

## Задания для практических занятий по курсу «Информатика» и оценка знаний обучающихся на основе балльной системы.

Данное учебно-методическое пособие предназначено для проведения практических занятий в первом семестре у студентов, обучающихся по направлению «Фундаментальная математика». Пособие содержит практические задания для самостоятельной разработки программы на языке С. Последовательность практических заданий представлена в порядке изложения теоретического материала в лекционном курсе.

В результате выполнения предложенных заданий обучающиеся осваивают первоначальные основы программирования, необходимые для конструирования простых алгоритмов. К концу данного этапа обучения студенты должны уметь проектировать, описывать на языке С и тестировать простые алгоритмы математических задач.

Используемое программное обеспечение: Microsoft Visual Studio или CodeBlocks.

В течение семестра студенты должны выполнить по 5-7 заданий в соответствии с программой курса. Тематика заданий приведена ниже и соответствует РПУД. Оценивание работы студентов во время практических занятий осуществляется по балльной системе. Выполнение всех задач позволяет получить 100 баллов. Часть заданий отмечена как «обязательные», за эти задания можно набрать 70 баллов.

Зачет по итогам семестра ставится за выполнение всех «обязательных» задач и общей сумме баллов не менее 80.

Тема	Сумма баллов / Общая сумма за обязательные задания / баллов
1. Линейный алгоритм <i>вопросы для обсуждения: структура проекта, типы данных, переменные, ввод/вывод, арифметические операции.</i>	0 / 5
2. Разветвляющийся алгоритм <i>вопросы для обсуждения: логические операции, условный оператор, оператор выбора.</i>	0 / 5
3. Циклический алгоритм <i>вопросы для обсуждения: арифметический цикл, цикл с постусловием, цикл с предусловием.</i>	8 / 10
4. Функции <i>вопросы для обсуждения: структура функции, область видимости переменных, передача параметров функции по значению и по указателю, прототип функции.</i>	6 / 10
5. Рекурсивные функции <i>вопросы для обсуждения: глубина рекурсии, прямая и не прямая рекурсия, условия выхода из рекурсии, рекурсия и итерация.</i>	8 / 10
6. Знакомство с графическими возможностями среды CodeBlocks <i>вопросы для обсуждения: подключение графической библиотеки, основные функции библиотеки «graphics.h».</i>	7 / 7
7. Одномерные массивы <i>вопросы для обсуждения: статические и динамические массивы, использование указателей, массив в качестве параметра функции.</i>	6 / 10
8. Сортировка одномерных массивов <i>вопросы для обсуждения: виды алгоритмов сортировки.</i>	7 / 10
9. Многомерные массивы	9 / 10

вопросы для обсуждения: статические массивы, использование указателей, динамические массивы, многомерный массив в качестве параметра функции

10. Массивы символов

9 / 13

вопросы для обсуждения: C-строки, функции библиотеки «string.h», текстовые файлы, ввод из файла, вывод в файл, запуск программы из командной строки, параметры функции main().

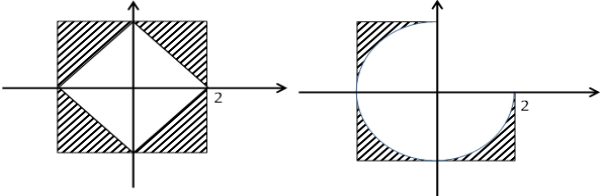
11. Контрольная работа

10 / 10

Итого

70 / 100

Набор задач в каждой теме может варьироваться в зависимости от уровня подготовки в группе.

