

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВПО «Омский государственный университет им. Ф.М. Достоевского»

Институт математики и информационных технологий

Информатика. Методические указания и задания.

Часть I.

Омск 2018

Информатика. Методические указания и задания. Часть I / Сост.: Н.С. Бахта, Е.В. Ушакова, Омск: ФГБОУ ВПО «ОмГУ им. Ф.М. Достоевского», 2018. с. 27

В методических указаниях по информатике приведены задания для лабораторных работ и варианты индивидуальных программ, рекомендации и требования к ним для студентов первого курса ИМИТ специальностей «Прикладная математика и информатика» и «Фундаментальная математика и механика». Указания могут быть использованы при проведении лабораторных работ по курсу «Информатика и вычислительная техника» для студентов естественных и экономических специальностей.

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ.....	4
ЛАБОРАТОРНАЯ РАБОТА №1. Задания на описание переменных и применение форматного ввода-вывода.....	5
ЛАБОРАТОРНАЯ РАБОТА №2. Задания на вычисление арифметических выражений.....	6
ЛАБОРАТОРНАЯ РАБОТА №3. Задания на применение операторов ветвления, цикла и выбора.....	8
ЛАБОРАТОРНАЯ РАБОТА №4. Задания на составление подпрограмм, использование статических массивов.....	9
ЛАБОРАТОРНАЯ РАБОТА №5. Задания на использование файлов, указателей и динамической памяти.....	10
ЛАБОРАТОРНАЯ РАБОТА №6. Задания на использование текстовых и двоичных файлов.....	11
ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №1.....	12
ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №2.....	21
ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №3.....	24
ЛИТЕРАТУРА.....	27

ПРЕДИСЛОВИЕ

В современных условиях специалист-математик сталкивается с необходимостью обращаться к помощи вычислительной техники, и все чаще компьютер помогает находить решение тех задач, которые до сих пор считались «чисто» математическими. Задачи же из области прикладной математики давно требуют написания программных комплексов.

Основная цель курса «Информатика» - научить студентов азам программирования, умению составлять алгоритмы для решения математических задач, используя для этого стандартные средства, создавать простейший человеко-машинный интерфейс. В процессе обучения студенты должны получить навыки конструирования программ из подпрограмм и модулей, оформляя их в соответствии с требованиями структурного, объектного и защитного программирования.

Методические указания содержат задания для лабораторных работ и варианты индивидуальных программ, а также требования и рекомендации к их выполнению. Предлагаемые задачи подкрепляют теоретический материал курса «Информатика».

Также указания могут быть использованы при проведении лабораторных работ по курсу «Информатика и вычислительная техника» для студентов естественных и экономических специальностей.

ЛАБОРАТОРНАЯ РАБОТА №1

Задания на описание переменных и применение форматного ввода-вывода

1. Пусть $x = 829,356$ и $y = 128,286$. Составьте программу, которая выводит на экран эти значения в указанных ниже форматах:

1) $829.356 + 128.286 = ?$

2) $8.293560e + 02$

3) $1.28e + 02$

4) 829.36

5)128.286

2. Составьте программу, позволяющую вычислить значение вещественной переменной z при заданных значениях $\text{int } x = 2$; $\text{int } y = 5$. Результат оформить в виде таблицы:

№	Выражение	Результат
1.	$z=2/5$	
2.	$z=2./5$	
3.	$z=2/5.$	
4.	$z=x/5$	
5.	$z=x/5.$	
6.	$z=x/y$	
7.	$z=\text{double}(x)/5$	
8.	$z=x/\text{double}(y)$	
9.	$z=2\%5$	
10.	$z=y\%x$	

3. Составьте программу, которая для вещественных переменных x и y вычисляет: $x + y$, $x - y$, $x * y$, $\frac{x}{y}$. Значения переменных вводит пользователь. Результат вывести на экран, запросив у пользователя количество позиций для вывода значения и количество отображаемых знаков в дробной части.

4. Написать программу, которая спрашивала бы пользователя, как его зовут, а затем выводила приветствие с использованием этого имени так, чтобы последняя буква приветствия находилась в 40-й позиции на экране от левого края.

ЛАБОРАТОРНАЯ РАБОТА №2

Задания на вычисление арифметических выражений

Справка: стандартная математическая библиотека в языке СИ подключается с помощью команды `#include <cmath>`. В эту библиотеку включены следующие математические функции:

`abs(a)` – модуль числа a ;
`sqrt(a)` – квадратный корень из неотрицательного числа a ;
`pow(x,y)` – возведение числа x в степень y ;
`sin(x)` – синус угла x , заданного в радианах;
`cos(x)` – косинус угла x , заданного в радианах;
`tan(x)` – тангенс числа x ;
`asin(x)` – арксинус числа x ;
`acos(x)` – арккосинус числа x ;
`atan(x)` – арктангенс числа x ;
`exp(x)` – экспонента числа x ;
`log(x)` – натуральный логарифм числа x ;
`floor(x)` – округление числа x до ближайшего целого, не большего x ;
`ceil(x)` – округление числа x до ближайшего целого, не меньшего, чем x .

