

Федеральное государственное бюджетное образовательное учреждение высшего
профессионального образования
«Московский государственный университет имени М.В.Ломоносова»
Факультет фундаментальной физико-химической инженерии

УТВЕРЖДЕН

на заседании Ученого совета

« 14 » июня 2013 г.

протокол № 4

Заместитель декана по учебной работе

_____ / Григорьева Л.Д. /

« 14 » июня 2013 г.

УЧЕБНО-МЕТОДИЧЕСКИЙ КОМПЛЕКС

ДИСЦИПЛИНЫ «ИНФОРМАТИКА, ЧИСЛЕННЫЕ МЕТОДЫ И ПРОГРАММИРОВАНИЕ»

Специальности
010701 "Физика"
020101 "Химия"

Квалификации
"Физик"
"Химик"

Форма обучения
очная

УМК соответствует учебному плану
подготовки,
утвержденному ректором Москов-
ского государственного университе-
та им. М.В.Ломоносова академиком
РАН В.А. Садовничим 23.10.2009

Москва 2013

Название дисциплины: Информатика, численные методы и программирование.

1. Цели и задачи освоения дисциплины:

Цель: научить работе с ЭВМ, познакомить с основными современными операционными системами и обучить базовым навыкам работы в них, показать полный спектр возможностей ЭВМ, разъяснить роль численных методов и компьютерного моделирования для решения практических задач во всех научных областях современного мира, дать представления о разработке программного обеспечения

Задачи: привить учащимся навыки логического мышления, научить учащихся самостоятельно разрабатывать программы, обучить базовым алгоритмам, используемым в прикладном программировании, обучить языку Си/Си++.

2. Требования к результатам освоения содержания дисциплины:

Изучение дисциплины предполагает знание студентами основ информатики и программирования; базовое умение работы на персональном компьютере. Необходимо знание общей структуры ПК и его составляющих.

В результате освоения дисциплины обучающийся должен:

Знать: - общее устройство компьютера;

- язык программирования С;

- основные методы и структуры данных процедурных языков программирования (в первую очередь для языка Си);

- основные принципы создания больших программ;

- иметь представление об устройстве компьютерных сетей;

Уметь: программировать предложенный алгоритм на языке Си; грамотно пользоваться основными структурами данных при решении прикладных задач, в том числе уметь реализовывать их на языке С.

Владеть: навыками составления программ; методами решения простейших алгоритмических задач.

Приобрести опыт деятельности: в работе с операционной системой семейства UNIX (например, Linux), в составлении алгоритмов решения прикладных задач, в разработке программы для решения прикладных задач (компьютерное моделирование, графические интерфейсы, работа в Интернет).

3. Содержание и структура дисциплины

3.1. Содержание разделов дисциплины (К – коллоквиум, Т – проверочная самостоятельная работа (тест), РК - рубежная контрольная работа, ДЗ – домашнее задание, РГЗ – расчетно-графическое задание)

№ раздела	Наименование раздела	Содержание раздела	Форма текущего контроля
1	Основы устройства компьютера	Устройство и логика операционных систем. Понятие колец защиты, безопасность компьютера. Классификация программ по их назначению и уровню защиты. Файловые системы, их виды и различия. Устройство компонентов компьютера, их назначение и взаимодействие. Устройство	ДЗ

		средств хранения данных, виды жестких дисков.
2	Основы логики	Устройство центрального процессора, логические схемы, понятие логических блоков. Высказывания. Основные логические связи, их применение в компьютере.
3	Программирование на языке Си	Введение в программирование. Общая структура программ. Виды языков программирования. Специфика программ на языке Си.
		Переменные. Типы переменных. Классификация по области видимости. Время жизни переменных. Пользовательские типы. Структуры. Понятие динамических переменных.
		Арифметические операции. Использование математической библиотеки, математические функции.
		Указатели. Два основных применения указателей. Ошибки при выделении памяти, специфика динамического выделения памяти. Исключительные ситуации, отладчик.
		Работа с файлами. Понятие потоков ввода/вывода.
		Автоматизация процессов.
		Операторы условия. Виды условий, применение логических выражений. Применение условий при программировании. Вложенность.
		Операторы цикла. Виды и назначение операторов цикла в Си. Заикливание. Рекурсия, потеря памяти и эффективное программирование.
Функции и процедуры в языках программирования. Устройство и специфика использования функций в языке Си. Возвращаемые значения. Функции не имеющие возвращаемого значения.		
4	Реляционные базы данных	Базы данных с точки зрения системного аналитика. Виды баз данных. Способы разработки структур баз данных. Язык SQL.
5	Компьютерные сети	Локальные и глобальные сети. Internet. Электронная почта, телеконференции, WWW.