<b>Линейный алгоритм</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>
1	Даны координаты трех вершин треугольника: (x1, y1), (x2, y2), (x3, y3). Найти его периметр и площадь.		0,5
2-1	Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести полученное число.		0,5
2-2	Дано трехзначное число. В нем зачеркнули первую справа цифру и приписали ее слева. Вывести полученное число.		
3	С начала суток прошло N секунд (N — целое). Найти количество минут и секунд, прошедших с начала последнего часа. Вывести в формате мм:сс (N=11107 - 3 часа, 5 минут, 7 секунд; вывод: 05:07)		1
4	Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, . . . , 6 — суббота, 7 — воскресенье. Дано целое число K, лежащее в диапазоне 1–365, и целое число N, лежащее в диапазоне 1–7. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было днем недели с номером N.		2
5	Даны целые положительные числа A, B, C. На прямоугольнике размера A × B размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике, а также площадь незанятой части прямоугольника.		1
<b>сумма</b>			<b>5</b>
<b>Разветвляющийся алгоритм</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>
1	Известны год, номер месяца и день рождения каждого из двух человек. Определить, кто из них старше.		0,5
2	Даны числа x, y. Проверить истинность высказывания: «Точка с координатами (x, y) лежит в заштрихованной области». 		1
3	Для данного вещественного x найти значение следующей функции f, принимающей значения целого типа: $f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \in [0, 1), [2, 3), \dots \\ -1, & x \in [1, 2), [3, 4), \dots \end{cases}$		0,5
4	Даны длины сторон двух треугольников. Определить подобны ли эти треугольники. (Предполагается сортировка трех чисел, с помощью		1

	условного оператора).		
5	Дан номер года (положительное целое число). Определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).		1
6-1	Дано целое число в диапазоне 20–69, определяющее возраст (в годах). Вывести строку-описание указанного возраста, обеспечив правильное согласование числа со словом «год», например: 20 — «двадцать лет», 32 — «тридцать два года», 41 — «сорок один год».		1
6-2	Ввести сумму денег (целое число копеек до рубля включительно) и вывести эту сумму словами. Например, если вводится число 23, надо вывести «двадцать три копейки».		
<b>сумма</b>			<b>5</b>
<b>Циклический алгоритм</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>
1	(while) Напечатать таблицу значений функции на отрезке [a, b] с шагом h. Функцию придумать самим, она должна иметь разрыв на отрезке [a, b].	Да	2
2	(for) Вывести первые n членов рекуррентной последовательности заданной формулой $y_n = y_{n-1} + y_{n-3}$ .		1
3	(do while) Вычислить сумму ряда Маклорена.	Да	2
4	Составить программу для нахождения цифрового корня натурального числа. (Цифровой корень числа 34697 равен 2 ( $3+4+6+9+7 = 29$ ; $2+9 = 11$ ; $1+1 = 2$ )).		1
5	Вывести на экран, из каких простых множителей состоит введенное натуральное число.	Да	2
6	Составить программу возведения заданного числа в третью степень, используя следующую закономерность: $1^3 = 1$ $2^3 = 3 + 5$ $3^3 = 7 + 9 + 11$ $4^3 = 13 + 15 + 17 + 19$ $5^3 = 21 + 23 + 25 + 27 + 29$	Да	2
<b>сумма</b>			<b>10</b>
<b>Функции</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>
1	Напишите функцию, которая принимает параметр – натуральное число N – и выводит на экран квадрат из звездочек со стороной N.		0,5
2	Написать функцию с тремя целыми параметрами m, n, d, изображающую в текстовом окне терминала при помощи символов «*» рамку размером m на n толщиной d.		1
3	Написать функцию с одним целым параметром, изображающую в текстовом окне терминала окружность заданного радиуса при помощи символов «*».		2
4	Написать функцию, которая определяет количество разрядов введенного целого числа.		0,5
5	Определите НОК трех натуральных чисел. Указание: $\text{НОК}(a, b) = a * b / \text{НОД}(a, b)$ .	Да	2
6	Найти сумму двух дробей a/b и c/d. Ответ записать обыкновенной дробью. Указание: написать функцию cancel(a,b,r,q), которая дробь a/b приводит к несократимому виду p/q .	Да	2