1. Дана система из двух линейных уравнений с двумя неизвестными:

$$\begin{cases} ax + by = e, \\ cx + dy = f. \end{cases}$$

Напишите программу, которая вычисляет x и y по формулам:

$x = \frac{\Delta x}{\Delta}$, $y = \frac{\Delta y}{\Delta}$, где $\Delta = ad - bc$, $\Delta x = ed - fb$, $\Delta y = af - ce$. Для того, чтобы проверить, правильно ли работает программа, придумайте систему уравнений с заранее известным решением и посмотрите, совпадет ли это решение с тем, которое вычисляет программа.

Что произойдет, если ввести коэффициенты уравнений так, что $ad - bc = 0$?

2. Заданы вершины треугольника (своими координатами на плоскости): $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Напишите программу, вычисляющую длину медианы, проведенной из вершины A по формуле:

$m_A = \frac{1}{2} \sqrt{2a^2 + 2b^2 - c^2}$, где a , b , c – длины сторон треугольника,

которые можно вычислить, используя координаты вершин.

3. Программу из предыдущего задания переделайте так, чтобы координаты вершин треугольника были организованы в виде структур с двумя полями. Первое поле – абсцисса, а второе – ордината.

4. Напишите программу, вычисляющую значения следующих функций (запрашивая необходимые данные у пользователя и считая, что вводимые данные принадлежат области определения соответствующей функции):

- 1) $20 \sin x - |x^3 - x^2|;$
- 2) $\frac{|x + y|}{x - y};$
- 3) $2^{x+y} \cdot \log_2(xy);$
- 4) $\sqrt[6]{ab(cd - ef)};$
- 5) $\frac{xy - 4|x + e^x|}{(10^6 + \sqrt{\ln y})};$
- 6) $2 \sin \frac{x+y}{2+x} \cos \frac{x-y}{2-x};$
- 7) $\text{tg} \frac{x}{3} \text{arctg } y;$
- 8) $\arcsin x.$

5. Запишите выражение на языке СИ в общепринятой математической форме: $x^2 + \sin(y^2 - \text{alfa}) / \text{abs}(x^4 - \exp(-5)) * x$.

6. Переменные a и c имеют значения 6 и 4 соответственно. Какое значение примет переменная b после выполнения оператора присваивания:

- 1) $b = a * 2 / 3 * c;$
- 2) $b = (a * 2) / (3 * c) ?$

ЛАБОРАТОРНАЯ РАБОТА №3

Задания на применение операторов ветвления, цикла и выбора

1. Напишите программу, которая проверяет, являются ли все три вещественных числа, введенных пользователем с клавиатуры, положительными и выводит соответствующее сообщение (например, «Не все числа положительны»).

2. Заданы три вещественных числа. Напишите программу, которая проверяет, существует ли треугольник, длинами сторон которого являются эти числа, и выводит соответствующее сообщение.

3. Заданы вершины треугольника (своими координатами на плоскости): $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$. Составьте программу, которая определяет тип треугольника (тупоугольный, прямоугольный или остроугольный), и выводит соответствующее сообщение на экран.

4. Напишите программу, которая вычисляет первые n членов геометрической прогрессии, заданной рекуррентным соотношением $b_k = b_{k-1} \cdot q$, где q - знаменатель прогрессии; при условии, что $b_1 = 2$, $b_2 = 6$.

5. Напишите программу, которая вычисляет сумму

$$S = \frac{1}{\sqrt{1 \cdot 3}} + \frac{1}{\sqrt{3 \cdot 5}} + \dots + \frac{1}{\sqrt{(n-2) \cdot n}},$$

где n – нечетное число, большее 1. Число n запрашивается у пользователя, при неверном вводе n предлагается ввести его снова до тех пор, пока число не будет соответствовать условиям.

6. Напишите программу, которая определяет тип введенного пользователем с клавиатуры символа: *латинская буква, цифра, русская буква, знак препинания, прочие символы*. На экран выводится соответствующее сообщение.

7. Напишите программу, которая запрашивает имя пользователя, а затем выводит на экран приветствие в рамочке, так чтобы размер рамочки зависел от длины введенного имени.

Примеры:

* Добрый день, Андрей! *

* Добрый день, Пётр Иванович! *

ЛАБОРАТОРНАЯ РАБОТА №4

Задания на составление подпрограмм, использование статических массивов

Указание: все входные данные подпрограммы должны получать через список параметров. Внутри подпрограмм не должно быть ввода с клавиатуры и вывода на экран, весь ввод-вывод должен быть в тестирующих программах.

1. Напишите программу, которая создает одномерный статический массив из 10 элементов, заполняет его случайными вещественными числами и находит наименьший элемент массива.

2. Составьте функцию, которая вычисляет среднее арифметическое элементов массива. Входные параметры: одномерный статический массив, количество элементов массива (передается по значению) и переменную для записи результата (передается по ссылке). Функция возвращает 0, в случае успешного выполнения, и 1, если размер массива оказался некорректным.

