

Учреждение образования
«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Н. Н. Пустовалова, Н. В. Пацей

ОСНОВЫ АЛГОРИТМИЗАЦИИ И ПРОГРАММИРОВАНИЯ
ЛАБОРАТОРНЫЙ ПРАКТИКУМ

1 семестр

Минск 2016

ПРЕДИСЛОВИЕ

Практикум содержит задания для выполнения лабораторных работ по дисциплине «Основы алгоритмизации и программирования». Основным предметом изучения является язык программирования C++, программы на котором выполняются на базе приложения **Microsoft Visual Studio**.

В каждой работе имеются краткие теоретические сведения по рассматриваемым вопросам. Преподаватель определяет, какие лабораторные работы должны выполнять студенты и в каком объеме. Предполагается, что выполнение большинства лабораторных работ занимает у студентов два академических часа.

При выполнении программ на компьютере возможны ситуации, когда возникают ошибки из-за того, что исходные данные заданы некорректно (деление на ноль, корень из отрицательного числа и т. п.). В таком случае надо ознакомиться с появившимися сообщениями об ошибках, проанализировать программу и выполнить с другими исходными данными.

В тексте данного практикума имеются кнопки, при нажатии на которые открываются тесты, предназначенные для контроля знаний студентов. Тестирование происходит по команде преподавателя и занимает несколько минут. Для работы тестирующих программ предварительно в приложении Word надо разрешить использование макросов. При этом тексты ответов на формах располагаются каждый раз случайным образом, и ответить на вопросы можно только один раз, так как после нажатия на кнопку «Результаты» форма с вопросами и вариантами ответов исчезает.

Для **оформления отчетов по лабораторным работам** используется приложение **Word**. Каждая работа должна содержать название работы, условия задач в соответствии со своим вариантом, алгоритмы, тексты разработанных программ, результаты. В верхнем колонтитуле записывается фамилия студента и номер группы, в нижнем – номера страниц. Все работы сохраняются в одном файле.

ОГЛАВЛЕНИЕ

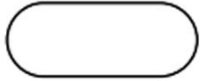
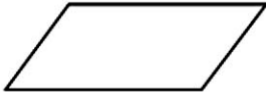
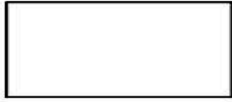





- Лабораторная работа № 1. [Способы представления алгоритмов](#)
- Лабораторная работа № 2. [Основные элементы языка C++. MS Visual Studio 2010](#)
- Лабораторная работа № 3. [Ввод и вывод информации](#)
- Лабораторная работа № 4. [Разветвляющиеся программы](#)
- Лабораторная работа № 5. [Циклические программы](#)
- Лабораторная работа № 6. [Отладка программ](#)
- Лабораторная работа № 7. [Вычисление сумм, произведений, экстремумов](#)
- Лабораторная работа № 8. [Решение инженерных задач на основе циклических программ](#)
- Лабораторная работа № 9. [Одномерные массивы](#)
- Лабораторная работа № 10. [Битовые операции](#)
- Лабораторная работа № 11. [Указатели и ссылки](#)
- Лабораторная работа № 12. [Обработка символьной информации](#)
- Лабораторная работа № 13. [Многомерные массивы](#)
- Лабораторная работа № 14. [Динамические массивы и функции пользователя](#)
- Лабораторная работа № 15. [Указатели как параметры и результаты функции](#)
- Лабораторная работа № 16. [Массивы и ссылки при работе с функциями](#)

Лабораторная работа № 1. Способы представления алгоритмов

Алгоритм – это система правил, определяющая последовательность действий над некоторыми объектами и приводящая к достижению поставленной цели после конечного числа шагов.

Для записи алгоритма решения задачи применяется:

- словесно-формульное описание;
- блок-схема (отдельные блоки представлены ниже).

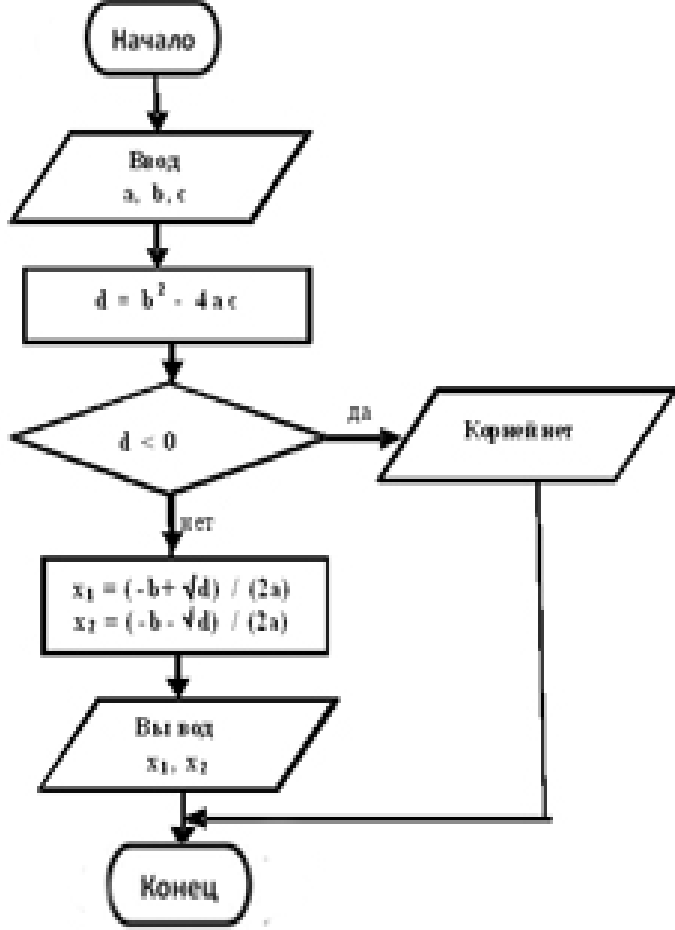
 Блок начала-конца алгоритма	 Блок ввода-вывода данных	 Блок вычислений	 Условный блок
 Предопределенный процесс	 Счетный цикл	 Комментарий	 Соединитель (используется при разрыве схемы)