7	<p>Напишите функцию, которая определяет является ли первый принимаемый параметр числом, равным возведению некоторого другого числа в степень равную второму параметру.</p> <p>Например:  <code>func(81,4) ==&gt;3</code> <math>81 = 3</math> в степени 4  <code>func(81,2) ==&gt;9</code> <math>81 = 9</math> в степени 2  <code>func(16,4) ==&gt;2</code> <math>16 = 2</math> в степени 4  <code>func(27,4) ==&gt;</code> без вариантов</p>	Да	2
<b>сумма</b>			<b>10</b>
<b>Рекурсивные функции</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>
1	Дано натуральное число N. Вычислите сумму его цифр. При решении этой задачи нельзя использовать циклы и массивы. Разрешена только рекурсия и целочисленная арифметика.		1
2	Рекурсивно описать функцию <code>pow(x,n)</code> , вычисляющую $x^n$ для любого вещественного $x \neq 0$ и любого целого n.		1
3	Рекурсивно описать функцию <code>maxdig(N)</code> , которая находит наибольшую цифру в десятичной записи неотрицательного целого числа N. Например: <code>maxdig(27306) = 7</code>	Да	2
4	Рекурсивно описать функцию <code>RevPrint(N)</code> , которая печатает в обратном порядке цифры десятичной записи целого неотрицательного числа N. Например: <code>RevPrint(12345)</code> должна вывести текст 54321.	Да	2
5	Дана последовательность чисел, завершающаяся числом 0. Найдите сумму всех этих чисел, не используя цикл. (Само число 0 в последовательность не входит, а служит как признак ее окончания).	Да	2
6	Дано натуральное число $n > 1$ . Проверьте, является ли оно простым. Программа должна вывести слово YES, если число простое и NO, если число составное. Указание. Понятно, что задача сама по себе нерекурсивна, т.к. проверка числа n на простоту никак не сводится к проверке на простоту меньших чисел. Поэтому нужно сделать еще один параметр рекурсии: делитель числа, и именно по этому параметру и делать рекурсию.	Да	2
<b>сумма</b>			<b>10</b>
<b>Знакомство с графическими возможностями среды CodeBlocks</b>			
<b>№</b>	<b>Задание:</b>	<b>Обязательность</b>	<b>Балл</b>
1	Изучить возможности графической библиотеки <code>OpenBGI</code> и нарисовать геометрический (конструктивный) фрактал.	Да	7
<b>сумма</b>			<b>7</b>
<b>Контрольная работа</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>
1	Практическое задание на одну из тем: линейный, разветвляющийся, циклический алгоритм.	Да	2
2	Практическое задание на тему функции.	Да	4
3	Практическое задание на тему рекурсия.	Да	4
<b>сумма</b>			<b>10</b>
<b>Одномерные массивы</b>			
<b>№</b>	<b>Задания:</b>	<b>Обязательность</b>	<b>Балл</b>

1-1	В массиве из $n$ элементов числа образуют неубывающую последовательность. Несколько элементов, идущих подряд, равны между собой. Сколько различных чисел имеется в массиве?		1
1-2	Дан массив ненулевых целых чисел. Определить, сколько раз элементы массива при просмотре от его начала меняют знак. Например, в массиве 10, -4, 12, 56, -4, -89 знак меняется 3 раза.		
2	Дан массив целых чисел. Рассмотреть отрезки массива (группы идущих подряд чисел), состоящие из нечетных чисел. Получить наибольшую из длин рассматриваемых отрезков.		2
3	Удалить из массива все повторяющиеся элементы, оставив их первые вхождения, т. е. в массиве должны остаться только различные элементы. (Под удалением элемента массива следует понимать: исключение этого элемента из массива путем смещения всех следующих за ним элементов влево на 1 позицию; присваивание последнему элементу массива значения 0.)	Да	2
4	Найти $\max_2$ в одномерном массиве. (Наибольший элемент массива при исключении максимального элемента этого массива; один проход по массиву)		1
5	Найти заданный элемент в упорядоченном массиве. (Бинарный поиск)	Да	2
6	Объединить два упорядоченных массива размером $m$ и $n$ в упорядоченный массив размером $m+n$ .	Да	2
<b>сумма</b>			<b>10</b>