3. Напишите программу, которая создает одномерный статический массив из 20 элементов и заполняет его случайными вещественными числами. Вызывает функцию из предыдущего задания. Выводит на экран среднее арифметическое, если работа вызываемой функции завершена корректно, в противном случае выводит сообщение об ошибке.

4. Составьте подпрограмму, которая заполняет элементы двумерного статического массива так, чтобы каждый элемент a_{ij} стал равен $i+j$.

5. Напишите программу, которая создает двумерный статический массив указанного пользователем размера (запрашивать размеры до тех пор, пока не будут введены размеры, не превышающие указанные при создании массива). Вызывает подпрограмму из предыдущего задания. Выводит полученный массив на экран, если работа вызываемой функции завершена корректно, в противном случае выводит сообщение об ошибке.

ЛАБОРАТОРНАЯ РАБОТА №5

Задания на использование текстовых файлов, указателей и динамической памяти

1. Напишите программу, которая запрашивает у пользователя количество чисел и границы диапазона значений для генерации случайных чисел; создает текстовый файл и заполняет его псевдослучайными числами.

2. Напишите программу, которая создает двумерный статический массив и заполняет его из текстового файла, читая последовательно элементы первой строки, потом второй и т.д. Если элементов в файле не хватает, то выдается сообщение об ошибке. Файл создается с помощью программы из предыдущего задания.

3. Модифицируйте программу из предыдущего задания так, чтобы сначала читались из файла число строк массива, затем число столбцов и, наконец, все элементы массива построчно. Размеры, прочитанные из файла не должны превышать размеры массива, указанные при его создании, в противном случае выдается сообщение об ошибке.

4. Напишите программу, которая создает одномерный динамический массив, прочитав из файла его размер, а затем заполнит его, читая из того же файла элементы.

5. Добавьте в программу из предыдущего задания функцию, которая получает в качестве входного параметра указатель на динамический одномерный массив и число его элементов и находит максимальный элемент этого массива. Найденный элемент передается как выходной параметр по ссылке. Выполняется проверка входных данных на допустимость, и соответствующий код завершения функция возвращает через основной выход.

6. Напишите программу, которая создает двумерный динамический массив и заполняет его случайными числами.

7. Добавьте в программу из предыдущего задания подпрограмму, которая получает в качестве входного параметра указатель на двумерный массив, количество строк и столбцов, а также номера строк, которые необходимо поменять местами. Функция меняет местами указанные строки.

ЛАБОРАТОРНАЯ РАБОТА №6

Задания на использование текстовых и двоичных файлов

1. Даны два файла in.txt, в котором содержатся целые числа и пустой файл out.txt. Напишите программу, которая переписывает из файла in.txt в файл out.txt все кратные трем числа.
2. Даны два файла с вещественными числами. Напишите программу, которая переписывает в новый файл сначала все отрицательные числа из первого файла, затем все положительные из второго.
3. Дан файл символов in1.txt. Напишите программу, которая переписывает в файл out1.txt все буквы и знаки препинания из файла in1.txt, а в файл out2.txt – все цифры.
4. Напишите программу, которая создает двоичный файл и записывает в него все степени числа 2 до степени n, которую указал пользователь. А затем выводит на экран все числа из файла. Можно использовать одну рабочую переменную, без использования массива.
5. Напишите программу, которая создает двоичный файл, состоящий из n вещественных чисел. Создает двумерный динамический массив, запросив размеры у пользователя. Заполняет массив числами из созданного файла, выдавая сообщение об ошибке, если элементов в файле недостаточно.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №1

Цель работы. Приобрести навыки составления программ на языке программирования разработки интерфейса «человек-компьютер» при вводе информации и выходных форм для результатов её обработки.

Содержание задания. Составить программу табулирования функции одной вещественной переменной на заданном отрезке с заданным шагом при некотором допустимом значении параметра, т.е. требуется вывести на экран таблицу значений функции. При этом точки, в которых будут вычисляться значения функции, расположены на одинаковом расстоянии друг от друга.

Рекомендации к программе:

1. Ввод исходного значения параметра должен сопровождаться контролем его допустимости. Форматы вывода чисел должны быть выбраны так, чтобы результаты могли быть отражены независимо от значений параметра, т.е. рекомендуется вывод значений функции в научном (плавающем формате).

2. Для простоты лучше заполнять таблицу, последовательно выводя на экран строки, а не столбцы.

3. Результаты табулирования должны быть представлены в виде набора страниц, каждая из которых должна содержать номер страницы, заголовок таблицы с введенным значением параметра и таблицу из нескольких строк и колонок для отображения значений аргумента функции и результата ее вычисления.

4. Для повышения гибкости программы рекомендуется в число условно-постоянных величин включить границы области допустимости параметра, начальную и конечные точки отрезка и шаг изменения аргумента, а также число строк и колонок таблицы, ширина полей для вывода значений аргумента и результата вычислений и т.п. Допускается включение условно-постоянных величин в раздел констант либо их чтение из конфигурационного файла.