3.2. Структура дисциплины

Общая трудоемкость дисциплины составляет 220 часов, из них лекции – 70 часов, семинары – 102 часа, самостоятельная работа студентов – 48 часов.

Вид работы	Семестр 1	Семестр 2	Семестр 3	Семестр 4	Всего
Общая трудоемкость	84	62	48	26	220
Аудиторная работа:	72	48	36	16	172
Лекции (Л)	36	16	18	0	70

Практические занятия (ПЗ)	36	32	18	16	102
Лабораторные работы (ЛР)	0	0	0	0	0
Самостоятельная работа	12	14	12	10	48
Вид итогового контроля	Зачет	Зачет	Зачет	Зачет	

Разделы дисциплины по семестрам

№ раздела	Наименование раздела	Количество часов				
		Всего	Аудиторная работа			Внеаудиторная работа
			Л	ПЗ	ЛР	
1	Основы устройства компьютера	42	18	18		6
2	Основы логики	42	18	18		6
3	Программирование на языке Си	62	16	32		14
4	Реляционные базы данных	48	18	18		12
5	Компьютерные сети	26	0	16		10
	Итого:	220	70	102		48

3.3. Практические занятия (семинары)

№ раздела	№ занятия	Тема	Кол-во часов
1	1	Устройство и логика операционных систем. Понятие колец защиты, безопасность компьютера. Классификация программ по их назначению и уровню защиты. Файловые системы, их виды и различия. Устройство компонентов компьютера, их назначение и взаимодействие. Устройство средств хранения данных, виды жестких дисков.	18
2	2	Устройство центрального процессора, логические схемы, понятие логических блоков. Высказывания. Основные логические связки, их применение в компьютере.	18
3	3	Введение в программирование. Общая структура программ. Виды языков программирования. Специфика программ на языке Си.	4
	4	Переменные. Типы переменных. Классификация по области видимости. Время жизни переменных. Пользовательские типы. Структуры. Понятие динамических переменных.	4
	5	Арифметические операции. Использование математической библиотеки, математические функции.	2
	6	Указатели. Два основных применения указателей. Ошибки при выделении памяти, специфика динамического выделения памяти. Исключительные ситуации,	4

		отладчик.	
	7	Работа с файлами. Понятие потоков ввода/вывода.	4
	8	Автоматизация процессов.	2
	9	Операторы условия. Виды условий, применение логических выражений. Применение условий при программировании. Вложенность.	4
	10	Операторы цикла. Виды и назначение операторов цикла в Си. Заикливание. Рекурсия, потеря памяти и эффективное программирование.	4
	11	Функции и процедуры в языках программирования. Устройство и специфика использования функций в языке Си. Возвращаемые значения. Функции не имеющие возвращаемого значения.	4
4	12	Базы данных с точки зрения системного аналитика. Виды баз данных. Способы разработки структур баз данных. Язык SQL.	18
5	13	Локальные и глобальные сети. Internet. Электронная почта, телеконференции, WWW.	16

3.4. Самостоятельное изучение разделов дисциплин

№ раздела	№ вопроса	Вопросы, выносимые на самостоятельное изучение	Кол-во часов
1	1	Основы устройства компьютера	6
2	2	Основы логики	6
3	3	Программирование на языке Си	14
4	4	Реляционные базы данных	12
5	5	Компьютерные сети	10

4. Образовательные технологии

4.1. Интерактивные образовательные технологии, используемые в аудиторных занятиях

Семестр	Вид занятия	Интерактивные образовательные технологии	Кол-во часов
1-4	Лекции, семинары	мультимедийный проектор, презентация, интерактивная доска, вычислительная техника	172
Итого			172

5. Оценочные средства для текущего контроля успеваемости и промежуточной аттестации

5.1. Типовые практические домашние задания:

В качестве домашней работы, как правило, выступают задачи, задаваемые для практической работы, которые невозможно решить в классе в заданное время, в связи с его ограниченностью.