### Сортировка одномерных массивов

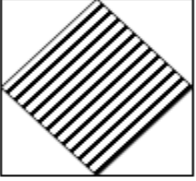
№	Задания:	Обязательность	Балл
Реализовать и сравнить эффективность сортировок. При сравнения сортировок для заданного массива вывести количество сравнений и количество перестановок элементов массива (сравнения и перестановки индексов элементов не считать)			
1	Сортировка пузырьком.	Да	2
2	Сортировка quick sort.	Да	5
3	Любой алгоритм сортировки на месте (шейкерная сортировка, сортировка вставками, сортировка Шелла, сортировка расчёской и др.)		3
<b>сумма</b>			<b>10</b>

### Многомерные массивы

№	Задания:	Обязательность	Балл												
1	Считаем, что двумерный массив размера $m$ на $n$ преобразуется в одномерный массив размера $m \cdot n$ путем последовательной записи строк. ( $m$ и $n$ заданы) Реализовать такое преобразование «туда» и «обратно». Например: <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <p>A</p> <table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>2</td><td>3</td></tr> <tr><td>11</td><td>22</td><td>33</td></tr> </table> </div> <div style="text-align: center;"> <p>B</p> <table border="1" style="border-collapse: collapse;"> <tr><td>1</td><td>2</td><td>3</td><td>11</td><td>22</td><td>33</td></tr> </table> </div> </div> <p>«туда»: Для каждого элемента <math>A[i, j]</math> найти <math>B[k]</math> (по значениям <math>i</math> и <math>j</math> вычислить значение <math>k</math>)  «обратно»: Для каждого элемента <math>B[k]</math> найти <math>A[i, j]</math> (по значению <math>k</math> вычислить значения <math>i</math> и <math>j</math>)</p>	1	2	3	11	22	33	1	2	3	11	22	33	Да	1
1	2	3													
11	22	33													
1	2	3	11	22	33										
2	Заполнить двумерный массив размером $n$ на $m$ так, как представлено в таблице для массива размером $5 \times 5$	Да	2												

1	2	3	4	5
2	1	2	3	4
3	2	1	2	3
4	3	2	1	2

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8

	5	4	3	2	1	13	12	11	10	9		
3	Дан квадратный массив целых чисел 9 x 9. Вычислить сумму элементов из заштрихованной области.											0,5
												
4	Отсортировать двумерный массив по возрастанию сумм/произведений элементов столбцов/строк											0,5
	9 5 2			2 5 9								
	3 3 3			3 3 3								
	12 8 5			5 8 12								
5	Решить систему линейных уравнений методом Гаусса. (n уравнений, n неизвестных, определитель не равен 0)										Да	2
6	Найти обратную матрицу методом Гаусса. (Определитель не равен 0, сделать проверку)										Да	2
7	Найти определитель матрицы, путем приведения к диагональному виду(определитель не равен 0)										Да	2
<b>сумма</b>											<b>10</b>	