5. Если вычисление функции не оформлено в виде подпрограммы, а непосредственно встроено в тело цикла табулирования, то рекомендуется провести чистку цикла, т.е. вычисление компонент функции, не зависящих от значения аргумента, производить вне цикла.

Пример выходной формы (последняя страница таблицы)

Экран № 4

Таблица значений функции
При параметре 2.65676E+00

!	-----	!	-----	!	-----	!	-----	!
!	Аргумент	!	Функция	!	Аргумент	!	Функция	!
!	-----	!	-----	!	-----	!	-----	!
!	1.5000	!	1.8793E-01	!	1.5300	!	1.0836E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.5600	!	2.8683E-02	!	1.5900	!	-5.1016E-02	!
!	-----	!	-----	!	-----	!	-----	!
!	1.6200	!	-1.3067E-00	!	1.6500	!	-2.1020E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.6800	!	-2.8955E-01	!	1.7100	!	-3.6864E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.7400	!	-4.4739E-01	!	1.7700	!	-5.2574E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.8000	!	-6.0362E-01	!	1.8300	!	-6.8096E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.8600	!	-7.5768E-01	!	1.8900	!	-8.3372E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.9200	!	-9.0901E-01	!	1.9500	!	-9.8348E-01	!
!	-----	!	-----	!	-----	!	-----	!
!	1.9800	!	-1.0571E+00	!	2.0100	!	-1.1297E+00	!
!	-----	!	-----	!	-----	!	-----	!
!	2.0400	!	-1.2013E+00	!	2.0700	!	-1.2719E+00	!
!	-----	!	-----	!	-----	!	-----	!
!	2.1000	!	-1.3413E+00	!		!		!
!	-----	!	-----	!	-----	!	-----	!

Для продолжения нажмите ENTER

Требования к программе

Основные:

1. При табулировании не следует заносить значения функции в массив; таблицу следует выводить в текстовом режиме при этом лучше обойтись самыми простыми символами для обозначения границ, без использования псевдографики и разных цветов (см. представленный выше пример).

2. К таблице на экране должны быть все необходимые пояснения: текущее значение параметра p , номер страницы, приглашение нажать клавишу для продолжения и т.д. (см. пример). При предложении пользователю ввести значение какой-либо величины должны быть даны все необходимые пояснения о ее назначении и допустимых значениях.

3. В качестве констант (const) должны быть определены все параметры табулирования:

- границы отрезка табулирования;
- границы допустимости параметра;
- шаг табулирования (расстояние между точками, в которых вычисляется значение функции);
- число строк и число столбцов таблицы (один столбец включает в себя и значения аргумента, и значения функции, например, в таблице, которая показана в примере, – два столбца);
- ширина вывода значения аргумента (с фиксированной точкой, т.е. должно быть два числа – общая ширина и число знаков после точки);
- ширина вывода значения функции (с экспоненциальной частью).

Таким образом эти параметры должны фигурировать в программе не в виде конкретных чисел, а в виде буквенных обозначений, значение которых определяется в разделе описаний и может быть изменено перед очередным запуском программы. При этом проверка допустимости этих значений не выполняется, а в разделе описаний есть комментарий, описывающий область допустимости этих значений. При неправильном задании констант, программа может работать некорректно.

4. Должна проверяться правильность ввода параметра p и в случае неверного ввода предлагаться повторный ввод.

Дополнительные:

5. Должна быть дана возможность переключения типа вывода значения функции – научный или с фиксированной точкой (тогда для числа знаков после точки тоже должна быть константа).

6. Значения табулируемой функции, т.е. $f(x)$, вычисляются в подпрограмме, которая вызывается при построении таблицы.

7. Весь процесс табулирования, т.е. весь вывод таблиц от начала и до конца, осуществляется в подпрограмме, которая все параметры табулирования (см. п. 3) получает в качестве входных параметров (даже если они описаны как

константы) и не использует никаких глобальных переменных.

8. Выполняется проверка согласованности констант, т.е. проверяется войдет ли таблица заказанных размеров на экран, поместятся ли числа в столбцах заказанной ширины и т.д.

9. Должна быть учтена возможность табулирования как от начала отрезка к концу, так и от конца к началу. При этом изменение направления должно производиться путем изменения минимального количества величин, а текст программы не должен расширяться большим числом проверок.

10. Параметры табулирования хранятся в конфигурационном файле. Заполнять этот файл лучше из отдельной программы и при заполнении проверять согласованность параметров, например, как в п.8, или сразу предлагать диапазон для всех величин, не разрешая из него выходить. При этом проверка параметров в подпрограмме табулирования все равно должна быть!

11. Проверяется корректность ввода имени файла (т.е. программа не пытается читать из того файла, которого нет), при чтении из файла проверяется корректность данных (в смысле, являются ли они числами в нужном формате).