5.2. Типы задач для самостоятельного выполнения.

Отчет о выполненной домашней работе представляется в виде программ на языке С. Кроме решения, собственно, задачи, в предоставляемой программе должен присутствовать корректно составленный тест, с помощью которого можно убедиться в работоспособности предоставляемой программы.

1. Работа с последовательностями

В каждой задаче входными данными являются числа, записанные в файле. Выходные данные должны выдаваться на экран. В случае невозможности считывания исходных данных на экран должно выдаваться соответствующее сообщение об ошибке.

1. Среднее арифметическое чисел из последовательности.
2. Количество чисел, больших предыдущего.
3. Есть ли в последовательности число X ?
4. Есть ли в последовательности числа, не принадлежащие локальным максимумам или локальным минимумам?
5. Каких элементов последовательности больше: с четными значениями, или с нечетными?
6. Каких элементов последовательности больше: равных первому элементу последовательности, или второму?
7. Каких элементов последовательности больше: больших предыдущего, или меньших предыдущего?
8. Каких элементов последовательности больше: больших первого элемента последовательности, или меньших?
9. Что больше: сумма элементов массива с четными индексами, или с нечетными?
10. Сколько элементов последовательности находятся в интервале между значениями первого и второго элемента последовательности, т.е. найти количество элементов последовательности, для которых выполняется соотношение:
$$\min(a_1, a_2) < a_i < \max(a_1, a_2).$$
11. Номер последнего числа, равного X .
12. Номер последнего числа, равного первому или второму элементу последовательности (ответ может быть равным 1 или 2).
13. Все ли элементы последовательности равны между собой?
14. Все ли элементы последовательности, принадлежащие локальным минимумам, равны между собой?
15. Что больше: сумма элементов последовательности с четными значениями, или с нечетными?
16. Является ли последовательность возрастающей, убывающей?
17. Является ли последовательность арифметической прогрессией?
18. Является ли последовательность геометрической прогрессией?
19. Является ли последовательность знакопеременной? Т.е. все ли соседние элементы последовательности имеют разный знак? Будем считать, что наличие нулевого элемента последовательности делает ее не знакопеременной.
20. Можно ли убрать из последовательности один элемент таким образом, чтобы вся оставшаяся последовательность состояла из равных элементов?
21. Можно ли убрать из последовательности один элемент таким образом, чтобы вся оставшаяся последовательность была бы возрастающей?
22. Можно ли изменить один элемент последовательности таким образом, чтобы вся оставшаяся последовательность стала бы строго возрастающей?
23. Удовлетворяют ли элементы последовательности данному рекуррентному соотношению $c_1 * a_{i+1} + c_2 * a_i + c_3 * a_{i-1} = b$?
24. Количество различных элементов неубывающей последовательности.
25. Количество различных элементов последовательности, значения элементов ко-

торой принадлежат множеству $\{1,2,3,4,5\}$.

26. Количество различных элементов последовательности, все значения элементов которой либо равны a_1 , либо равны a_2 , либо равны a_3 , либо равны a_4 .
27. Общее количество элементов в постоянных участках последовательности.
28. Номер первого числа, равного максимуму из всех чисел.
29. Номер последнего числа, равного минимуму из всех чисел.
30. Количество чисел, равных минимальному.
31. Количество чисел, не равных максимальному.
32. Количество чисел, равных полусумме предыдущего и последующего элементов последовательности.
33. Количество элементов последовательности, для которых выполняется соотношение: $a_i > i$.
34. Количество элементов последовательности в последнем локальном минимуме последовательности.
35. Максимальная длина локального максимума последовательности.
36. Максимальное значение $|a_i - a_{i+1}|$ в последовательности (максимальный перепад значений).
37. Среднее квадратическое отклонение от среднего арифметического.
38. Экспоненциально взвешенное среднее.
339. Величину максимального отклонения элементов последовательности от среднего значения.
40. Количество возрастающих участков последовательности.
41. Длина наибольшего постоянного участка.
42. Длина наибольшего возрастающего участка.
43. Максимальное расстояние между локальными минимумами.
44. Максимальная сумма подряд идущих элементов последовательности.
45. Последовательность чисел представляет собой коэффициенты многочлена по возрастанию степеней. Вычислить многочлен и его производную в точке x .

