрать способ представления данных и составить список используемых переменных. Сформулировать задачу математически.
2. Определить какой из рассмотренных на занятиях типовых алгоритмов может быть использован для решения задачи.
3. Составить блок-схему и программу.
4. Подготовить тесты. Проверить работу программы на ПК.

Варианты задач 1 уровня.

1. Информация о количестве выпавших в течение месяца осадков задана в виде массива. Определить общее количество осадков за месяц.
2. Информация о температуре воздуха за месяц задана в виде массива. Определить, сколько раз температура опускалась ниже 0°C.
3. Информация о средней суточной температуре воздуха за месяц задана в виде массива. Определить, температура скольких дней была ниже среднемесячной.
4. Регистрация направления ветра на горном плато производится один раз в день по очереди двумя исследователями. Каждый месяц все результаты сводятся в одну таблицу. Составить программу, выполняющую эту операцию.

Указание. Направление ветра кодируется следующим образом:

1 - северный	5 - северо-западный
2 - южный	6 - северо-восточный
3 - восточный	7 - юго-западный
4 - западный	8 - юго-восточный

5. Информация о количестве осадков, выпавших в течение месяца, и о температуре воздуха задана в виде двух массивов. Определить, какое количество осадков выпало в виде дождя, какое в виде снега. (Считать, что идет дождь, если температура воздуха >0°C).
6. Рост учеников класса представлен в виде массива. Рост девочек кодируется знаком +, рост мальчиков знаком «минус». Определить средний рост мальчиков.
6. В области 10 районов. Известны площади, засеваемые пшеницей, и средняя урожайность (ц /га) в каждом районе. Определить количество пшеницы, собранное в области, и среднюю урожайность по области.

7. В области 10 районов. Заданы площади, засеваемые в каждом районе пшеницей, и урожай, собранный в каждом районе. Определить среднюю урожайность пшеницы по каждому району и по области в целом.
8. Результаты переписи населения хранятся в памяти ЭВМ. Используя массивы фамилий и года рождения, напечатать фамилии и подсчитать общее число жителей, родившихся раньше 1978 года.
9. Ртутные термометры могут использоваться для измерения температуры до $-39,4$ °С. Используя информацию о минимальной температуре, зафиксированной в каждом году из последних 100 лет в городе ХХХХ, за полярным кругом, определить, можно ли поставлять ртутные термометры в этот город.
10. Задан список участников соревнований по плаванию и их результаты. Напечатать фамилию и результат чемпиона.
11. В памяти ЭВМ хранится информация о валовом сборе зерна по областям республики отдельно за 2008 и 2009 гг. Определить суммарный сбор зерна по каждой области республики за 2 года.
12. Задан список участников соревнований по плаванию и их результаты. Расположить результаты и фамилии участников в соответствии с занятыми местами.
13. Фамилии участников соревнований по фигурному катанию после короткой программы расположены в порядке, соответствующем занятому месту. Составить список участников в порядке их стартовых номеров для произвольной программы (участники выступают в порядке, обратном занятым местам).
14. При поступлении в институт лица, получившие двойку на первом экзамене, ко второму экзамену не допускаются. Считая фамилии абитуриентов и их оценки после первого экзамена исходными данными, составить список абитуриентов, допущенных ко второму экзамену.

Указания к решению задач I уровня.

1. Задача сводится к суммированию элементов массива. Максимальный размер массива - 31 (наибольшее число дней в месяце). Количество дней конкретного месяца ввести оператором ввода (при желании можно количество дней в месяце определять программно по его названию).
2. См. указание к задаче 1.
3. Задача сводится к комбинации двух алгоритмов: суммирование элементов массива для определения среднемесячной температуры и определение количества элементов массива, удовлетворяющих заданному условию. См. также указание к задаче 1.
4. Задача сводится к объединению двух массивов в один с чередованием элементов исходных массивов. См. также указание к задаче 1.

5. Задача сводится к суммированию элементов массива осадков в двух переменных. Элемент массива осадков должен прибавляться к одной из двух сумм в зависимости от значения соответствующего элемента массива температур. См. также указание к задаче 1.
6. Задача сводится к вычислению суммы и количества отрицательных чисел в массиве, после окончания суммирования знак суммы нужно изменить.
7. Площади, засеваемые пшеницей, и урожайность для каждого района нужно задать в двух массивах. Количество пшеницы, собранное в области, вычисляется как сумма произведений элементов двух массивов, а для получения средней урожайности нужно вычисленное общее количество пшеницы разделить на сумму элементов массива площадей.
8. Исходные данные — урожай в каждом районе и засеваемые площади - задать в двух массивах. Для определения урожайности по каждому району найти частное от деления элементов массивов, содержащих исходные данные. Для определения средней урожайности по области найти частное от деления сумм элементов массивов исходных данных.
9. Задача сводится к определению числа элементов массива «года рождения» со значениями меньше 1978 и вывода на печать соответствующих элементов массива фамилий.
10. Задача сводится к определению того, есть ли в массиве минимальных температур хотя бы один элемент, меньший или равный -39,4.
11. Задача сводится к определению минимального элемента массива результатов и его индекса, и выводу на печать элемента массива фамилий с этим индексом.
12. Задача сводится к суммированию двух массивов.
13. Задача сводится к упорядочению массива результатов в порядке возрастания, но при выполнении перестановки в массиве результатов нужно переставлять соответствующие элементы и в массиве фамилий.
14. Задача сводится к инвертированию массива фамилия участников соревнований.
15. Задача сводится к формированию массива из заданного включением в него только элементов, удовлетворяющих заданному условию (условие проверяется для соответствующего элемента массива оценок).

Задание II уровня.

Выбрать целесообразный способ представления данных, используя, если нужно, и двумерные массивы. Составить список используемых переменных. Сформулировать задачу математически. Разработать алгоритм решения задачи, используя, если возможно, типовые алгоритмы, изложенные в теоретическом введении. Решение многих задач не может быть полностью сведено к типовым алгоритмам и требует творческого подхода.

Составить схему и программу. Особое внимание обратить на вывод результатов в наглядной форме - с подробными пояснениями, если можно - в виде таблиц. Выполнить программу вручную для некоторых наборов данных. Проверить работу программы на ПК.

Варианты задач II уровня.

1. При выборе места строительства жилого комплекса при металлургическом заводе необходимо учитывать «угрозу ветров» в данной местности. На основании данных ежедневного определения направления ветра, проводимого в течение года, определить целесообразное взаимное расположение промышленной жилой зоны (кодирование направлений ветра задано в задаче 4 уровня I).