Варианты задания №1

Ниже использованы обозначения: x - аргумент функции $f(x)$; h - шаг табулирования; p - параметр, выбираемый пользователем при выполнении расчета.

1. Протабулировать

$$f(x) = \begin{cases} \lg(1 + |px|) & \text{при } x^9 \leq e^p \\ \frac{\pi - \operatorname{arctg}^2 px}{\sqrt{x}} & \text{при } x^9 > e^p \end{cases}$$

на $[-2.4, 2.4]$ с шагом $h = 0.03$; $p \in [-2, 2]$.

2. Протабулировать

$$f(x) = \begin{cases} \lg \sqrt{(x-p)^2 + p^4} & \text{при } x < \operatorname{arctg}(p) \\ \pi + \left\lfloor \frac{x}{p} \right\rfloor & \text{при } x \geq \operatorname{arctg}(p) \\ pe^{0.5x} & \end{cases}$$

на $[-2, 4]$ с шагом $h = 0.02$; $p \in (-\infty, -0.5] \cup [0.5, \infty)$.

3. Протабулировать

$$f(x) = \begin{cases} \sqrt{0.25 + e^{px}} & \text{при } x^7 < |p| \\ \lg \frac{x}{\pi + \operatorname{arctg}^2 px} & \text{при } x^7 \geq |p| \end{cases}$$

на $[-2, 6]$ с шагом $h = 0.05$; $p \in [-3, 3]$.

4. Протабулировать

$$f(x) = \begin{cases} p \cdot \cos \frac{x^4}{4+p} & \text{при } x \leq \arctg(p) \\ \lg(\pi + e^{|px|}) & \text{при } x > \arctg(p) \end{cases}$$

на $[-4, 2]$ с шагом $h = 0.04$; $|p| \leq 2.5$.

5. Протабулировать

$$f(x) = \begin{cases} \frac{\sqrt[3]{1+x^2 p^2}}{2\pi - x} & \text{при } x \leq \sin p \\ \ln(e^{|px|} + 1) & \text{при } x > \sin p \end{cases}$$

на $[-4, 2]$ с шагом $h = 0.03$; $p \in [-2, 2]$.

6. Протабулировать

$$f(x) = \begin{cases} \lg^5(2\pi + x^2 p^2) & \text{при } x < \arctg(p) \\ \frac{\cos(x^2 + p)}{1 + |px|} & \text{при } x \geq \arctg(p) \end{cases}$$

на $[-2, 3]$ с шагом $h = 0.04$; $|p| \leq 4$.

7. Протабулировать

$$f(x) = \begin{cases} \sqrt[5]{1+4p^2} \cdot \arctg^2 px & \text{при } x < \sqrt{p} \\ \lg(1 + |px|) - \pi & \text{при } x \geq \sqrt{p} \end{cases}$$

на $[-2.4, 2.6]$ с шагом $h = 0.025$; $p \in [0, 3]$.

8. Протабулировать

$$f(x) = \begin{cases} \sqrt[3]{\frac{x^4 + 1}{0.2 + p^2 x^2}} & \text{при } x \geq \arctg(p - 2) \\ \lg(2\pi + e^{|px|}) & \text{при } x < \arctg(p - 2) \end{cases}$$

на $[-2.7, 2.3]$ с шагом $h = 0.05$; $|p| \leq 3$.

9. Протабулировать

$$f(x) = \begin{cases} \frac{\sqrt[5]{1 + x^2 p^4}}{2\pi - x} & \text{при } x \leq \arctg(p + x) \\ \lg(1 + e^{|px|}) & \text{при } x > \arctg(p + x) \end{cases}$$

на $[-2.7, 2.3]$ с шагом $h = 0.025$; $|p| \leq 3$.

10. Протабулировать

$$f(x) = \begin{cases} \frac{\sqrt[3]{1 + x^4 p^4}}{2\pi - x} & \text{при } x \leq \sin p \\ \lg^5(e^{|px|} + 1) & \text{при } x > \sin p \end{cases}$$

на $[-4, 2]$ с шагом $h = 0.03$; $p \in [-3, 3]$.

11. Протабулировать

$$f(x) = \begin{cases} \ln(\pi^2 + e^{|px|}) & \text{при } x < \arctg(p - x) \\ \sqrt{\frac{x^2 + 1}{0.25 + p^4 x^4}} & \text{при } x \geq \arctg(p - x) \end{cases}$$

на $[-2.3, 2.7]$ с шагом $h = 0.05$; $|p| \leq 5$.

12. Протабулировать

$$f(x) = \begin{cases} \frac{1 - \operatorname{arctg}^4 px}{\sqrt{x}} & \text{при } x^5 > e^{|px|} \\ \ln(\pi + |px|) & \text{при } x^5 \leq e^{|px|} \end{cases}$$

на $[-3, 1.5]$ с шагом $h = 0.025$; $|p| \leq 3$.

13. Протабулировать

$$f(x) = \begin{cases} \lg^3(\pi + x^2 p^2) & \text{при } x > \operatorname{arctg}(p) \\ p + \sqrt{\frac{x^2}{1 + p^2 + x^4}} & \text{при } x \leq \operatorname{arctg}(p) \end{cases}$$

на $[-3, 3]$ с шагом $h = 0.05$; $|p| \leq 10$.

14. Протабулировать

$$f(x) = \begin{cases} \sqrt[3]{2x + |p|} & \text{при } x^7 > e^{1/p} \\ \lg \frac{\pi + x^2 p^2}{\operatorname{arctg}(p^2)} & \text{при } x^7 \leq e^{1/p} \end{cases}$$

на $[-2.7, 2.3]$ с шагом $h = 0.025$; $|p| > 1$.

15. Протабулировать

$$f(x) = \begin{cases} \lg^2(1 + |px|) & \text{при } x > \operatorname{arctg}(p) \\ \cos \frac{x^2}{4 + p} & \text{при } x \leq \operatorname{arctg}(p) \\ \sqrt[3]{\pi + x^2} & \end{cases}$$

на $[-3, 2]$ с шагом $h = 0.04$; $|p| \leq 5$.

16. Протабулировать

$$f(x) = \begin{cases} \lg \frac{2\pi + |px|}{\operatorname{arctg}(p^2)} & \text{при } x^9 < e^{1/p} \\ \sqrt{2x + p^2} & \text{при } x^9 \geq e^{1/p} \end{cases}$$

на $[-2.7, 2.3]$ с шагом $h = 0.025$; $p \in (-\infty, -0.5) \cup (1, \infty)$.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №2

Цель работы. Приобрести и развить навыки составления подпрограмм и их локального тестирования. Изучить способы передачи параметров, включая передачу многомерных массивов неопределенных размеров. Изучить методы конструирования циклов с использованием расширений неиндуктивных функций, инварианта и других методов.

Содержание задания. Составить подпрограмму обработки прямоугольной таблицы вещественных чисел. Подпрограмма должна реализовывать вычислительный алгоритм, который получает саму таблицу и ее размеры (т.е. количество строк и столбцов) в качестве входных параметров, а в качестве выходного параметра – результат вычислений. Кроме того, написать программу, обеспечивающую локальное тестирование разработанной подпрограммы.

Рекомендации к программе:

1. Подпрограмма должна быть составлена так, чтобы в одной программе ее можно было использовать для обработки нескольких таблиц различных размеров. При этом, конечно, предполагается один и тот же метод хранения табличных данных в памяти. При выполнении данного задания рекомендуются следующие способы реализации табличных данных:

– для представления таблицы используется многомерный массив достаточно больших размеров, в котором данные занимают часть выделенной памяти (автоматический класс памяти).

– двумерная таблица может рассматриваться как совокупность строк или столбцов. Для каждой подструктуры память выделяется динамически, а указатели также образуют динамически выделяемый массив.

2. Алгоритм обработки должен быть экономичным и достаточно гибким. Следует, насколько это возможно, избегать использования рабочих объектов, требующих больших ресурсов памяти.

3. Интерфейс подпрограммы с вызывающей программой должен быть реализован только через список параметров. Для удобства вызова подпрограммы входные скалярные параметры должны передаваться только значениями, даже если они не изменяются ею. Реализация подпрограммы в виде функции допускается только при отсутствии побочного эффекта. Все вспомогательные величины следует локализовать в теле подпрограммы. Подпрограмма обработки данных должна быть свободна от операций ввода-вывода. Сообщения об ошибках следует передавать через значения кода завершения, который допускается возвращать как значение функции.

4. Тестирование следует проводить достаточно тщательно, в том числе и на «вырожденных» данных. При подборе тестов необходимо учитывать структуру применяемых алгоритмов и данных.

Требования к программе

Основные:

1. Таблица хранится в виде двумерного статического массива достаточно больших размеров. Определен тип «таблица» (это нужно для того, чтобы передать таблицу в подпрограмму). Максимальные индексы массива, используемые при описании типа, хранятся в виде констант.

2. Основная подпрограмма не содержит операций ввода-вывода. Параметры подпрограммы:

- таблица (по ссылке);
- фактические размеры таблицы (по значению);
- основной результат (по ссылке).

Подпрограмма содержит описание спецификации (в виде комментария под заголовком).

3. Работа тестирующей программы организована в виде простейшего меню, позволяющего создавать таблицу, показывать таблицу на экране, выполнять ее обработку, показывать результат и заканчивать работу.

4. Фактические размеры таблицы указываются пользователем. Заполнение таблицы осуществляется двумя способами (по выбору пользователя) с клавиатуры (простейший последовательный ввод элементов) и случайным образом.

5. Алгоритм реализован без создания вспомогательной таблицы. При необходимости можно пользоваться вспомогательным одномерным массивом, число элементов которого не должно совпадать с числом элементов таблицы. Это может быть вспомогательная строка и/или вспомогательный столбец, в зависимости от задания.