2. Работа с массивами

В каждой задаче входными данными являются числа, записанные в файле. Выходные данные должны выдаваться тоже в файл. В случае невозможности считывания исходных данных на экран должно выдаваться соответствующее сообщение об ошибке.

0. Решить все задачи предыдущего раздела, заменив последовательность на массив значений.

1. Симметричны ли значения элементов массива?
2. Переставить элементы массива в обратном порядке.
3. Циклически сдвинуть элементы массива на одну позицию вправо.
4. Сравнить два неупорядоченных целочисленных массива A и B как числовые множества: $A = B$ и $A \subset B$.
5. Удалить из массива все отрицательные значения, а оставшиеся уплотнить (сдвинуть) с сохранением исходного порядка к началу массива.
6. Циклически сдвинуть элементы массива на K позиций вправо с затратой $O(N)$ действий (N -длина массива)
7. Каждый элемент массива заменить на полусумму соседей (кроме первого и последнего)
8. Назовем x -отрезком группу подряд идущих элементов массива, каждый из которых равен x . Для заданного числа x заменить элементы каждого x -отрезка на полусумму элементов, прилегающих к этому отрезку справа и слева. Если x -отрезок расположен в начале или конце массива, считать недостающий крайний элемент равным нулю.
9. Сгруппировать положительные элементы массива в его начале, а отрицательные

- в конце с сохранением их порядка.

10. Получить массив биномиальных коэффициентов для степени N , последовательно вычисляя строки треугольника Паскаля (можно использовать только один массив).
11. Элементы массива не убывают. Двоичным поиском определить позицию, где в этот массив можно вставить данное число x .
12. Даны два неубывающих массива. Построить третий неубывающий массив, который является объединением первых двух.
13. Пусть в массиве последовательно записаны цифры некоторого длинного десятичного числа. Реализовать функции "прибавляющие единицу" и "вычитающие единицу" из такого числа.
14. Удалить все элементы массива с четными значениями, уплотнив массив к его началу. Функция должна возвращать количество элементов в получившемся массиве.
15. Заменить все локальные минимумы в массиве одним элементом, значение которого равно элементу массива с минимальным значением. Функция должна возвращать количество элементов в получившемся массиве.
16. Заменить каждый элемент массива количеством элементов массива с меньшими индексами, имеющими значение, меньше данного элемента, т.е. каждый элемент массива a_i заменить количеством элементов массива a_j , таких что $j < i$ и $a_j < a_i$.
17. Отнормировать все элементы массива $\{a_i\}_{i=1, \dots, N}$ на интервал $[\min(a_1, a_N), \max(a_1, a_N)]$, т.е. в случае, когда все элементы массива совпадают, изменять массив не следует, иначе заменить каждый элемент массива на значение $\min(a_1, a_N) + [(a_i - \min(a_1, a_N)) / |a_1 - a_N|]$.
18. Заменить все минимальные элементы массива на значение элемента массива, большего минимального, но не большего всех остальных элементов массива.
19. Удалить из массива наиболее часто встречающееся значение. Если таких значений несколько, то выбрать любое из них. Функция должна возвращать количество элементов в получившемся массиве.
20. Удалить из массива наименее часто встречающееся значение. Если таких значений несколько, то удалить все эти значения. Функция должна возвращать количество элементов в получившемся массиве.
21. В каждом интервале строгого возрастания в массиве заменить все значения в данном интервале на среднее значение в интервале (рассматриваются интервалы возрастания в массиве, которые нельзя удлинить).
22. Поменять местами в массиве локальные минимумы с соседними порядковыми номерами, т.е. поменять местами в массиве локальные минимумы с номерами $2i$ и $2i+1$ для всех возможных i .
23. Удалить из массива элементы, содержащиеся во всех интервалах строгого возрастания с длиной не более 3. Функция должна возвращать количество элементов в получившемся массиве.