2. Известно, что в городе ХХХХ самыми теплыми являются дни с 15 июля по 15 августа. Для проведения фестиваля были выбраны 7 следующих подряд дней, наиболее теплых по данным за последние 10 лет. Составить программу для выполнения этой работы на ЭВМ.

3. Составить результирующую таблицу первенства по футболу, в котором участвуют 10 команд. В качестве исходной информации задан счет: количество забитых (пропущенных) мячей в каждой проведенной игре. Для получения итогового результата необходимо по заданной таблице забитых (пропущенных) мячей составить таблицу очков (выигрыш - 2, ничья - 1, проигрыш - 0). Далее определить сумму очков для каждой команды и в соответствии с этим распределить команды по местам. Если сумма очков у двух команд одинакова, то сравниваются разности забитых и пропущенных мячей.

4. Для формирования сборной страны по хоккею предварительно выбрано 30 игроков. На основании протоколов игр (всего 10 игр) составлена таблица, в которой содержится штрафное время каждого игрока по каждой игре (штрафное время может составлять 2,5 или 10 мин).

5. Составить программу, которая составляет предварительный список кандидатов в сборную и определяет для каждого из них суммарное штрафное время. Игроки, оштрафованные хотя бы один раз на 10 мин, из кандидатов в сборную исключаются.

6. В задаче 3 определить, в скольких играх разность забитых и пропущенных мячей была большей или равной 3. Какая команда имеет наибольшее количество побед с таким крупным счетом?

7. Составить программу для ведения протокола баскетбольной игры. Во время игры машина ведет учет набранных очков и фолов каждого игрока. Игрок, получивший 5 фолов, удаляется из игры (эта информация должна появляться на экране). В конце игры должна выводиться информация о сумме очков, набранных каждым игроком, в порядке убывания.

8. Составить программу для обработки результатов кросса на 500 м для женщин. В кроссе участвует не более 100 студенток. Для каждой участницы ввести фамилию, шифр группы, фамилию преподавателя, результат.

Получить результирующую таблицу, упорядоченную по результатам, в которой содержится также информация о выполнении нормы ГТО. Определить суммарное количество студентов, выполнивших норму ГТО.

9. Для шифрования используется смешанный алфавит, полученный случайной перестановкой букв исходного алфавита. Например:

АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЬЪЮЯ
ШЗУКЛЖЦАЯЭЮФЪДВБПГЧЬБЕОИМРХНЦСТЙ

10. Для усложнения шифра вместо букв используются целые числа (А соответствует 1, Б — 2 и т. д.), и для кодирования каждой следующей буквы сообщения используют новый смешанный алфавит, полученный из исходного циклической перестановкой букв вправо на заданное число позиций. Число позиций, на которые осуществляется сдвиг, и частота их повторения известны заранее. Пусть, например, это числа 5, 3, 1, 8. Тогда для кодирования первой буквы будет использован алфавит, полученный из исходного циклическим сдвигом на 5 позиций вправо, второй буквы - на 3 позиции, третьей на 1, четвертой - на 8, пятой — снова на 5 и т. д. Составить программу, которая подготавливает таблицу для шифрования, содержащую четыре необходимых алфавита. Считать, что исходный алфавит задан в числовом виде.

11. Результаты соревнований фигуристов по одному из видов многоборья представлены оценками судей в баллах (от 0 до 6). По результатам оценки судьи определяется место каждого участника у этого судьи. Места участников определяются далее по сумме мест, которые каждый участник занял у всех судей. Составить программу, определяющую по исходной таблице оценок фамилии и сумму мест участников в порядке занятых ими мест. Число участников не более 15, число судей не более 10.

12. Пластика размером $30 \times 50 \text{ см}^2$ разделена на квадраты размером $2 \times 2 \text{ см}^2$. Произведено измерение температуры каждого квадрата. Используя эту информацию, построить изотермы при помощи символов *, +, ©, P, %, # разделив интервал изменения температуры на 6 равных интервалов и поставив в соответствие каждому подинтервалу один из указанных символов.

13. Японская радиоккомпания провела опрос 250 радиослушателей по вопросу: «Какое животное Вы связываете с Японией и японцами?». Составить программу получения пяти наиболее часто встречающихся ответов и их долей (в %).

14. Траектории движения двух самолетов заданы уравнениями

$$x_i = x_i(t), y_i = y_i(t), z_i = z_i(t), \quad i = 1, 2.$$

Определить, в какой момент времени расстояние между самолетами будет минимальным. Рассмотреть интервал времени между двумя заданными моментами. Время определить с точностью $1/50$ исходного интервала.

15. Сформировать из матрицы $A(10, 10)$ матрицу $B(10, 10)$ по следующим правилам:

а) элементы матриц A и B принимают только значения 0 или 1;

б) соседями элемента a_{ij} считаются все элементы, расположенные рядом с данным по горизонтали, вертикали или диагонали:

в) если сумма S значений соседей элемента a_{ij} меньше двух или больше трех, то $b_{ij} = 0$

г) если сумма S значений соседей элемента a_{ij} равна двум, то $a_{ij} = b_{ij}$.

д) если сумма S значений соседей элемента a_{ij} равна трем, то $b_{ij} = 1$.

По окончании формирования матрицы B значения элементов построчно вывести на печать, заменяя 0 - символом i , 1 - символом $*$.

Это преобразование является основой так называемой игры «Жизнь» (см., например, Узерелл Ч. Этюды для программистов, - М. Мир, 1982), позволяющей из исходного получить новый вариант расположения в матрице нулей и единиц (или при выводе в символьном виде - конфигурацию звездочек).

16. Составить программу - тест «Решительны ли Вы?».

На экране по очереди появляются 12 вопросов. На каждый из них нужно ввести ответ (да или нет). В зависимости от вопроса и ответа к общей сумме прибавляется определенное число очков. После окончания опроса по общей сумме очков делается заключение по поводу решительности опрашиваемого. В программе предусмотреть защиту от неправильного ответа.

Исходные данные для программы:

- 1) Сможете ли Вы на старом месте работы приспособиться к новым правилам?
- 2) Быстро ли адаптируетесь в новом коллективе?
- 3) Способны ли высказать свое мнение публично?
- 4) Согласитесь ли Вы без колебаний перейти в другое учреждение на должность с более высоким окладом?
- 5) Склонны ли отрицать свою вину в допущенной ошибке и искать оправдания?
- 6) Объясняете ли Вы обычно причины своего отказа от чего-либо истинными мотивами или стараетесь их завуалировать?
- 7) Сможете ли изменить свой прежний взгляд или убеждения в результате серьезной дискуссии?
- 8) Если Вам не нравится стиль изложения работы, с которой Вы знакомитесь, будете ли Вы править текст и настойчиво предлагать изменить его?
- 9) Купите ли Вы вещь, которая Вам очень нравится, но не так уж необходима?
- 10) Можете ли изменить свое решение под влиянием уговоров обаятельного человека?
- 11) Планируете ли заранее свой отпуск?
- 12) Всегда ли выполняете данное Вами обещание?