Различают алгоритмы линейные, разветвляющиеся, циклические.

Линейный алгоритм – это такой алгоритм, в котором все действия выполняются последовательно друг за другом и только один раз. Схема представляет собой последовательность блоков, которые располагаются сверху вниз в порядке их выполнения.

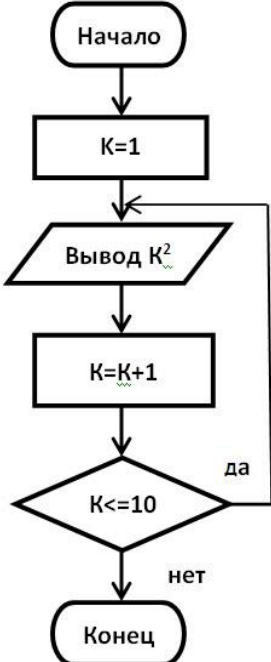

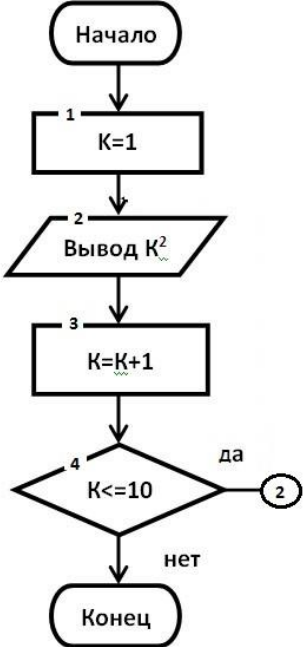
С помощью алгоритмов *разветвляющейся* структуры можно описать задачи, в которых выбор направления продолжения вычисления осуществляется по итогам проверки заданного условия.

Для решения многих задач характерно многократное повторение отдельных участков вычислений. Для решения таких задач применяются алгоритмы *циклической* структуры.

Словесно-формульное описание алгоритма решения квадратного уравнения	Блок-схема алгоритма решения квадратного уравнения
<p>1. Ввести числовые значения переменных a, b и c.</p> <p>2. Вычислить d по формуле $d = b^2 - 4ac$.</p> <p>3. Если $d < 0$, то перейти к п. 4, иначе перейти к п. 5.</p> <p>4. Напечатать сообщение «Корней нет» и перейти к п. 7</p> <p>5. Вычислить:</p> $x_1 = \frac{-b + \sqrt{d}}{2a}, \quad x_2 = \frac{-b - \sqrt{d}}{2a}$ <p>6. Напечатать значения x₁ и x₂</p> <p>7. Конец вычислений.</p>	 <pre> graph TD Start([Начало]) --> Input[/Ввод a, b, c/] Input --> CalcD[d = b² - 4ac] CalcD --> Decision{d < 0} Decision -- да --> NoRoot[/Корней нет/] Decision -- нет --> CalcX["x₁ = (-b + √d) / (2a) x₂ = (-b - √d) / (2a)"] CalcX --> Output[/Вывод x₁, x₂/] NoRoot --> End([Конец]) Output --> End </pre>

Пример. Словесно-формульное описание алгоритма вычисления квадратов значений K , изменяющихся от 1 до 10.

1. Определить значение $K = 1$.
2. Вычислить K^2 и вывести.
3. Увеличить значение K на единицу.
4. Если $K \leq 10$, то перейти к п. 2. Иначе перейти к п. 5.
5. Конец вычислений.

Первый вариант блок-схемы алгоритма вычисления квадратов значений K	Второй вариант блок-схемы алгоритма вычисления квадратов значений K	Третий вариант блок-схемы алгоритма вычисления квадратов значений K (нет кружка 2 для входа)
 <pre> graph TD Start([Начало]) --> K1[K=1] K1 --> Out1[/Вывод K^2/] Out1 --> Kinc[K=K+1] Kinc --> Dec1{K<=10} Dec1 -- да --> Out1 Dec1 -- нет --> End1([Конец]) </pre>	 <pre> graph TD Start([Начало]) --> Dec2{K=1,10} Dec2 -- да --> Out2[/Вывод K^2/] Out2 --> End2([Конец]) Dec2 -- нет --> Dec2 </pre>	 <pre> graph TD Start([Начало]) --> S1[K=1] S1 --> S2[/Вывод K^2/] S2 --> S3[K=K+1] S3 --> Dec3{K<=10} Dec3 -- да --> S2 Dec3 -- нет --> End3([Конец]) </pre>

Задание

1. В приложении Word разработать словесно-формульное описание и блок-схему алгоритма в соответствии со своим вариантом для решения задачи из таблицы. Номер варианта определяет преподаватель.

Все блоки схемы должны иметь одинаковую ширину, кроме блоков начала и конца, которые имеют меньшие размеры. В схеме использовать нумерацию блоков (в разрыве верхней линии слева) и соединители.

№ варианта	Условие задачи
1	Даны действительные числа x, y, z . Поменять значения переменных так, чтобы $x \geq y \geq z$.
2	Даны действительные числа $x1, y1, x2$. Вычислить $\max(x1, y1, x2)$.
3	Даны действительные числа x, y, z . Вычислить $\max(x + y + z, xuz) * 3$.
4	Определить, есть ли среди заданных целых чисел A, B, C два чётных.
5	Даны действительные числа x, y, z . Поменять значения переменных так, чтобы $x \leq y \leq z$.
6	Среди заданных целых чисел k, f, t найти пары кратных.
7	Даны числа a, b, c . Если $a < b < c$, то каждое число заменить наибольшим, если $a > b > c$, то числа оставить без изменений, в противном случае все числа заменить их квадратами.
8	Даны действительные числа a, b, k . Поменять значения переменных так, чтобы $a > b > k$.
9	Даны действительные числа x, y, x . Вычислить $\max(x, y, x) * \min(x, y, x)$.
10	Даны x, y, z – вещественные числа. Существует ли треугольник с длинами сторон x, y, z ? Если существует, то ответить, является ли он остроугольным.

11	Даны действительные числа x, y, z . Вычислить $\max(x, y, z) * \min(xy, yz, zx)$.
12	Даны действительные числа x, y, z . Вычислить $\min(x+y+z / 2, xyz) + \max(xy, y)$.
13	Даны действительные числа x_1, y_1, y_2 . Вычислить $\max(x_1+y_1, x_1y_1, y_1 y_2) + \min(x_1, y_1, y_2)$.
14	Даны действительные числа y_1, x_2, y_2 . Вычислить $\min(y_1+x_2y_2, y_1+x_2+y_2, y_2)+5$.
15	Даны действительные числа x, y, z . Вычислить $\min(x + y + z, xyz) * \min(x, y, z)$.
16	Определить, есть ли среди заданных целых чисел A, B, C хотя бы одно нечётное.

2. В соответствии со своим вариантом разработать словесно-формульное описание и блок-схему алгоритма для решения задачи из таблицы лабораторной работы № 4.

Тест "Способы представления алгоритмов"

[В начало практикума](#)

Лабораторная работа № 2. Основные элементы языка C++. MS Visual Studio 2015

На языке C++ используются четыре базовых типа литералов (лексемы программы, обозначающие константные значения, встроенные в исходный код): целые литералы, вещественные (действительные с плавающей точкой), символьные и строковые литералы. Они записываются в переменные разных типов. Основные типы данных:

логические: **bool** – тип, способный хранить одно из двух значений: **true** (истина) или **false** (ложь);

целочисленные: **int** – целочисленный тип данных (с ним могут использоваться модификаторы: **signed**, **unsigned**, **short**, **long**, **long long**;

плавающей точкой: **float** – с плавающей точкой одинарной точности; **double** – с плавающей точкой двойной точности; **long double** – тип с плавающей точкой повышенной точности;

символьные: **char** – для представления символов (может иметь модификаторы **signed** и **unsigned**); **wchar_t** – тип для широкого представления символов; **char16_t** – тип для представления символов в UTF-16; **char32_t** – тип для представления символов в UTF-32.

Целый литерал может быть десятичным числом (**–15, 9...**), восьмеричным (**020, 0360**) или шестнадцатеричным (**0x2B, 0x10**). Занимает в памяти **4** байта.

Вещественный литерал с плавающей точкой – это десятичное число с точкой или экспонентой (**115.75, 1.5E-2, –0.025**). Имеет тип **double** (занимает обычно **8** байтов). Если литерал заканчивается буквой **f** или **F**, то он имеет тип **float** (**4** байта). Если литерал заканчивается буквой **l** или **L**, то он имеет тип **long double** (обычно **10** байт).

Символьный литерал состоит либо из одного печатного символа, заключенного в апострофы (**' ', 'Q'**), либо из специального управляющего кода (непечатный символ), заключенного в апострофы (**'\n', '\\'**). Имеет тип **char**, занимает **1** байт.

Величина *логического литерала* (**bool**) может принимать только значения **false** (соответствует 0) и **true** (1). Занимает **1** байт.

Если требуется, чтобы значение переменной не изменялось, то для ее описания используется *ключевое слово* **const**.

В выражениях используются *знаки операций*: **+**, **–**, *****, **/**. Математические функции вычисляются с помощью стандартных встроенных функций.

Математическая запись	sinx	cosx	e^x	$/x/$	tgx	lnx	\sqrt{x}	x^y
-----------------------	-------------	-------------	-------	-------	------------	------------	------------	-------

Запись на C++	sin(x)	cos(x)	exp(x)	abs(x)	tan(x)	log(x)	sqrt(x)	pow(x, y)
---------------	---------------	---------------	---------------	---------------	---------------	---------------	----------------	------------------

Программы начинаются с заголовка, который содержит описание стандартных файлов библиотек, пользовательских файлов и др. Заголовочные файлы включаются в текст программы с помощью директивы препроцессора **#include**. После заголовка следует оператор начала текста программы, например: **void main()** (используется, если у функции нет возвращаемого значения). В фигурных скобках (**{...}**) записывается тело функции.