3. Битовые операции

В каждой задаче входные данные считываются с экрана. Выходные данные должны выдаваться на экран. В случае невозможности считывания исходных данных или их некорректности на экран должно выдаваться соответствующее сообщение об ошибке.

1. Возвести число в степень N за не более чем $2 \log_2 N$ умножений.
2. Вывести в файл все подмножества множества $\{1, \dots, N\}$.
3. Вывести в файл все k -элементные подмножества мн-ва $\{1, \dots, N\}$.
4. Проверить четность количества единиц в двоичном представлении данного целого

числа.

5. Найти первые N целых чисел, у которых младший байт является зеркальным отражением следующего байта.
6. Определить позицию самой старшей единицы в битовом представлении данного целого числа.
7. Написать функции, записывающие 0 или 1 в указанный бит данного целого числа и оставляющие остальные биты без изменения.

4. Сортировки

Решение должно содержать отдельную функцию для сортировки массива, функцию для проверки массива на упорядоченность, функцию для чтения массива из файла и функцию для генерирования случайного массива указанной длины. Также нужно определить и вывести время, затраченное на сортировку массива.

Алгоритмы сортировки массивов:

1. Простая сортировка обменов.
2. Пузырьковая сортировка.
3. Сортировка просеиванием.
4. Вставка с последовательным поиском.
5. Вставка с бинарным поиском.
6. Сортировка слиянием.
7. Быстрая сортировка (quicksort).
8. Линейная сортировка целого массива.
9. Сортировка целого массива группировкой с последовательным упорядочиванием битов.

5. Списки

В каждой задаче входные данные считываются из файла. Выходные данные должны выдаваться на экран. В задачах не допускаются отступления от стандартного определения списка. Т.е., например, в списке можно работать только с текущим элементом.

1. Симметричны ли значения элементов списка?
2. Переставить элементы списка в обратном порядке.
3. Циклически сдвинуть элементы списка на одну позицию вперед.
4. Сравнить два списка целых чисел A и B как числовые множества: $A = B$ и $A \subset B$.
5. Удалить из списка все отрицательные значения.
6. Каждый элемент списка заменить на полусумму соседей (кроме первого и последнего)
7. Назовем x -отрезком группу подряд идущих элементов списка, каждый из которых равен x . Для заданного числа x заменить элементы каждого x -отрезка на полусумму элементов, прилегающих к этому отрезку справа и слева. Если x -отрезок расположен в начале или конце списка, считать недостающий крайний элемент равным нулю.
8. Сгруппировать положительные элементы списка в его начале, а отрицательные - в конце с сохранением их порядка.
9. Даны два неубывающих списка. Построить третий неубывающий список, который является объединением первых двух.
10. Пусть в списке последовательно записаны цифры некоторого длинного десятичного числа. Реализовать функции "прибавляющие единицу" и "вычитающие единицу" из такого числа.
11. Удалить все элементы списка с четными значениями. Функция должна возвращать количество элементов в получившемся списке.
12. Заменить все локальные минимумы в списке одним элементом, значение которого равно элементу списка с минимальным значением. Функция должна возвращать количество элементов в получившемся списке.

13. Заменить каждый элемент списка количеством элементов списка с меньшими индексами, имеющими значение, меньше данного элемента, т.е. каждый элемент списка a_i заменить количеством элементов списка a_j , таких что $j < i$ и $a_j < a_i$.
17. Отнормировать все элементы списка $\{a_i\}_{i=1, \dots, N}$ на интервал $[\min(a_1, a_N), \max(a_1, a_N)]$, т.е. в случае, когда все элементы списка совпадают, изменять список не следует, иначе заменить каждый элемент списка на значение $\min(a_1, a_N) + [(a_i - \min(a_1, a_N)) / |a_1 - a_N|]$.
18. Заменить все минимальные элементы списка на значение элемента списка, большего минимального, но не большего всех остальных элементов списка.
19. Удалить из списка наиболее часто встречающееся значение. Если таких значений несколько, то выбрать любое из них. Функция должна возвращать количество элементов в получившемся списке.
20. Удалить из списка наименее часто встречающееся значение. Если таких значений несколько, то удалить все эти значения. Функция должна возвращать количество элементов в получившемся списке.
21. В каждом интервале строгого возрастания в списке заменить все значения в данном интервале на среднее значение в интервале (рассматриваются интервалы возрастания в списке, которые нельзя удлинить).

6. Учебно-методическое обеспечение дисциплины

6.1. Основная литература

1. Д.Кнут. Искусство программирования для ЭВМ. тт 1-3. Москва. Мир. 1996-1998
2. Т.Кормен, Ч.Лейзерсон, Р.Ривест. Алгоритмы. Построение и анализ. Москва. МЦНМО. 1999.
3. Препарата Ф., Шеймос М. Вычислительная геометрия. Москва. Мир. 1989. Альфред В. Ахо, Джон Э. Хопкрофт Тчоалс, Джеффри Д. Ульман. Структуры данных и алгоритмы. Издательский дом "Вильямс". Москва. Санкт-Петербург. Киев. 2003.
4. Роберт В Себеста. Основные концепции языков программирования. 5-е издание. Вильямс. 2001.
5. Вирт Н. Алгоритмы и структуры данных. М.: Мир. 1977.
6. Хусаинов Б.С. Структуры и алгоритмы обработки данных. Примеры на языке Си (+CD). М.: Финансы и статистика, 2004.
7. Подбельский В.В., Фомин С.С. Программирование на языке Си., М.: Финансы и статистика, 2003.
8. Керниган Б., Ритчи Д. Язык программирования Си (2-е издание). Вильямс. 2009.

6.2. Дополнительная литература

1. IEEE Standard for Binary Floating-Point Arithmetic Copyright 1985 by The Institute of Electrical and Electronics Engineers, Inc 345 East 47th Street, New York, NY. 10017, USA
2. Иван Скляров. Головоломки для хакера. СПб.: БХВ-Петербург. 2005
3. Михаил Фленов. Программирование на C++ глазами хакера. СПб.: БХВ-Петербург. 2005.
4. М.Сибуя, Т.Ямамото. Алгоритмы обработки данных.-М.: "Мир", 1986.
5. Р. Круз. Структуры данных и проектирование программ. Бином. Лаборатория знаний. 2008.
6. Л. Г. Гагарина, В. Д. Колдаев. Алгоритмы и структуры данных. Финансы и статистика, Инфра-М. 2009.

7. В. Е. Алексеев, В. А. Таланов. Графы и алгоритмы. Структуры данных. Модели вычислений. Бинوم. Лаборатория знаний, Интернет-университет информационных технологий. 2009.

8. С. М. Окулов. Алгоритмы обработки строк. Бинوم. Лаборатория знаний. 2009.

6.3. Интернет-ресурсы

1. Андрей Богатырев. Руководство полного идиота по программированию (на языке Си) http://physchem.msu.ru/doc/bogatirev_c.htm

2. Вики учебник http://ru.wikibooks.org/wiki/Язык_Си_в_примерах

3. Стандарт языка Си <http://www.open-std.org/jtc1/sc22/WG14/www/docs/n1256.pdf>

6.4. Программное обеспечение современных информационно-коммуникационных технологий

- компилятор GCC
- среды разработки CodeBlocks, NetBeans и QtCreator.

7. Материально-техническое обеспечение дисциплины

Для проведения занятий по дисциплине «Информатика, численные методы и программирование» предоставляется компьютерный класс

- с интерактивной доской;
- с локальным сетевым оборудованием и выходом в сеть Интернет;
- системное программное обеспечение: операционная система Ubuntu Linux, служебные утилиты, компилятор GCC, среды разработки CodeBlocks и NetBeans, среда QtCreator.