Таблица для определения набранных очков

Вопрос	Да	Нет	Вопрос	Да	Нет
1	3	0	7	3	0
2	4	0	8	2	0
3	3	0	9	0	2
4	2	0	10	0	3
5	0	4	11	1	0
6	2	0	12	3	0

Определение результата тестирования:

- От 0 до 9 очков — Вы очень нерешительны. Старайтесь изменить свой характер.
- От 10 до 18 очков — Вы принимаете решения осторожно, но не пасуете перед серьезными проблемами. Больше полагайтесь на свой опыт.
- От 19 до 28 очков — Вы достаточно решительны. Принятые решения отстаиваете до конца. Старайтесь быть объективным. Консультируйтесь по вопросам, в которых недостаточно компетентны.
- От 29 очков и выше - Вы принимаете решения единолично, не считаясь с мнением других. Это подавляет инициативу подчинённых, наносит ущерб психологическому климату коллектива. Нужно срочно менять стиль работы.

17. Составить программу для контроля знаний по курсу «Химия», раздел: «Номенклатура неорганических веществ».

В памяти ЭВМ хранятся вопросы и правильные ответы. Вопросы высвечиваются на экране по очереди, и на каждый вопрос обучаемый вводит ответ, который сравнивается с правильным.

После ответа на все вопросы предусмотреть выставление оценки.

Указания к решению задач II уровня.

- Задача сводится к определению того, какое значение чаще всего встречается в массиве направлений ветра. Для ее решения нужно сформировать массив размером 8 (индекс совпадает с кодом направления), каждый элемент которого является количеством элементов массива направлений ветра, имеющих заданное значение, а затем определить максимальный элемент этого массива и соответствующий ему индекс.
- Исходные данные целесообразно представить в виде двумерного массива 10x32. По исходному массиву нужно далее сформировать массив 10x26, каждый элемент которого — сумма температур за каждые 7 из 32 дней. Далее нужно просуммировать столбцы этого массива и определять, для какого столбца сумма элементов является максимальной.
- Предусмотреть числовой массив размером 10x10 для количества забитых (пропущенных) мячей по каждой игре и символьный массив размером 10 для названий команд. Количество мячей, забитых i -й командой j -й команде,

следует помещать в j -й столбец i -й строки. Для формирования результирующей таблицы суммировать строки массива очков. Полученный в результате одномерный массив нужно упорядочить по убыванию, меняя при этом и названия команд. Полученный упорядоченный массив нужно проверить на совпадение его элементов. В случае равенства очков по названию команд следует отыскать соответствующие строки в матрице забитых (пропущенных) мячей и найти разности сумм забитых и пропущенных мячей. (Для получения суммы мячей, забитых i -й командой, необходимо вычислить сумму элементов i -й строки, а пропущенных i -й командой - сумму элементов i -го столбца.) В случае необходимости соседние команды поменять местами. Вычисленную худшую разность сумм забитых и пропущенных мячей нужно сохранять на тот случай, если следующая сумма очков совпадает с двумя предыдущими.

4. Исходную информацию целесообразно хранить в виде двумерного массива 30×10 . Предусмотреть также массив размером 30 для фамилий игроков. Для решения задачи нужно суммировать строки массива, сравнивая при этом значение каждого слагаемого с 10. Если найдено такое слагаемое, то соответствующая фамилия в результирующий список не включается и суммирование строки дальше не производится.

5. См. указание к задаче 3. Для решения задачи можно изменять индексы только одного треугольника матрицы (например, лежащего выше главной диагонали), а при вычислении разности мячей вычитаемое должно извлекаться из симметричного элемента другого треугольника (для этого индексы нужно поменять местами). Если разность мячей по модулю больше или равна 3, то нужно увеличить результат (количество игр с крупным счетом) на 1. Для решения второй части задачи предусмотреть массив размером 10. Для игры с крупным счетом надо увеличивать на 1 элемент этого массива с индексом, равным номеру строки исходного массива, если разность мячей положительна, или номеру столбца, если разность мячей отрицательна.

6. Нужно предусмотреть три массива для фамилий игроков, числа фолов и очков каждого игрока. При окончании ввода данных (окончании игры) предусмотреть ввод специального значения (см. работу 3). При увеличении числа фолов оно каждый раз должно сравниваться с 5. После окончания игры нужно упорядочить массив очков (меняя при этом местами и фамилии игроков).

7. Использовать алгоритмы, изложенные на лекциях.

8. Исходный алфавит задан в виде массива размером 32, элементами которого являются целые числа от 1 до 32 в произвольном порядке. Результат получить в виде двумерного массива размером 4×32 , в котором каждая строка получена циклической перестановкой исходного массива на заданное количество позиций.

9. Таблицу оценок (массив размером не менее, чем 10×15) преобразовать в таблицу мест. Просуммировать элементы ее столбцов. Полученный массив

упорядочить по возрастанию, переставляя одновременно элементы массива фамилий участников (размером не менее чем 15).

10. Исходная информация (температура каждого квадрата 2X2) может быть представлена в виде двумерного массива размером 15x25. Этот двумерный числовой массив нужно преобразовать в символьный массив такого же размера и напечатать его. Для выполнения этого преобразования нужно:

- найти значения минимального и максимального элементов исходного массива для определения интервала изменения температур;
- разделить этот интервал на 6 для определения шести подинтервалов;
- сформировать символьный массив, просматривая элементы исходного массива и определяя, какому из шести подинтервалов принадлежит элемент этого массива;
- вывести на печать полученный символьный массив.

11. Исходную информацию нужно разместить в символьном массиве размером не менее 250. Далее сформировать два массива - символьный, в который нужно поместить все отличающиеся друг от друга ответы, и числовой, в котором подсчитывается частота их появления в исходном массиве. Упорядочить массив частот (переставляя при этом и элементы массива ответов). Определить доли, первых пяти ответов в % и напечатать их вместе с соответствующими ответами.

Замечание. Если для упорядочения используется алгоритм, изложенный в п. 8, то упорядочение можно прекратить, как только будут сформированы первые пять элементов упорядоченного массива.

12. Расстояние L между самолетами в каждый момент времени вычисляется

как $L = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$. Для решения задачи нужно сформировать массив размером 50, вычисляя расстояние между самолетами с шагом по времени в 1/50 исходного интервала. Далее нужно найти минимальный элемент этого массива. Функции $x_i(t), y_i(t), z_i(t), i = 1, 2$, задать самостоятельно. Для их определения в программе использовать оператор DEF FN.

13. Для решения задачи нужно, предварительно заполнив нулями матрицу В, просматривать элементы матрицы А и для каждого элемента определять сумму элементов, являющихся его соседями. В зависимости от результата присваивать элементу массива В соответствующее значение (значение 0 присваивать не нужно). Перед выводом на печать преобразовать матрицу В к символьному виду. Чтобы просмотр элементов матрицы А и суммирование ее соседей не зависел от того, находится элемент на границе или со всех сторон имеет соседей, можно работать с расширенной матрицей 12x12. Элементам, расширяющим матрицу ($i = 1, 12; j = 1, 12$), значений не присваивать, они всегда имеют значения 0.

14. Для исходных данных предусмотреть символьный массив вопросов, два числовых массива очков, начисляемых в случае ответов «да» и «нет» а также символьный массив размером 4 с текстами, выводимыми на экран в зависимости от набранной суммы очков.

15. Вопросы (например, «Название вещества $\text{Ca}(\text{HCO}_3)_2$?») и правильные ответы на них (например, «Гидрокарбонат кальция») помещаются в два символьных массива. В третий символьный массив можно поместить комментарий, который будет высвечиваться в случае неправильного ответа (например, «Повторите раздел «Номенклатура солей»). Причем, поскольку слова «Название вещества» в каждом вопросе повторяются, их можно в явном виде указать в операторе вывода, а в массив вопросов поместить только название вещества.

Вопросы (из первого массива) высвечиваются по очереди. Обучаемый вводит ответ в символьную переменную. Этот ответ сравнивается с правильным ответом (соответствующим элементом второго массива). Результат сравнения можно высветить на экране вместе с комментарием.

В программе предусмотреть определение количества правильных ответов и выставление оценки после ответа на все вопросы. По количеству вопросов N и количеству правильных ответов P оценку можно вычислить следующим образом:

$$M = \text{INT}\left(\frac{P}{N} * 5 + 0.9\right)$$

и, если $M < 3$, положить $M = 2$.

Исходные данные для работы программы (массивы вопросов, ответов и комментарии) целесообразно разместить в программе (в операторах DATA: в одном операторе DATA - вопрос, ответ и комментарий).

Задание III уровня.

Решение задач III уровням требует использования подпрограмм. Требуется также выполнить все пункты задания II уровня. (подпрограмма-это отдельная функция в C++).

Варианты задач III уровня.

1. Составить программу для обработки результатов соревнований по фигурному катанию, проводившихся по трем видам многоборья (обязательная, короткая и произвольная программы).

Обработку результатов по каждому из видов (см. задачу 9 уровня II) осуществлять в подпрограмме.

2. Японская радиокompания провела опрос 250 радиослушателей по трем вопросам:

- 1). Какое животное Вы связываете с Японией и японцами?
- 2). Какая черта характера присуща японцам больше всего?
- 3). Какой неодушевленный предмет или понятие Вы связываете с Японией?

Большинство опрошенных прислали ответы на все или часть вопросов. Составить программу получения первых пяти наиболее часто встречающихся ответов по каждому вопросу и доли (в %) каждого такого ответа. Предусмотреть необходимость сжатия столбца ответов в случае отсутствия

ответов на некоторые вопросы. Обработку информации по каждому вопросу осуществлять в подпрограмме.

3. Соревнования по футболу между командами проводятся два этапа. Для проведения первого этапа участники разбиваются на две группы по 12 команд. Для проведения второго этапа выбирается 6 лучших команд каждой группы по результатам первого этапа. Составить список команд участников второго этапа. Составить список команд участников второго этапа.

Определение результатов соревнования внутри одной группы осуществлять в подпрограмме. Задаваемые исходные данные и правила определения результатов соревнований приведены в задаче 3 уровня II.

4. Чемпионат города по футболу проводится по двум группам: сильная — 12 команд и слабая — 12 команд. Замещение мест в сильной группе проводится по результатам переходного турнира. В турнире участвуют три худшие команды из сильной группы и три лучшие — из слабой. Составить список команд — участниц переходного турнира. Обработку результатов соревнований в одной группе (см. задачу 3 уровня II) осуществлять в подпрограмме.

5. В соревнованиях участвуют три команды по 6 человек. Результаты соревнований в виде мест участников каждой команды (1-18) размещены в трех массивах, содержащих по 6 компонент.

Определить команду - победителя, вычислив количество баллов, набранное каждой командой. Участнику, занявшему 1-е место начисляется 5 баллов, 2-е - 4, 3-е - 3, 4-е - 2, 5-е - 1, остальным — 0 баллов. Определение числа баллов, набранных одной командой, осуществлять в подпрограмме.

6. В памяти ЭВМ хранится информация о засеваемых площадях и урожае зерновых по районам для 10 областей (в каждой области не более 10 районов). Определить среднюю урожайность по каждой области и область, добившуюся наибольшей урожайности. Определение средней урожайности одной области выполнять в подпрограмме.

7. Информация о результатах сессии (5 экзаменов) по каждой группе хранится в памяти ЭВМ. Для подведения итогов социалистического соревнования определить средний балл для пяти групп одного потока студентов и выдать список групп в порядке убывания среднего балла. Определение среднего балла одной группы осуществлять в подпрограмме.

8. Соревнования (лыжные тонки) проводятся двумя группами по 10 человек. Результаты соревнований представлены списками участников и их результатов по каждой группе. Предварительное подведение итогов проводится по каждой группе, результатом его являются списки участников по группам в порядке занятых ими мест. Необходимо получить общий список, в котором участники расположены в порядке, соответствующем показанным результатам. Определение результатов по каждой группе осуществлять в подпрограмме.

9. Написать программу построения латинского квадрата. Латинский квадрат - это матрица $N*N$, элементы которой выбраны от 1 до N так, что каждое

число встречается только один раз в каждой строке и в каждом столбце. Циклический сдвиг (см. указание к задаче) осуществляется в подпрограмме.

10. В задаче 13 уровня II исходную конфигурацию задать как десять символьных строк, состоящих из символов `_` и `*`. Все преобразования осуществлять с числовой матрицей. Проследить за развитием первоначальной конфигурации в нескольких поколениях. Каждую новую конфигурацию и номер ее выводить на экран. Предусмотреть возможность прекращения работы программы по желанию пользователя. Преобразование числового представления в символьное и вывод на печать очередной конфигурации осуществлять в подпрограмме.

11. Имеется n сосудов, в каждом из которых лежит по одному камню белого, синего или красного цвета. «Заглядывая» в каждый сосуд по одному разу, расположить их в таком порядке, чтобы в первой группе сосудов лежали только красные, затем - синие, далее - белые камин (этот порядок цветов соответствует голландскому национальному флагу).

12. В задаче 15 уровня II предусмотреть возможность выбора случайным образом части вопросов, используемых для контроля.

Для выставления оценки считать, что вопросы имеют разные веса по важности. Формирование номеров вопросов для контроля одного обучаемого осуществлять в подпрограмме.

13. Составить программу для контроля знаний. В программе задается один вопрос, ответ на который включает несколько наименований (например, «Назовите все элементы периодической системы, представляющие группу галогенов» или «Назовите все города с населением свыше 1 млн.» и т. п.). В памяти ЭВМ хранится список наименований, являющийся полным ответом на вопрос. Введенный ответ необходимо сравнить с правильным ответом.

14. В памяти ЭВМ хранится список фамилий абонентов в алфавитном порядке и номеров их телефонов. Составить программу, обеспечивающую быстрый поиск фамилии абонента по номеру телефона.

15. В памяти ЭВМ хранятся списки номеров телефонов и фамилий абонентов, упорядоченные по номерам телефонов, для каждого из пяти телефонных узлов города. Один телефонный узел включает несколько АТС (не более 10). Номера АТС (первые две цифры номера телефона), относящихся к каждому телефонному узлу, также хранятся в памяти ЭВМ. Составить программу, обеспечивающую быстрый поиск фамилии абонента по заданному номеру телефона.

Указания к решению задач III уровня.

1. Результаты соревнований по каждому виду в виде числа участников (некоторые из участников могут выбыть), их фамилии и мест судей вводить в память по очереди (можно в один и тот массив). Результаты обработки по каждому виду хранить в памяти, для получения окончательного результата. Окончательный результат получать только для участвовавших во всех трех видах. Под программу составить в общем виде для произвольного числа

участников и судей (в пределах задаваемых ограничений). См. та указание к задаче 9 уровня II.

2. Исходную информацию задать в виде двумерного массива размером 250×3 . Перед обращением к подпрограмме, обрабатывающей ответы на один вопрос, переслать столбец исходного массива в одномерный массив, используемый в подпрограмме (пустые ответы не пересылать). В результате пересылки нужно определить также количество непустых ответов, чтобы использовать его в подпрограмме и печатать в качестве итоговой информации.

5. Исходную информацию задать в виде двумерного массива: размером 3×6 и массива названий команд. При обработке результатов одной команды использовать то соображение, что сумма номера места, если оно ≤ 6 , и количества очков равна 6.

8. Для решения задачи нужно упорядочить два массива результатов и объединить их в один так же упорядоченный массив. Одновременно нужно объединить и массивы фамилий (в соответствии с результатами).

9. Первую строку матрицы заполнить цифрами от 1 до N, каждую следующую строку получать, циклически сдвигая предыдущую на 1 элемент вправо (влево). Очередную строку матрицы получать в виде одномерного массива в подпрограмме, используя массив, сформированный при предыдущем обращении, как исходный при выполнении следующего.

10. Исходную конфигурацию задать как символьный массив размером 10×10 . В программе нужно предусмотреть переменную, в которую перед получением очередной конфигурации можно вводить специальное значение для прекращения работы программы (см. теоретическое введение к работе 3).

11. Цвет камней можно представить либо символами, например, К, В, С, либо закодировать их числами, например, 1 - красный, 2 - белый, 3 - синий. Просматривая элементы массива (сосуды), начиная с первого, накапливать количество встретившихся синих камней (например, в переменной С) и белых (например, в переменной В) и сразу ставить их в нужном порядке. Для этого, если в 1-м сосуде - С, то его нужно поменять местами с первым из В (элемент с индексом (1-В)). Если в 1-м сосуде - К, то его нужно поменять местами сначала с первым элемент с индексом (1 - В)), а затем с первым из С (т. е. поменять местами (1-В)-й и (1-В-С)-й элементы). Перестановку элементов осуществлять в подпрограмме.

12. (См. указание к задаче 15 уровня II.) Для выбора случайным образом заданного числа (К) вопросов можно использовать алгоритм получения К различных случайных целых чисел из интервала [А; В], где $A=1$, $B=N$, N - общее число вопросов (см. теоретическое введение к работе 12, программа 5.11).

Номера вопросов, которые будут предложены одному обучаемому, из общего списка вопросов целесообразно поместить в массив, например, D.

Веса вопросов хранить в массиве, например, V . Для получения оценки суммировать веса правильных ответов ($P = P + V(I)$, где $I = D(J)$ - номер вопроса из общего списка, J - порядковый номер задаваемого вопроса. Далее нормировать эту сумму, разделив ее на общий вес всех правильных ответов $P = P / \sum_{j=1}^K V(D(j))$. Оценку можно вычислять, например, по формуле $M = INT(P * 5 + 0.9)$, и если $M < 3$, то положить $M = 2$.

13. Список наименований, являющийся полным ответом на вопрос, хранить в символьном массиве. Ответы обучаемого вводить в другой символьный массив. Для повышения эффективности программы можно правильные ответы ввести в массив упорядоченными (или упорядочить сразу после ввода), а после ввода ответов, перед сравнением, упорядочить и массив ответов. Для сравнения использовать алгоритм сравнения двух упорядоченных массивов.

Алгоритмы упорядочения массивов и сравнения двух упорядоченных массивов оформить в виде подпрограмм.

14. Переформировать исходные списки, упорядочив их по номерам телефонов. Для поиска нужной фамилии использовать алгоритм бинарного поиска в массиве номеров телефонов.

Алгоритм упорядочения оформить в виде подпрограммы.

15. Список номеров телефонов хранить в двумерном массиве (например, T), содержащем пять строк (одна строка относится к одному узлу). Список фамилий - в двумерном символьном массиве (например, $F@$) того же размера. Список АТС, сгруппированный по телефонным узлам, хранить в одномерном массиве, например A). Индексы начала списка по каждому узлу хранить в отдельном массиве, например, C .

Решение задачи включает следующие шаги:

1) Выделить первые две цифры в заданном номере телефона (см. теоретическое введение к работе II), т.е. определить номер АТС.

2) Определить индекс элемента массива - A , содержащего этот номер.

3) Определить, к какому узлу относится АТС. Для этого определить, к какому из интервалов изменения индексов в массиве - A (номер интервала - номер узла) относится найденный индекс. (Использовать массив C .)