### Массивы символов - строки

№	Задания:	Обязательность	Балл
1	Обработка строки: Дана строка, может заканчиваться точкой. Между словами, в начале строки и перед точкой могут стоять «лишние» пробелы (более одного пробела между словами, пробелы в начале строки, пробелы между последним словом и точкой или в конце строки).		
1-a	Удалить лишние пробелы, используя функции библиотеки <code>string.h</code>	Да	2
1-b	Удалить лишние пробелы, выполнив один проход без создания дополнительной строки (использовать два указателя — для чтения и записи символов).		3
1-c	Напечатать слова этой строки, не совпадающие с последним словом. входная строка: " df dfgdfg df asd df . "	Да	2
	результат: "dfgdfg asd"		
1-d	Индивидуальное задание: напечатать слова, полученные в п.1-с, которые удовлетворяют условию или преобразованы по правилу. <b>Варианты индивидуальных заданий :</b> 1. Удалить последнюю букву слова. (‘abcd ab’->’abc a’) 2. Поменять местами первую и последнюю букву каждого слова. (‘abcd’->’dbca’) 3. Если первая буква слова входит в это слово еще раз. (‘abcd as aaa’->’abcd aaa’) 4. Если первая буква слова входит в это слово только один раз. (‘abcd as aaa’->’as’) 5. Если первая буква слова входит в это слово ровно два раза. (‘abcd as aaa’->’abcd’) 6. Если длина слова максимальна. (‘abcd as’->’abcd’) 7. Если длина слова минимальна. (‘abcd as’->’as’) 8. Если в слове нет повторяющихся букв. (‘sabcd asds’->’sabcd’) 9. Если каждая буква слова входит в это слово не менее двух раз.	Да	3

	<p>(‘abab aba’-&gt;’abab’)</p> <p>10. Удалить из слова все последующие вхождения первой буквы этого слова.(‘abcab’-&gt;’abcb’)</p> <p>11. Удалить из слова все предыдущие вхождения последней буквы этого слова.(‘abcab’-&gt;’acab’)</p> <p>12. Оставить в слове только первые вхождения каждой буквы. (‘abcab’-&gt;’abc’)</p> <p>13. Если слово нечетной длины, удалить его среднюю букву. (‘abcab’-&gt;’abab’)</p> <p>14. Если слово четной длины, вставить в его середину первую букву этого слова. (‘abab’-&gt;’abaab’)</p> <p>15. Перевернуть слово. (‘abcab’-&gt;’bacba’)</p> <p>16. Если первая и последняя буква слова совпадают. (‘abcab aba’-&gt;’aba’)</p> <p>17. Удвоить вхождение каждой буквы слова. (‘abcab’-&gt;’aabbccaabb’)</p> <p>18. Если каждая буква входит в слово ровно два раза. (‘aab abcabcc abba’-&gt;’abba’)</p> <p>19. Если последняя буква входит в слово не более двух раз. (‘ab abcabcc abba’-&gt;’ab abba’)</p> <p>20. Оставить в слове только последние вхождения каждой буквы. (‘abcab’-&gt;’cab’)</p> <p>21. Удалить одно вхождение удвоенной буквы. (‘aab abcabcc’-&gt;’ab abcab’)</p> <p>22. Оставить в слове только последнее вхождение первой буквы. (‘abc ababa’-&gt;’abc bba’)</p> <p>23. Оставить в слове только первое вхождение последней буквы. (‘abc ababa’-&gt;’abc abb’)</p> <p>24. Удалить удвоенные буквы. (‘aab abcabcc’-&gt;’b abcab’)</p>		
2	Ввод строки из файла и запись в файл.	Да	2
3	Передача имен файлов через параметры функции main() и запуск программы из командной строки.		1
<b>сумма</b>			<b>13</b>
<b>ИТОГО</b>			<b>100</b>

### Литература

1. Керниган , Б.У. Язык программирования С/Брайан У. Керниган, Деннис М. Ритчи –2 издание - “Вильямс”, Москва-Петербург-Киев, 2017. -288 с.
2. Кнут Д.Э. Искусство программирования, тт.1-3. “Диалектика-Вильямс ”, Москва-Петербург-Киев, 2005 г.
3. Вирт Н. Алгоритмы и структуры данных. (Новая версия для Оберона ), ДМК Пресс, 2014 г
4. Златопольский Д. М. Сборник задач по программированию. 3-е изд., СПб.: БХВ-Петербург, 2011. 304 с.: