

Информатика на Физико-Химическом Факультете МГУ имени М.В.Ломоносова

В краткой программе курса обозначены основные положения, которые должны знать студенты, для сдачи зачёта. Студенты также должны уметь работать с ЭВМ, составлять программный код на языке С, компилировать программный код и запускать полученную программу.

Краткая программа курса:

1. Основы работы в операционной системе семейства UNIX.
2. Идентификаторы. Ключевые слова. Стандартные типы данных.
3. Основные операторы и работа с ними.
4. Инструкции выбора. Циклы.
5. Функции.
6. Массивы. Строки.
7. Указатели.
8. Основы ввода-вывода в языке С.
9. Основные библиотеки С.
10. Системы счисления.
11. Суммы и перебор.
12. Работа с файлом.

Далее идёт расширенная программа курса с пояснениями, и тем, что должны знать и уметь студенты. В Приложении 1 находится список задач, которые необходимо уметь решать. В Приложении 2 находится список литературы по языку С и некоторым алгоритмам. В Приложении 3 находятся небольшие инструкции по установке среды программирования на ПЭВМ.

Рекомендации:

1. Студентам, которые не научились программированию на занятиях, настоятельно рекомендуется дома написать несколько программ самостоятельно. Эти студенты должны обратить внимание на то, что процесс обучения программированию может занять существенное время, поэтому рекомендуется не откладывать подготовку на последний день.
2. Студентам, знающим другие языки программирования такие, как паскаль или бейсик, рекомендуется составить таблицу перевода операторов из их языка в язык С, потому что с ходу (на зачёте) это сделать не получится.
3. Студентам рекомендуется освежить знания по арифметике и геометрии, так как некоторое количество задач опирается на знания 8-9 классов школьной программы.
4. Студентам рекомендуется освоить простейшие навыки математической логики.

1 Основы работы в операционной системе семейства UNIX.

1.1. Работа в shell. Простейшие команды: ls, cat, cp, mv, touch, rm.

Студенты должны знать:

- основные команды ls, cat, cp, mv, touch, rm;

Студенты должны уметь:

- входить в систему со своим логином и паролём;
- создавать файл;
- копировать, перемещать и удалять файл/каталог;
- редактировать файл.

1.2. Компилятор. Сборка программы на языке C компилятором GCC. Запуск программ.

Студенты должны знать:

- схему работы компилятора;

Студенты должны уметь:

- компилировать программу с помощью gcc;
- запускать скомпилированную программу.

2 Основы программирования на языке высокого уровня С.

2.1. Идентификаторы. Ключевые слова. Стандартные типы данных: символы, три типа целых чисел, модификаторы `signed` и `unsigned`, числа с плавающей запятой. Явное преобразование типов данных. Классы памяти: объявление переменных на внешнем и внутреннем уровнях, правила определения области видимости переменных.

Студенты должны знать:

- стандартные типы данных и их представление в памяти ЭВМ;
- преобразования типов данных и классы памяти.

Студенты должны уметь:

- правильно объявлять переменные и константы;
- правильно преобразовывать типы данных;
- определять область видимости переменной.

2.2. Побитовые, арифметические, логические операторы, условный оператор и оператор "запятая". Инкрементирование и декрементирование. Операторы сдвига, присваивания и сравнения. Приоритеты выполнения операторов.

Студенты должны знать:

- основные операторы и приоритеты их выполнения.

Студенты должны уметь:

- записывать выражения по правилам языка С;
- вручную вычислять значения сложных выражений.

2.3. Инструкции выбора: `if`, `if/else`, условный оператор `?:`, конструкция `switch/case`. Циклы: `for`, `while`, `do/while`. Инструкции `break` и `continue`. Функция `exit()`.

Студенты должны знать:

- все стандартные инструкции выбора;
- циклы, инструкции `break` и `continue`.

Студенты должны уметь:

- правильно использовать инструкции выбора при написании программ;
- записывать алгоритмы, содержащие циклы;
- вручную определять условия окончания циклов.

2.4. Функции. Прототипы функций. Синтаксис объявления функции. Аргументы функции: формальные и фактические аргументы, аргументы типа `void`, `char`, `int`, `float`, `double`. Типы значений, возвращаемых функциями: `void`, `char`, `int`, `long`, `float`, `double`. Область видимости переменных.