4) В строке массива - T , относящейся к этому узлу, найти элемент, совпадающий с заданным номером телефона, используя алгоритм бинарного поиска.

5) Напечатать соответствующий элемент массива $F@$.

Вопросы, ответы на которые помогут подготовить теоретическую часть.

1. Какими способами можно задать исходные данные для программы? Операторы ввода. В каких случаях целесообразно их использование?
2. Отладка и тестирование программы. Средства отладки. Что такое тест?

3. Алгоритм определения числа и суммы элементов, удовлетворяющих заданному условию. Какие при этом используются типовые структуры алгоритмов?

4. Алгоритм объединения двух массивов одинакового размера; в один с чередованием элементов. Какова связь между индексами исходного массива и формируемого?

6. Алгоритм инвертирования массива. Что является управляющей переменной цикла и как она изменяется?

6. Алгоритм формирования массива из элементов другого массива; удовлетворяющих заданному условию. Индексы элементов какого массива является управляющей переменной цикла? Как определить количество элементов в сформированном массиве? Почему индекс для формируемого массива целесообразно получать перед пересылкой очередного элемента в этот массив?

7. Алгоритм циклического сдвига элементов массива:

а) с использованием одной вспомогательной переменной;

б) с использованием вспомогательного массива.

В чем достоинства и недостатки этих способов? Почему при циклическом сдвиге вправо перемещение элементов массива в «хвост» необходимо осуществлять с конца?

8. Алгоритм упорядочения массива. Почему при перестановке элементов массива не требуется дополнительная переменная? В каких пределах изменяется управляющая переменная внешнего и внутреннего циклов и почему? Какие изменения необходимо внести в программу, если при перестановке элементов упорядочиваемого массива нужно точно так же переставлять элементы другого массива?

9. Алгоритм проверки массива на упорядоченность. Что является результатом этого алгоритма? Как формально обеспечить завершение цикла, если условие упорядоченности для какой-либо пары элементов не выполняется?

10. Алгоритм поиска заданного элемента в упорядоченном массиве (бинарный поиск). В чем преимущество бинарного поиска?

11. Алгоритм объединения двух упорядоченных массивов в один так же упорядоченный. Какие типовые структуры алгоритмов здесь используются? Почему целесообразно использование циклов «Пока»?

12. Алгоритм определения количества совпадающих элементов двух упорядоченных массивов. Какие типовые структуры алгоритмов здесь используются? Какая переменная является управляющей переменной цикла? Как организовать завершение цикла, если один из массивов полностью исчерпан?

13. Как составить программу в общем виде для обработки массивов произвольного размера? Какие преимущества дает использование переменных для обозначения размеров массивов?

ЗАДАНИЯ К ТЕМЕ 1

Задание I уровня. Это задание требует использования одного из приведенных в лекциях алгоритмов для конкретного размера массива.

Для решения задачи составить программу. В программе предусмотреть ввод и печать исходных данных и печать полученного результата с поясняющим текстом. Использовать циклы для ввода, вывода и обработки массивов.

Привести список использованных переменных. Подготовить тесты.

Проверить работу программы на ЭВМ.

Варианты задач I уровня.

1. Найти сумму элементов одномерного массива размером 4. Разделить каждый элемент исходного массива на полученное значение. Результат получить в том же массиве. Напечатать в одной строке.
2. Вычислить сумму и разность двух заданных одномерных массивов размером 5. Результат напечатать в виде двух параллельных столбцов.
3. Просуммировать элементы строк матрицы размером 4x3. Результат получить в одномерном массиве размером 4.
4. Задан массив X размером 5. Вычислить значения функции $y=0,5 * \ln x$ при значениях аргумента, заданных в массиве X, и поместить их в массив Y. Напечатать результат (массивы X и Y) в виде двух столбцов.
5. Найти среднее значение элементов заданного массива размером 5. Преобразовать исходный массив, вычитая из каждого элемента среднее значение.
6. Решить уравнение $ax = b$ для пяти пар значений a и b, заданных в виде двух массивов. Результат поместить в массив X.
7. Вычислить скалярное произведение двух векторов (X, Y) размером 4. (Скалярное произведение вычисляется по формуле

$$S = \sum_{i=1}^4 x_i y_i .)$$

8. Вычислить длину вектора X размером 4. (Длина вектора

вычисляется по формуле $L = \sqrt{x_1^2 + \dots + x_4^2} .)$

9. Вычислить сумму двух заданных матриц размером 3x3.
10. Найти сумму всех элементов матрицы размером 4x3.
11. Просуммировать элементы столбцов заданной матрицы размером 4x3. Результат получить в одномерном массиве размером 3.
12. Определить среднее значение элементов массива. Найти далее индекс элемента массива, наиболее близкого к среднему значению.
13. Задан массив размером 10. Сформировать два массива размером 5, включая в первый элементы исходного массива с четными индексами, а во второй — с нечетными.
14. Заданы матрица размером 5x5 и число K. Разделить элементы K-й строки на диагональный элемент, расположенный в этой строке.

15. Заданы матрица A размером 4×4 и числа K и L ($K \neq L$, $1 \leq K, L \leq 4$). Из L -й строки вычесть K -ю, умноженную на $\alpha LK / \alpha K K$.

Указания к решению задач I уровня.

4. При выводе результатов оператором PRINTF в каждую строку выводить пару $X(I), Y(I)$ в зонном формате.

5. Для вычисления среднего значения найти сумму элементов и разделить на число элементов.

6. Ввести массивы A и B . В цикле по I вычислять $X(I) = B(I)/A(I)$ при $I = 1, \dots, 5$.

13. В цикле выполнять $P(I) = X(2 * I), T(I) = X(2 * I - 1)$ при $I = 1, \dots, 5$.

Задание II уровня. Требуется применения алгоритмов, приведенных в лекциях, и их сочетаний. Программу необходимо составить в общем виде так, чтобы она могла быть использована для обработки массивов различных размеров в пределах заданных ограничений на размеры (для одномерных не более 100, для двумерных не более 10 по каждому измерению) (см. пример 2).

В программе предусмотреть ввод и печать (выполнить в виде отдельных функций) исходных данных и печать полученных результатов с необходимыми поясняющими текстами. Предусмотреть контроль ввода.

Привести список использованных переменных. Подготовить тесты.

Проверить работу программы на ЭВМ.

Варианты задач II уровня.

1. Для заданной квадратной матрицы сформировать одномерный массив из ее диагональных элементов. Найти след матрицы, суммируя элементы одномерного массива.

Преобразовать исходную матрицу по правилу: четные строки разделить на полученное значение, нечетные оставить без изменения. Преобразованную матрицу напечатать по строкам.

2. Задана Прямоугольная матрица. Получить транспонированную матрицу и напечатать ее по строкам.

3. Заданы матрица и вектор. Получить их произведение. Напечатать в строку.

4. Заданы два одномерных массива различных размеров. Объединить их в один массив, включив второй массив между K -м и $(K+1)$ -м элементами первого (K задано).

5. Заданы две матрицы A и B размером $N \times N$. Сформировать из них прямоугольную матрицу X размером $N \times 2N$, включая в первые N столбцов матрицу A , в следующие — матрицу B .

6. Задан массив A размером $N \times M$ и вектор B размером M . Элементы первого столбца массива A упорядочены по убыванию. Включить массив B в качестве новой строки в массив A с сохранением упорядоченности по элементам первого столбца.

7. Матрица размещена в одномерном массиве по строкам. Удалить K -ю строку матрицы (K задано) из одномерного массива. Результат напечатать по строкам.

8. В задаче 7 удалить K -й столбец. Результат напечатать по строкам.

9. В задаче 7 поменять местами K-ю и L-ю строки. Результат вывести по строкам.
 10. В задаче 7 поменять местами K-й и L-й столбцы. Результат вывести по строкам.
 11. Из заданной матрицы удалить K-ю строку и L-й столбец.
 12. В заданной матрице заменить K-ю строку и L-й столбец нулями, кроме элемента, расположенного на их пересечении.
 13. В одномерном массиве размещены: в первых N элементах значения аргумента в порядке возрастания, в следующих — соответствующие им значения функции, и задана пара чисел — значения аргумента и функции. Поместить их в массив с сохранением упорядоченности по значениям аргумента. Напечатать полученный массив в виде двух параллельных столбцов (аргумент и функция).
 14. Заданную квадратную матрицу преобразовать, используя умножение строки на число и сложение строк, таким образом, чтобы все элементы первого столбца обратились в нуль, кроме элемента, расположенного на главной диагонали (см. п. 18 введения к работе).
 15. Заданную квадратную матрицу преобразовать, используя умножение столбца на число и сложение столбцов, таким образом, чтобы все элементы первой строки обратились в нуль, кроме элемента, расположенного на главной диагонали (см. п. 18 введения к работе).
- Указания к решению задач II уровня.**
1. Для формирования одномерного массива X из диагональных элементов матрицы A размером N×N необходимо выполнять $X(I)=A(I, I)$ при $I=1, \dots, N$.
 4. «Хвост» первого массива (начиная с (K+1)-го элемента) переместить вправо на число позиций, равное размеру второго массива. Далее заполнить первый массив, начиная с (K+1)-го элемента, элементами второго массива.
 5. Внутренний цикл (по номеру столбца J) целесообразно выполнять от 1 до N, присваивая для каждого J значения двум элементам строки: $X(I, J)=A(I, J)$, $X(I, N+J)=B(I, J)$.
 6. Место включения новой строки определяется местом включения в упорядоченный первый столбец первого элемента новой строки. Новую строку нужно включить перед строкой (I), для которой выполняется условие $A(I, 1) < B(1)$ (см. п. 13 теоретического введения).
 7. K-я строка матрицы расположена в одномерном массиве в элементах с $((K-1)*M+1)$ -го до $(K*M)$ -го.
 8. K-й столбец матрицы расположен в элементах с индексами $(I-1)*M+K$, $I=1, 2, \dots, N$.
 9. См. указание к задаче 7.
 10. См. указание к задаче 8.
 12. Перед заполнением нулями K-й строки и L-го столбца целесообразно элемент, лежащий на их пересечении, переслать во вспомогательную переменную, а после заполнения нулями восстановить его значение.
 13. Значение аргумента включить в первую половину массива с сохранением упорядоченности. Значение функции включить на соответствующее место во

вторую половину массива перед $(N+I+1)$ -м элементом, где I — индекс элемента, в который помещено значение аргумента, N — исходное число значений аргумента и функции.

14. Из каждой строки, начиная со второй, необходимо вычитать элементы первой строки, умноженные на коэффициент $P = A(K,1)/A(1,1)$, т.е. в цикле по номеру строки K ($K=2, \dots, N$) организовать цикл по номеру столбца J ($J=1, \dots, N$), в котором выполнять $A(K, J) = A(K, J) - A(1, J) * P$. (Предполагается, что $A(1, 1) \neq 0$.)

15. Из каждого J -го столбца, начиная со второго, необходимо вычитать элементы первого умноженного на $P = A(1, J)/A(1, 1)$, т.е. в цикле по номеру столбца J ($J = 2, \dots, N$) организовать цикл по номеру строки I ($I=1, \dots, N$), в котором выполнять $A(I, J) = A(I, J) - A(1, J) * P$.

Задание III уровня. Требуется использования типовых алгоритмов (см. лекции), а также некоторого творческого подхода. Печать массивов осуществлять по заданному формату, Необходимо также выполнить все пункты задания II уровня.

Варианты задач III уровня.

1. Задан массив X размером N . Сформировать из него матрицу A , содержащую по L элементов в строке. Недостающие элементы в последней строке (если такие будут) заполнить нулями. Напечатать матрицу по строкам.
2. Вычислить значения функции $y = \cos x + x \sin x$ в n точках отрезка $[a, b]$. Вычисляемые значения помещать в одномерный массив параметрами x_i, y_i ($i=1, \dots, n$). Напечатать полученный массив в два столбца (аргумент и функция), используя для аргумента вывод по формату с фиксированной точкой, а для функции — по формату с плавающей точкой (с порядком).
3. Задана матрица A размером $N \times N$. Сформировать два одномерных массива. В один переслать по строкам верхний треугольник матрицы, включая элементы главной диагонали, в другой — нижний треугольник. Распечатать верхний и нижний треугольники по строкам.
4. Квадратная матрица задана в виде одномерного массива по строкам. Напечатать верхний треугольник матрицы (включая элементы главной диагонали) по строкам.
5. Матрица, симметричная относительно главной диагонали, задана верхним треугольником в виде одномерного массива по строкам. Восстановить исходную квадратную матрицу и напечатать по строкам.
6. Задана квадратная матрица. Переставить строку с максимальным элементом на главной диагонали со строкой с заданным номером.
7. Задана квадратная матрица. Исключить из нее строку и столбец, на пересечении которых расположен максимальный элемент главной диагонали.
8. Заданы матрица (размером $N \times N$) и число K ($1 \leq K \leq N$). Строку с максимальным по модулю элементом в K -м столбце переставить с K -й строкой.

9. Заданы матрица (размером $N \times N$) и число K ($1 \leq K \leq N$). Столбец с максимальным по модулю элементом в K -й строке переставить с K -м столбцом.

10. Задана матрица размером $N \times N$. Найти максимальный по модулю элемент матрицы. Переставить строки и столбцы матрицы таким образом, чтобы максимальный по модулю элемент был расположен на пересечении K -й строки и K -го столбца.

11. Используя преобразования, описанные в лекции 3, привести заданную квадратную матрицу к такому виду, чтобы все элементы ниже главной диагонали были нулевыми.

12. Используя преобразования, описанные в лекции 3, привести заданную квадратную матрицу к такому виду, чтобы все элементы выше главной диагонали были нулевыми.

13. Используя преобразования, описанные в лекции 3, привести заданную квадратную матрицу к такому виду, чтобы элементы ниже и выше главной диагонали были нулевыми.

14. Найти произведение двух заданных матриц (лк 2).

15. Задан одномерный массив. Преобразовать его таким образом, чтобы все его элементы принадлежали отрезку $[-1; 1]$. Вывести на печать параметры преобразования и полученный массив.

В программе предусмотреть возможность обратного преобразования.

Указания к решению задач III уровня.

1. Необходимо определить число полных строк матрицы $K = \text{INT}(N/L)$ и число недостающих элементов в последней строке $L1 = N - K * L$. Заполнять матрицу по строкам. (Соотношение между индексами одномерного и двумерного массивов см. в п. 20 введения к работе.)

Если $L1 \neq 0$, то отдельно заполнить последнюю строку (первые $L1$ элементов — последними элементами одномерного массива, последние $L - L1$ элементов — нулями).

2. Для каждого I помещать $X(I)$ в $(2 * I - 1)$ -й, $Y(I)$ — в $(2 * I)$ -й элементы одномерного массива.

3. Для решения первой части задачи нужно в цикле по номеру строки I организовать цикл по номеру столбца, изменяемому в пределах от I до N . Для определения соотношения между индексами одномерного и двумерного массивов следует иметь в виду, что длина пересылаемой части строки зависит от номера строки и составляет $L = N - I + 1$, т. е. элемент $A(I, J)$ пересылается в $((I - 1) * L + J)$ -й элемент одномерного массива.

Для второй части задачи решение аналогично, но цикл по J организуется в пределах от 1 до $I - 1$ и длина пересылаемой части строки на 1 меньше.

4. См. указание к задаче 3.

5. См. указание к задаче 3.

Верхний и нижний треугольники заполнять в одном цикле, помещая один и тот же элемент одномерного массива в два расположенных симметрично относительно диагонали элемента матрицы. Диагональ матрицы заполнить в отдельном цикле.

6. Диагональ переслать в одномерный массив. Найти максимальный элемент и его индекс M. Далее переставить M-ю строку со строкой с заданным номером.

7. См. указание к задаче 6.

8. K-й столбец переслать в одномерный массив. Найти в этом массиве номер элемента, являющегося максимальным по модулю. Если максимальным по модулю является K-й элемент, перестановку не ВЫПОЛНЯТЬ.

9. См. указание к задаче 6.

10. Найти положение максимального по модулю элемента матрицы и выполнить последовательно перестановку строк и перестановку столбцов.

11. Необходимо организовать цикл по номеру столбца и при каждом его прохождении обращать в 0 элементы, расположенные ниже главной диагонали (см. указание к задаче 14 уровня II).

Чтобы исключить возможность деления на 0 и повысить точность вычислений перед преобразованием очередного J-го столбца, нужно найти максимальный по модулю элемент в J-м столбце и перевести его на главную диагональ соответствующей перестановкой строк.

12. Необходимо организовать цикл по номеру строки и при каждом его прохождении обращать в 0 элементы, расположенные правее главной диагонали (см. указание к задаче 15 уровня II). Перед преобразованием очередной 1-й строки нужно найти максимальный по модулю элемент в этой строке и соответствующей перестановкой столбцов перевести его на главную диагональ.

13. Сначала заполнить нулями нижний треугольник матрицы (см. указание лекции 3), затем верхний.

14. Перед выполнением умножения проверить соответствие умножаемых матриц друг другу по размеру. Если число столбцов первой матрицы не равно числу строк второй, умножение не выполнять.

15. Для выполнения преобразования найти максимальный α_{\max} и минимальный α_{\min} элементы в массиве. Далее, каждый элемент преобразовать по формуле

$$\alpha_i = \alpha_i + \beta,$$

где α , β — параметры преобразования — вычисляются из соотношений

$$\alpha_{\max} + \beta = 1$$

$$\alpha_{\min} + \beta = -1$$

Вопросы для освещения в курсовой работе, соответственно заданиям.

1. Что такое массив? Как осуществляется доступ к элементам массива?

2. Для чего нужно описывать массивы? Как осуществляется описание массивов на Си?

3. Как осуществить ввод вектора, матрицы: а) по строкам; б) по столбцам?

4. Как организовать ввод двух массивов одинакового размера; разных размеров?

5) Как организовать вывод вектора: а) в строку; б) в столбец; матрицы — по строкам? Какие операторы(функции) надо использовать?

6. Как вывести два вектора одинакового размера в два параллельных столбца?
7. Типовые алгоритмы обработки массивов: суммирование элементов: а) вектора; б) матрицы; вычисление следа матрицы (суммирование диагональных элементов).
8. Алгоритм суммирования: а) векторов; б) матриц.
9. Алгоритм транспонирования матрицы с получением результата: а) в другом массиве; б) в том же массиве.
10. Алгоритм умножения: а) матрицы на матрицу; б) матрицы на вектор.
11. Алгоритм удаления: а) элемента из вектора; б) строки из матрицы.
12. Алгоритм включения элемента: а) в заданную позицию вектора; б) в упорядоченный массив с сохранением упорядоченности.
13. Как поменять местами: а) элементы в векторе; б) строки в матрице?
14. Алгоритм поиска минимального (максимального) элемента массива: а) вектора; б) матрицы.
15. Алгоритм преобразования: а) матрицы в вектор; б) вектора в матрицу.
16. В чем особенности организации программ для обработки массивов произвольных размеров?

**Методические указания и задания к выполнению
курсовой работы по курсу «Информатика»**

Рассмотрена и рекомендована к
тиражированию на ученом совете
ТУИТ от «24» марта 2011 (прот. №38)

Разрешена к изданию 24 апреля 2011 года

Составители:

доц. Идиатулина Э. М.

асс. Хасанова А. З.

асс. Ибрагимова К. А.

Ответственный редактор

доц. Джайлаов А. А.

Редактор

Корректор