6. В подпрограмме выполняется проверка входных параметров на допустимость. Подпрограмма возвращает код завершения по ссылке или в качестве значения функции, который показывает, правильно ли выполнялась подпрограмма.

Дополнительные:

7. Возможность заполнения таблицы из текстового (или бинарного) файла с проверкой корректности имени файла и его содержимого. При этом размеры таблицы хранятся в начале того же файла.

8. Делается проверка корректности ввода чисел, т.е. защита от ввода вместо чисел посторонних символов.

9. Таблица реализована как двумерный динамический массив с использованием механизма указателей.

Варианты задания №2

1. Переставить столбцы таблицы так, чтобы в заданной строке сначала стояли отрицательные, потом нули, потом положительные числа. Порядок их следования сохранить.
2. Подсчитать, сколько раз за нулем следует положительное число, если обходить таблицу по спирали против часовой стрелки, начиная с нижнего правого угла.
3. Изменить таблицу, заменив каждый ее элемент на максимальный среди исходных элементов, лежащих не выше и не левее его.
4. Изменить таблицу, заменив каждый ее элемент на сумму исходных элементов, лежащих не выше и не правее его.
5. Найти номер первой из строк таблицы, сумма абсолютных величин элементов в которых максимальна.
6. Среди сумм элементов таблицы, расположенных на пересечении строк и столбцов, разность номеров которых одинакова, найти минимальное значение.
7. Произвести циклический сдвиг столбцов таблицы на заданное число позиций.
8. Найти скалярные произведения всех строк таблицы на последнюю из строк, имеющих наибольшее количество отрицательных элементов.
9. Найти максимальный среди элементов тех строк таблицы, которые упорядочены по неубыванию.
10. Найти номер первого столбца таблицы, для которого скалярное произведение на заданный вектор минимально.
11. Найти скалярные произведения всех столбцов таблицы на последний из столбцов, имеющих наибольшую сумму элементов.
12. Найти минимальный среди максимальных элементов столбцов таблицы.
13. В строках таблицы, содержащих наибольшее количество нулей, найти минимальную в строке сумму абсолютных величин.
14. Среди элементов заданной строки таблицы найти самый левый из тех, которые принадлежат наибольшему количеству столбцов.
15. Изменить таблицу, заменив нулями элементы тех строк и столбцов, которые содержат отрицательные числа.
16. Подсчитать число столбцов таблицы, в каждом из которых все числа различны.

ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ №3

Цель работы. Приобрести и развить навыки построения табличных структур данных на основе использования динамической памяти. Закрепить опыт конструирования алгоритмов методами расширения индуктивной функции, рекурсии, инварианта цикла и т.п. Развить навыки интерфейса «человек- компьютер».

Содержание задания. Требуется написать подпрограмму обработки прямоугольной таблицы вещественных чисел, которая хранится в памяти как двумерный динамический массив. Подпрограмма должна реализовывать вычислительный алгоритм, который получает указатель на таблицу и ее размеры (т.е. количество строк и столбцов) в качестве входных параметров, а в качестве выходного параметра – результат вычислений (если он есть). Кроме того, должна быть написана тестирующая программа, которая позволяет создать таблицу заказанных пользователем размеров, вызвать подпрограмму обработки и вывести на экран результат вычислений.

Рекомендации к программе:

1. Реализация алгоритма должна быть максимально эффективна. Следует, насколько это возможно, не использовать рабочих объектов, требующих больших ресурсов памяти.

2. Подпрограмма должна обрабатывать ошибки входной информации, адекватно реагировать на отсутствие требуемого решения, а возможно, и на его неоднозначность. В связи с тем, что в используемом языке программирования для этого нет специальных средств, рекомендуем для этих целей возвращаемый код признака результата.

3. Следует подготовить несколько файлов для наиболее полного тестирования программы, в том числе на «вырожденных» данных.

4. Основная программа может быть составлена так, чтобы одновременно можно было хранить несколько таблиц различных размеров. При этом, конечно, предполагается один и тот же метод хранения табличных данных в памяти. Для выполнения данного задания рекомендуются организовать структуры, содержащие поля: число строк, число столбцов, указатель на таблицу. Возможны следующие способы реализации:

- создать динамический массив из этих структур;
- организовать список из этих структур.

Требования к программе

Основные:

1. Данные хранятся в виде динамической таблицы, организованной с помощью столбца указателей на строки (или строки указателей на столбцы – в зависимости от алгоритма). Описаны типы «указатель на таблицу» и «указатель на строку (столбец) таблицы».

2. Основные вычисления оформлены в виде подпрограммы, не содержащей операций ввода-вывода. Параметры подпрограммы: указатель на таблицу, фактические размеры таблицы (по значению), основной результат – по ссылке, код завершения по ссылке или в качестве основного выхода функции. Подпрограмма содержит описание спецификации (в виде комментария под заголовком).

3. Фактические размеры таблицы указываются пользователем. Заполнение таблицы осуществляется тремя способами (по выбору пользователя) с клавиатуры (простейший последовательный ввод элементов), случайным образом и из текстового файла (с проверкой существования файла).

4. Работа тестирующей программы организована в виде простейшего меню, позволяющего создавать таблицу, выполнять ее обработку, показывать результат, сохранять таблицу в файл и заканчивать работу. Предполагается, что можно последовательно обработать несколько таблиц, не выходя из программы. При создании новой таблицы старая уничтожается.

5. Алгоритм реализован без создания вспомогательной таблицы.

6. В подпрограмме выполняется проверка входных параметров на допустимость.

7. Динамическая память для таблиц выделяется и освобождается корректно.

Дополнительные:

8. Проверка корректности ввода чисел (т.е. отсутствие фатального завершения программы при вводе недопустимых символов в числах).

9. При вводе таблицы из файла предполагается бинарный файл, а не текстовый.

10. Все пункты меню представляют собой вызовы подпрограмм (например, для создания таблицы в динамической памяти или заполнения уже созданной таблицы случайными числами), оформленных по всем правилам (т.е. все объекты с которыми подпрограмма работает, должны передаваться через список формальных параметров).

11. Возможность работы с несколькими таблицами, указатели на которые хранятся в массиве. В меню добавляется выбор текущей таблицы (т.е. все действия выполняются с той таблицей, которая является текущей в данный момент). Если массив заполняется, то таблицы больше не создаются.

12. Расширение предыдущего пункта возможностью создания произвольного числа таблиц (пока памяти хватает).

13. Перед уничтожением таблицы (в том числе при выходе из программы) предлагается сохранить ее в файл.

Варианты задания №3

1. Отыскать седловую точку таблицы, то есть положение и значение элемента, который является минимальным в строке и максимальным в столбце.

2. В таблице с монотонно возрастающими элементами в строках и столбцах определить положение заданного числа.

3. Найти элемент таблицы, встречающийся в каждой строке и каждом столбце.

4. Среди чисел заданной строки таблицы найти такое, которое принадлежит наибольшему количеству столбцов таблицы.

5. Подсчитать количество столбцов таблицы, каждый из которых составлен из попарно различных элементов.

6. Найти максимальный элемент таблицы, встречающийся более одного раза в каждой строке.

7. Найти минимальный элемент таблицы, встречающийся менее трех раз в каждом столбце.

8. Найти строку таблицы, содержащую самую длинную серию строгих локальных экстремумов таблицы.

9. Каждый элемент таблицы заменить на максимальный элемент строки и столбца, на пересечении которых он находится.

10. Каждый элемент таблицы заменить на сумму элементов всех строк и столбцов, кроме тех, на пересечении которых он находится.

11. В таблице обнулить элементы, стоящие в строках или столбцах, содержащих нули.

12. Найти две строки таблицы, скалярное произведение которых максимально.

13. Вычислить максимальную среди сумм абсолютных величин элементов строк таблицы и номер строки, на которых максимум достигается.

14. Среди минимальных элементов строк таблицы найти наибольший и его положение.

15. На пересечении строк и столбцов, сумма элементов которых положительна, найти максимальный элемент и его положение.

16. Найти значение и положение в таблице элемента, для которого сумма элементов, лежащих не выше и не левее его, максимальна.

ЛИТЕРАТУРА

Вирт Н. Алгоритмы + структуры данных = программы / пер. с англ. – М.:Мир, 1985. – 406 с.

Кнут Д.Э. Искусство программирования. Т. 3. Сортировка и поиск / пер. с англ. – 2-е изд. – М.: Вильямс, 2000. – 832 с.

Конова Е. А., Поллак Г. А. Алгоритмы и программы. Язык С++: Учебное пособие. — 2е изд., стер. — СПб.: Издательство «Лань», 2017. —384с:

Кушниренко А.Г., Лебедев Г.В. Программирование для математиков: учеб. Пособие для вузов. – М.:Наука, 1988. – 384 с.

Лафоре Р. Объектно-ориентированное программирование в С++. – 4-е изд., СПб.: Питер, 2004. – 922с.

Методические указания и задания по вычислительной практике: в 2 ч. Ч. I / сост. Л.В. Камаева, С.Е. Макаров, С.А. Терентьев, А.С. Толстуха. – Омск: Изд-во Омск. Гос. Ун-та, 1998. – 32 с.

Методические указания и задания по вычислительной практике: в 2 ч. Ч. II / сост. Л.В. Камаева, С.Е. Макаров, С.А. Терентьев, А.С. Толстуха. – Омск: Изд-во Омск. Гос. Ун-та, 1998. – 32 с.

Прата С. Язык программирования С. Лекции и упражнения, 6-е изд.: Пер. с англ. – М.: ООО «И.Д. Вильямс», 2015. – 928 с.

Страуструп Б. Язык программирования С++. – М.: Бином, 2015. – 1166 с.

Учебная практика на ЭВМ: методические указания и задания: в 2 ч. Ч. I / [сост. Н.С. Бахта, Н.П. Тыркова]. – Омск: Изд-во Омск. Гос. Ун-та, 2013. – 30 с.