Студенты должны знать:

- преимущества написания функций перед линейным кодом;
- технику применения функций.

Студенты должны уметь:

- правильно объявлять функции;
- правильно передавать аргументы в функцию;

2.5. Массивы. Свойства массивов. Объявления массивов. Явная инициализация массивов. Доступ к элементам массива. Вычисление размера массива в байтах. Выход за пределы массива. Массивы символов. Строки. Функции работы со строками и массивами символов: `sprintf()`, `strcpy()`, `strcat()`, `strncmp()` и `strlen()`.

Студенты должны знать:

- что такое одномерные массивы;
- представление массивов в памяти ЭВМ;
- что такое строка;
- функции работы со строками и массивами символов.

Студенты должны уметь:

- правильно объявлять и инициализировать массивы;
- получать доступ к элементам массива;
- вычислять размер массива в байтах;
- правильно применять функции работы со строками и массивами символов.

2.6. Указатели. Объявление, использование и инициализация указателей. Арифметические операции над указателями. Применение указателей при работе с массивами. Физическая реализация указателей.

Студенты должны знать:

- что такое указатели;
- физическую реализацию указателей.

Студенты должны уметь:

- правильно объявлять, инициализировать и использовать указатели;
- правильно применять операторы разыменования и взятия адреса;
- вручную определять действия арифметических операций над указателями.

2.7. Основы ввода-вывода в языке C. Подсистема ввода-вывода в языке C. Функции: `printf`, `fprintf`, `scanf`, `fscanf`. Форматирование строк и числовых значений. Файловый ввод-вывод.

Студенты должны знать:

- подсистему ввода-вывода в языке C;
- функции `printf`, `fprintf`, `scanf`, `fscanf`.

Студенты должны уметь:

- правильно использовать функции ввода и вывода данных;
- форматировать строки и числовые значения по заданному образцу;
- получать данные из файла;
- выводить данные в файл;

2.8. Основные библиотеки C. Библиотека `stdio.h`. Преобразование символов в ASCII-код. Библиотека `string.h`. Строковые функции. Библиотека `math.h`. Математические функции.

Студенты должны знать:

- основные библиотеки C;

Студенты должны уметь:

- правильно применять стандартные библиотечные, символьные, строковые и математические функции;

3 Построение простейших алгоритмов и программ

3.1. Функция main. Чтение данных с клавиатуры. Вывод результата на экран.

Студенты должны знать:

- с чего программы запускается;
- где программа завершается.

Студенты должны уметь:

- создавать простейшую программу, читающую целое число с клавиатуры и выводить его на экран;

3.2. Арифметические операции с числами. Деление на цело и остаток от деления. Подсчёт количества разрядов в числе.

Студенты должны знать:

- все простейшие операции.

Студенты должны уметь:

- находить все делители числа;
- проверять число на простоту;
- разбирать число на разряды.

3.3. Системы счисления: двоичная, десятичная, шестнадцатеричная и остальные.

Студенты должны знать:

- как перевести число из одной системы счисления в другую.

Студенты должны уметь:

- читать число с клавиатуры в любой системе счисления;
- выводить на экран число в любой системе счисления.

3.4. Квадратные уравнения. Системы уравнений.

Студенты должны знать:

- как решать квадратные уравнения;
- как решать простейшие системы линейных уравнений.

Студенты должны уметь:

- находить различные характеристики квадратных уравнений;
- находить различные характеристики системы линейных уравнений.

3.5. Геометрические задачи. Площадь треугольника, длина различных элементов в треугольнике.

Студенты должны знать:

- формулы площади треугольника;
- формулы для нахождения внутренних элементов треугольника.

Студенты должны уметь:

- в языке C работать с формулами для вычисления площади и различных элементов в треугольнике.

3.6. Суммы и перебор.

Студенты должны знать:

- как использовать циклы в программе.

Студенты должны уметь:

- с использованием циклов, считать различные суммы;
- подбором находить решения некоторых задач.

3.7. Работа с файлом.

Студенты должны знать:

- как устроен файловый ввод-вывод.

Студенты должны уметь:

- открывать и закрывать файлы;
- читать данные из файла;
- записывать данные в файл.

3.8. Задачи.

Студенты должны знать:

- как решать все задачи из Приложения 1.

Студенты должны уметь:

- решать все задачи из Приложения 1.

4 Приложение 1. Задачи.

Задача 1

Написать программу, которая будет считывать с клавиатуры целое число типа **int** и выводить на экран число разрядов в этом числе. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 2

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводить все четырехзначные числа из диапазона $[1000, n]$, у которых сумма первых двух цифр равна сумме двух последних. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 3

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводить на экран это число в обратном порядке. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 4

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводить на экран 1, если это число простое, и 0, иначе. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 5

Написать программу, которая будет считывать с клавиатуры три числа a, b, c типа **double** и выводить на экран максимум функции $f(x) = ax^2 + bx + c$. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 6

Написать программу, которая будет считывать с клавиатуры 6 чисел a, b, c, d, e, f типа **double** и выводить на экран количество решений системы уравнений

$$\begin{cases} ax + by = c, \\ dx + ey = f. \end{cases}$$

Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 7

Написать программу, которая будет считывать с клавиатуры 3 числа a, b, γ типа **double** и выводить на экран длину третьей стороны треугольника с вершинами A, B, C (использовать стандартные обозначения для треугольника). Для данной программы необходимо в коде программы подключить математическую библиотеку *math.h*. При сборке программы необходимо слинковать эту библиотеку:

gcc -lm a.c

Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 8

Написать программу, которая будет считывать с клавиатуры 3 числа a, b, γ типа **double** и выводить на экран площадь треугольника с вершинами A, B, C (использовать стандартные обозначения для треугольника). Для данной программы необходимо в коде программы подключить математическую библиотеку *math.h*. При сборке программы необходимо слинковать эту библиотеку:

gcc -lm a.c

Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 9

Написать программу, которая будет считывать с клавиатуры целое число n типа **int**, заменить в нём все «единицы» на «пятерки» и выводить полученное число на экран. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 10

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводит на экран число из отрезка $[1, n]$, имеющее наибольшее количество делителей. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 11

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводит на экран число равное количеству троек в десятичной записи n . Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 12

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводит на экран его в двоичной системе счисления. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 13

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** и выводит на экран его в троичной системе счисления. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 14

Написать программу, которая будет считывать с клавиатуры два целых числа a, b типа **int** и выводить на экран их наибольший общий делитель. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 15

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** в 15-ричной системе счисления и выводить его в десятичной системе счисления. Для обозначения цифр 10, 11, 12, 13, 14 при вводе числа использовать буквы A, B, C, D, E, соответственно. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

Задача 16

Написать программу, которая будет считывать с клавиатуры целое число n типа **int** в десятичной системе счисления и выводить его в восьмеричной системе счисления. Для обозначения цифр 10, 11, 12 при вводе числа использовать буквы A, B, C, соответственно. Программа должна учитывать все крайние случаи и выводить соответствующее сообщение в случае ошибки.

5 Приложение 2. Литература.

1. Керниган Б., Ритчи Д.. Язык программирования С.
2. Подбельский В.В., Фомин С.С. Программирование на языке С.
3. Курсков С.Ю. Введение в язык С. (<http://dfe3300.karelia.ru/koi/posob/c/c.htm>)
4. А. Шень. Программирование: теоремы и задачи.

6 Приложение 3. Среды разработки

1. Установка среды программирования в системе Linux (Ubuntu Linux, Debian Linux , ...).

Вам необходимо поставить на вашу систему только два пакета **gcc** и **mc**. После чего ваш Linux будет полностью схож с той системой, которая стоит в классе. Команды на установку (использую интернет):

```
apt-get install gcc
```

```
apt-get install mc
```

2. Установка среды программирования в системе Windows

Можно ставить программный комплекс Microsoft Visual C++ Express Edition, который есть в бесплатном доступе на сайте Microsoft.

3. Можно ставить эмулятор Unix shell'a **cygwin**, что позволит на системе Windows воссоздать интерфейс, который стоит в компьютерном классе. При установке cygwin необходимо выбрать в списке пакетов пакеты **gcc** и **mc**.