Все элементы данных должны быть определены в программе перед их использованием. Определения данных и операторы всегда завершаются точкой с запятой. Один оператор может распространяться на несколько строк либо несколько операторов могут находиться в одной строке.

Программы на языке C++ выполняются в приложении **Microsoft Visual Studio 2015** (а также в приложениях **Microsoft Visual Studio 2012, 2013** и др.), для запуска которого надо выполнить: **Пуск / Программы / Microsoft Visual Studio 2015 / Visual Studio 2015**.

Задание	Краткие теоретические сведения
1. Создать новый проект с именем, которое состоит из фамилии студента и номера лабораторной работы.	<p>Для создания нового проекта надо в окне Начальная страница выполнить Создать проект. Среда Visual Studio отобразит окно Создать проект. В окне следует развернуть узел Visual C++, выбрать Win32 и на центральной панели выбрать Консольное приложение Win32. Затем в поле Имя надо ввести имя проекта, в поле Расположение указать путь размещения проекта, или выбрать его с помощью кнопки Обзор.</p> <p>После нажатия ОК откроется окно Мастер приложений Win32. Надо нажать кнопку Далее, в появившемся окне поставить галочку в поле Пустой проект. После нажатия кнопки Готово на экране появится окно Обозреватель решений.</p>
2. Создать новый файл с программой под именем main.cpp	<p>В контекстном меню окна Обозреватель решений выполнить Добавить / Создать элемент. В левой части появившегося окна выбрать Код, в центре выбрать Файл C++. В поле Имя задать имя нового файла, указать расширение ".cpp", нажать Добавить.</p>

<p>3. Ввести программу, текст которой приведен справа, проанализировать ее.</p> <p>Выполнить программу, ознакомиться с результатом.</p>	<pre>#include <iostream> void main() { float x = 3, y; y = x * x + sin(x); std::cout << y; }</pre>	<p>Программа рассчитывает значение $y=2 \cdot x + \sin(x)$ для $x=3$. Она начинается с комментария, который заключен внутри комбинации символов <code>/*</code> и <code>*/</code>.</p> <p>Строка <code>#include <iostream></code> будет заменена содержимым файла <code><iostream.h></code>, который находится в стандартной библиотеке языка и отвечает за ввод-вывод данных на экран.</p> <p>В строке <code>float x=3,y;</code> определяется тип переменных <code>x</code> и <code>y</code> в виде действительных чисел. В <code>x</code> помещается число 3.</p> <p>С помощью оператора <code>std::cout</code> значение <code>y</code> выводится на экран.</p> <p>При наборе кода автоматически отображаются список методов, параметры методов и функций. При наведении мыши на элемент дается его краткое описание.</p> <p>При наборе текста программы одинаковые прописные и строчные буквы считаются <i>различными</i> символами. Если допущена ошибка в ключевом слове, то слово подчеркивается красной волнистой чертой.</p> <p>Если номера строк не отображаются, то надо выполнить Сервис / Параметры / Текстовый редактор / C++. Отметить пункт Показывать номера строк.</p> <p>Для компиляции программы следует выполнить Построение / Компилировать (или нажать клавиши Ctrl–F7). Запуск программы на выполнение: Отладка / Запуск без отладки (или Ctrl–F5).</p> <p>Повторный вызов консольного окна на экран – также Ctrl–F5.</p>
<p>4. В предыдущем проекте заменить текст программы на программу, записанную в правой</p>	<p>Исходные данные: $k = 4$; $a = 4.1$; $x = 5 \cdot 10^{-5}$. Вычислить:</p>	<pre>#include <iostream> void main() { double t, u, k = 4, a = 4.1, x = 5e-5; t = 2 * tan(k) / a * log(abs(3 + x)) + exp(x); u = sqrt(t + 1) * (sin(x) * cos(t)); }</pre>

части данного пункта, выполнить ее.	$t = 2\text{tg}(k)/a - \ln(3 + x) + \exp(x);$ $u = \sqrt{t+1} \cdot (\sin(x) - \cos(t));$	<pre>std::cout << "t=" << t; std::cout << "u=" << u; }</pre>
--	--	--

5. В таблице ниже приведены условия задач. В соответствии с вариантом, номер которого определяет преподаватель, разработать программу для задачи. Опробовать работу программы и проанализировать результаты. Возможны ситуации, когда исходные данные заданы некорректно и при выполнении программы возникают ошибки (деление на ноль, корень из отрицательного числа и т. п.). В таком случае надо выполнить программу с другими исходными данными.

№	Формулы для вычислений	Исходные данные	№	Формулы для вычислений	Исходные данные
1	$t = 2k/a + \ln(2 + x)$ $u = \sqrt{k - 1}/(t + 1)$	$k = 4; x = 2 \cdot 10^{-4};$ $a = 8,1$	9	$z = e^{cx} / y + 3$ $u = \sqrt{z^3 - 0,1z}$	$y = 0,5; c = 1,4;$ $x = 2 \cdot 10^{-4}$
2	$z = 1/(x - 1) + \sin x - \sqrt{n}$ $y = (e^{-b} + 1)/2z$	$n = 2; b = -0,12;$ $x = 1,3 \cdot 10^{-4}$	10	$y = (z + \ln z)/(e^{-3} + \sqrt{a})$ $s = (1 + m \cdot n)/\ln(1 + z)$	$z = 1,7; a = 4 \cdot 10^{-8};$ $m = 3; n = 3$
3	$d = \text{tg}(-x \cdot i) / \sqrt{x - z}$ $f = \sin(2d) / d$	$i = -6; x = 4,5;$ $z = 1,5 \cdot 10^{-6}$	11	$w = \text{tg} 1 \cdot (1 + x) + z - e^a$ $d = 9\sqrt{2 - 3 \cdot x} + a + 1 $	$a = 1,5; x = -1,8$ $z = 15 \cdot 10^{-9}$
4	$p = 2,6t + \cos(y/(3x + y))$ $q = \sin t / \cos t$	$t = 6; y = -1,2;$ $x = 0,4 \cdot 10^6$	12	$t = \text{tg} x + r \cdot (1 - \ln x)$ $u = t/(x^3 + 1)/(1 - e^{k-4})$	$k = 6; r = 5 \cdot 10^{-7}$ $x = 0,095$

№	Формулы для вычислений	Исходные данные	№	Формулы для вычислений	Исходные данные
5	$w = (a + b) \operatorname{tg} x / (x + 1)$ $v = 1 / 2 \cdot b - \sqrt{w - a \cdot b}$	$b = 40; x = 1,1;$ $a = 5 \cdot 10^{-6}$	13	$t = (d \cdot c + a \cdot \sqrt{c - 1})$ $y = 0,5t / d + e^a$	$d = 0,5 \cdot 10^{-8}; c = -9;$ $j = -4; a = 1,5$
6	$t = 1 / \sqrt{y} + 14a$ $u = (t + 1) / (a + 2)$ $s = \ln((2n / 3) + e^{-n} / u)$	$y = 0,956;$ $a = 5 \cdot 10^{-6};$ $n = 4$	14	$s = \sqrt{t \cdot a / t + 1} + 4e^{2 \cdot b}$ $w = s \cdot a / (1 + 0,1a)$ $v = s + j \cdot \sqrt{a^2 + b^2}$	$a = 1,5; b = -8,1;$ $j = 4; t = 4 \cdot 10^{-8}$
7	$y = \sqrt{1 + x} - \cos(2 / m)$ $w = 0,6z - 2e^{-2 \cdot y \cdot m}$	$x = 1,4; m = 6;$ $z = 0,05 \cdot 10^{-5}$	15	$y = a \cdot e^{-2b} - \sqrt{1 + a}$ $r = \ln(1 + 20b) / (1 + a)$	$a = 1,75;$ $b = 4,5 \cdot 10^{-4}$
8	$s = 0,4x - 1 / j \cdot \operatorname{tg} y$ $t = s - \sin s$	$x = 0,1; j = 12;$ $y = 5 \cdot 10^{-6}$	16	$y = \cos(5m) / \sin^2(0,4m)$ $w = 4z \cdot y - 7e^{-2 \cdot y}$	$m = 6;$ $z = 0,05 \cdot 10^{-5}$

6. К номеру своего варианта прибавить 3 и написать программу для новых исходных данных (для вариантов с 14 по 16 перейти к вариантам с 1 по 3). Проверить выполнение программы.

В тесте - не верная проверка по количеству ответов

Тест "Основные элементы языка C++"

В начало практикума

Лабораторная работа № 3. Ввод и вывод информации

На языке C++ используются следующие способы ввода-вывода информации: потоковый, форматированный, символьный.

Задание	Краткие теоретические сведения
<p>1. Выполнить программу, записанную справа, которая использует <i>потоковый</i> вывод данных. Проанализировать ее текст. Добавить в программу ввод и вывод переменных различных типов.</p>	<div data-bbox="622 435 1543 985"><pre>void main() { setlocale(LC_CTYPE, "Russian"); int t; cout << "Введите t="; cin >> t; cout << "t=" << t << endl; cout << "Тип Размер в байтах" << endl; cout << "int: " << sizeof(int) << endl; cout << "char: " << sizeof(char) << endl; cout << "float: " << sizeof(float) << endl; cout << "double: " << sizeof(double) << endl; // sizeof определяет размер объекта в байтах }</pre></div> <div data-bbox="1563 435 2112 992"><p>setlocale (LC_CTYPE, "Russian"); используется для вывода кириллицы.</p><p>cout – стандартный поток вывода (на экран дисплея); cin – стандартный поток ввода (с клавиатуры); cerr, clog – стандартный поток сообщений об ошибках; endl – перевод на новую строку.</p><p>Если в программе записано using namespace std; (используя стандартное пространство имен),</p></div> <p>то перед каждым оператором вывода можно не писать std::</p> <p>// – начало комментария, конец которого определяется концом строки.</p> <p>Выражение cin >> x; используется для ввода числа с клавиатуры в переменную x. В данном примере вводится целое число, которое при выполнении программы нужно набрать на клавиатуре и нажать клавишу Enter.</p>

2. Опробовать работу программы, приведенной в правой части.

Добавить в программу ввод и вывод переменных с использованием манипуляторов.

```
#include <iomanip>
void main()
{
    setlocale(LC_CTYPE, "Russian");
    using namespace std;
    char c, probel; probel = ' ';
    cout << "Введите символ "; cin >> c;
    cout << setw(35) << setfill(probel) << probel;
    cout << setw(10) << setfill(c) << c << endl;
    cout << setw(34) << setfill(probel) << probel;
    cout << setw(12) << setfill(c) << c << endl;
    cout << setw(33) << setfill(probel) << probel;
    cout << setw(14) << setfill(c) << c << endl;
}
```

символ-заполнитель; **setprecision(int n)** – установить количество цифр после запятой при выводе вещественных чисел; **setbase(int n)** – установить систему счисления для вывода чисел (**n** может принимать значения 0, 2, 8, 10, 16, причем 0 означает систему счисления по умолчанию, т.е. 10).

Некоторые манипуляторы определены в заголовочном файле **<iomanip>**, поэтому при их использовании надо добавлять директиву **#include <iomanip>**.

Пример вывода числа из 6 символов (3 до запятой и 2 после):

```
using namespace std;
.....
double x;
cout << setw(6) << setprecision(2) << x << endl;
```

При выводе данных могут использоваться *манипуляторы*: **endl** – перейти на новую строку; **ends** – вывести нулевой байт (признак конца строки символов); **dec** – вывод числа в десятичной системе; **oct** – вывод числа в восьмеричной системе; **hex** – вывод числа в шестнадцатеричной системе счисления; **setw(int n)** – установить ширину поля вывода в **n** символов; **setfill(int n)** –

<p>3. Выполнить программу, записанную в правой части, которая использует <i>форматированный</i> ввод-вывод данных.</p> <p>Изменить программу так, чтобы выводилась своя фамилия, имя и отчество, факультет, номер группы русскими буквами.</p> <p>При выводе использовать управляющие коды.</p>	<p>Можно управлять перемещением курсора на экране и выполнять некоторые другие функции, используя <i>управляющие коды</i>: \n - перемещает курсор в начальную позицию следующей строки; \t – перемещает курсор в следующую позицию табуляции экрана; \r – выполняет «возврат каретки», перемещая курсор к началу той же строки без перехода на следующую; \b – передвигает курсор только на одну позицию влево.</p> <p>Пример программы, выводящей слово "Privet".</p> <table><tr><td><pre>#include <stdio.h> #include <conio.h> void main() { printf("\n\t Privet\n"); printf("\n... Press key"); _getch(); }</pre></td><td><p>Заголовочный файл с именем stdio.h может использоваться при форматированном вводе-выводе данных.</p><p>Заголовочный файл conio.h поддерживает функцию _getch(), которая ожидает нажатия клавиши на клавиатуре.</p><p>Функция printf() печатает информацию.</p></td></tr></table>	<pre>#include <stdio.h> #include <conio.h> void main() { printf("\n\t Privet\n"); printf("\n... Press key"); _getch(); }</pre>	<p>Заголовочный файл с именем stdio.h может использоваться при форматированном вводе-выводе данных.</p> <p>Заголовочный файл conio.h поддерживает функцию _getch(), которая ожидает нажатия клавиши на клавиатуре.</p> <p>Функция printf() печатает информацию.</p>
<pre>#include <stdio.h> #include <conio.h> void main() { printf("\n\t Privet\n"); printf("\n... Press key"); _getch(); }</pre>	<p>Заголовочный файл с именем stdio.h может использоваться при форматированном вводе-выводе данных.</p> <p>Заголовочный файл conio.h поддерживает функцию _getch(), которая ожидает нажатия клавиши на клавиатуре.</p> <p>Функция printf() печатает информацию.</p>		
<p>4. Внести изменения в программы лабораторной работы № 2 с тем, чтобы осуществлялся <i>форматированный</i> ввод и вывод данных.</p>	<p>Каждому аргументу функции printf() соответствует своя спецификация преобразования, которая начинается с символа процента (%), после которого стоит буква, указывающая тип данных: %d – десятичное целое число; %f – вещественное число типа float или double; %c – символ; %s – строка; %p – указатель; %u – беззнаковое целое число; %o – целые числа в восьмеричной системе счисления; %x – целые числа в шестнадцатеричной системе счисления; %e – вещественное число в экспоненциальной форме.</p> <p>В модификаторах формата после символа % можно указывать строку цифр, задающую минимальную ширину поля вывода, например: %5d (для целых), %4.2f (для вещественных –</p>		

	<p>две цифры после запятой для поля шириной 4 символа). Если указанной ширины не хватает, происходит автоматическое расширение.</p> <p>Функция scanf() позволяет вводить данные любых типов. В качестве аргументов используются указатели объектов &. Например: scanf(“ %d%f ”, &x, &y);</p> <p>Если нужно ввести значение строковой переменной, то использовать символ & не нужно. Строка – массив символов, а имя массива эквивалентно адресу его первого элемента:</p> <pre>char name[20]; scanf(“%s”, name);</pre> <p>В новых версиях Visual Studio компилятор C++ считает функцию scanf небезопасной и выдаёт ошибку во время компиляции. Можно использовать безопасную версию scanf_s либо добавить строку для отключения этой ошибки:</p> <pre>#define _CRT_SECURE_NO_WARNINGS</pre> <p>То же самое относится и к некоторым другим функциям, описанным ниже.</p>
5. Написать программу, реализующую диалог, используя пример, записанный в правой части.	<div><pre>#include <windows.h> void main() { SetConsoleOutputCP(1251); SetConsoleCP(1251); char name[60]; puts("Как вас зовут? "); gets_s(name); printf("Привет, %s\n", name); }</pre></div> <p>переменная: char letter; letter='G'; putchar (letter);</p> <p>Функция getchar() вводит с клавиатуры единичный символ: int letter; letter = getchar();</p>

	<p>Для корректного ввода и вывода кириллицы на консоль в примере использовали пары функций: SetConsoleOutputCP() и SetConsoleCP(). В качестве единственного параметра обеим функциям передается номер кодовой страницы, кириллицы - это 1251.</p>
--	---

6. В соответствии со своим вариантом написать программы по условиям, приведенным в таблице ниже. Использовать различные способы для ввода и вывода данных.

№ варианта	Условие задачи
1	1. Нарисовать полукруг, закрашенный некоторым введенным символом.
	2. Поменять значения двух переменных: а) используя дополнительную переменную; б) не используя дополнительной переменной.
2	1. Нарисовать шестиугольник по центру консольного окна, закрашенный введенным символом.
	2. Разделить число m на две части, разность которых равна 5.
3	1. Нарисовать елочку, закрашенную введенным символом.
	2. Найти площадь прямоугольника, основание которого в n раз больше высоты, а площадь равна периметру.
4	1. Нарисовать круг по центру консольного окна, закрашенный введенным символом.
	2. Даны действительные числа X , Y , Z . Определить среднее арифметическое и среднее геометрическое их модулей.
5	1. Нарисовать треугольник по центру консольного окна, закрашенный введенным символом

№ варианта	Условие задачи
	2. Животновод в начале каждой зимы повышает отпускную цену на молоко на $p\%$, а каждым летом снижает на столько же процентов. Изменится ли цена на молоко и если да, то в какую сторону и на сколько через n лет?
6	1. Нарисовать звезду, закрашенную введенным символом.
	2. Определить расстояние на плоскости между двумя точками с заданными координатами $M1(x1,y1)$ и $M2(x2,y2)$.
7	1. Нарисовать трапецию по центру консольного окна, закрашенную введенными символами.
	2. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.
8	1. Нарисовать человечка, закрашенного введенным символом.
	2. Вычислить площадь треугольника по формуле Герона, если заданы его стороны.
9	1. Нарисовать разнонаправленные стрелки ($\rightarrow \uparrow \downarrow \leftarrow$), состоящие из введенного символа.
	2. Дано действительное число a . Не пользуясь никакими другими операциями, кроме умножения, получить a^4 за две операции.
10	1. Нарисовать овал, закрашенный введенным символом.
	2. N школьников делят K яблок поровну, остаток остается в корзинке. Сколько яблок достанется каждому школьнику?
11	1. Нарисовать квадрат, закрашенный введенным символом.
	2. Известна диагональ квадрата. Вычислить его площадь.

№ варианта	Условие задачи
12	1. Нарисовать знак +, закрашенный введенным символом.
	2. Сформировать число, представляющее собой реверсную (обратную в порядке следования разрядов) запись заданного трехзначного числа.
13	1. Нарисовать сердце ♥, закрашенное введенным символом.
	2. В бригаде, работающей на уборке сена, имеется N косилок. Первая из них работала М часов, а каждая следующая на 10 мин. Больше, чем предыдущая. Сколько часов работала вся бригада?
14	1. Нарисовать ромб ♦, закрашенный введенным символом.
	2. Автомобиль на каждом из пяти одинаковых по длине участках дороги шел с известной средней скоростью. Составьте алгоритм и программу для определения средней скорости на всем пути.
15	1. Нарисовать домик, закрашенный введенным символом.
	2. Определить номера подъезда и этажа по номеру квартиры девятиэтажного дома, считая, что на каждом этаже ровно 4 квартиры, а нумерация квартир начинается с первого подъезда.
16	1. Нарисовать грибок, закрашенный введенным символом.
	2. В компьютерной игре игрок выигрывает 50 очков, если он сбивает самолет; 100 очков, если он сбивает ракету; 200 очков, если он сбивает спутник. Определить число очков игрока, который сбил А самолетов, В ракет и С спутников.

Тест "Ввод и вывод информации"

[В начало практикума](#)

Лабораторная работа № 4. Разветвляющиеся программы

В разветвляющихся программах выполняются вычисления по одним или другим формулам в зависимости от некоторых условий.

Задание	Краткие теоретические сведения	
<p>1. Выполнить программу, приведенную в правой части, несколько раз с различными значениями переменной j.</p> <p>Переделать программу с тем, чтобы ввод и вывод осуществлялся с помощью потоковых операторов ввода-вывода.</p>	<pre>#include <iostream> void main() { setlocale(LC_CTYPE, "Russian"); float b, z, x = 4e-3, s = 1.1, j; printf_s("Введите j "); scanf_s("%f", &j); b = s + (5 * x + j); if (b < 1.5) z = sqrt(x + 0.3 * j) + b; else z = abs(x * j + b); printf("j=%f\t", j); printf("z=%f\n", z); }</pre>	<p>Пример. Вычислить значение z по формулам:</p> $b = s + (5 * x + j),$ $z = \begin{cases} \sqrt{x + 0,3j} + b, & b < 1,5 \\ /xj + b/, & b \geq 1,5 \end{cases},$ <p>где x = 4·10⁻³; s = 1,1; j = {2; 6,8; 0,03; 55; ...}.</p> <p>Здесь в программе для организации выбора используется условный оператор if. Формат оператора: if<условие> оператор-1; [else оператор-2;] или if<условие> оператор-1;</p> <p>Выполнение оператора if начинается с проверки условия. Если условие истинно, то выполняется оператор 1, если условие ложно, то выполняется оператор 2.</p> <p>Если условие ложно и отсутствует оператор 2, то выполняется следующий за if оператор.</p>

	При каждом повторении операторов цикла если выражение b < 1,5 истинно, то z вычисляется по первой формуле, в противном случае – по второй.	
<p>2. Выполнить программу, записанную в правой части.</p> <p>Изменить программу с тем, чтобы ввод и вывод осуществлялся с помощью потоковых операторов ввода-вывода.</p>	<pre> #include <iostream> void main() { setlocale(LC_CTYPE, "Russian"); float x, y, z, mn, mx; printf("Введите x "); scanf_s("%f", &x); printf("Введите y "); scanf_s("%f", &y); printf("Введите z "); scanf_s("%f", &z); if ((x + y + z) < (x * y * z)) mx = (x * y * z); else mx = (x + y + z); mn = x; if (mn > y) mn = y; if (mn > z) mn = z; printf("max(x + y + z, xyz) * min(x, y, z) = %f \n", mn*mx); } </pre>	<p>Пример.</p> <p>Ввести с клавиатуры вещественные числа x, y, z, используя функции</p>
	форматированного ввода. Вычислить max(x + y + z, xyz) * min(x, y, z) .	
<p>3. Выполнить программу с использованием оператора выбора switch, записанную в правой части.</p>	<p>Для организации выбора из множества различных вариантов используется оператор выбора switch. Формат оператора:</p> <p>switch (выражение) { [case константное выражение 1]: [список операторов 1];</p>	

Записать условие задачи.
Изменить программу с тем, чтобы в программе присутствовал оператор **goto**.

.....
[**case** константное выражение n]: [список операторов n];
[**default**: [список операторов];] }

```
#include <iostream>
int main()
{
    setlocale(LC_CTYPE, "Russian");
    int k;
    puts("Вы хотите купить автомобиль? (1-да, 2-нет)");
    std::cin >> k;
    switch (k)
    {
        case 1: { puts("Какой? (1- Майбах, 2-Тойота, 3-Лада)");
                  std::cin >> k;
                  switch (k)
                  {
                      case 1: puts("Не слабо"); break;
                      case 2: puts("Хороший выбор"); break;
                      case 3: puts("Сомнительный выбор");
                             break;
                  }
                  break;
        }
        case 2: puts("Полезно ходить пешком"); break;
        default: puts("Некорректный вариант"); break;
    }
    return 0;
}
```

Выражение в круглых скобках может быть целой или символьной константой. Выполняется оператор следующим образом: вычисляется выражение в круглых скобках и полученное значение последовательно сравнивается с константными выражениями, следующими за ключевыми словами **case**. Если одно из выражений совпадает со значением константного

выражения, то управление передается оператору, помеченному соответствующим ключевым словом **case**. Если совпадения нет, то управление передается на оператор, помеченный ключевым словом **default**, а в случае его отсутствия – на оператор, следующий после **switch**.

Оператор **break** передает управление оператору, следующему за **switch**.

Оператор **return 0** завершает выполнение функции и возвращает системе значение **0**.

В некоторых случаях приходится использовать оператор безусловного перехода **goto**, который передает управление на оператор, помеченный меткой (например, оператор **goto a**; передает управление оператору: **a: std::cin>>k**; Здесь **a** – метка оператора). Без крайней необходимости не рекомендуется использовать оператор **goto**.

4. Написать и выполнить программу по первой блок-схеме лабораторной работы № 1.
5. В соответствии со своим вариантом написать программу по условию, приведенному в таблице ниже.

№ варианта	Условие задачи
1	Ввести три числа m, n, p. Подсчитать количество отрицательных чисел.
2	Определить, имеется ли среди трёх чисел a, b и c хотя бы одна пара равных между собой чисел.
3	Даны три числа a, b и c. Найти среднее геометрическое этих чисел, если все они отличны от нуля, и среднее арифметическое в противном случае.
4	По номеру месяца напечатать пору года.
5	Определить, есть ли среди заданных целых чисел A, B, C, D хотя бы одно нечётное.
6	Дано натуральное трехзначное число n. Верно ли, что среди его цифр есть 0 или 9?

№ варианта	Условие задачи
7	В переменную Y ввести номер года. Определить, является ли год високосным.
8	Дано натуральное четырехзначное число n. Верно ли, что все его цифры различны?
9	Для натурального числа K напечатать фразу "мы нашли K грибов в лесу", согласовав окончание слова "гриб" с числом K.
10	Число делится на 3 тогда, когда сумма его цифр делится на 3. Проверить этот признак на примере заданного трехзначного числа.
11	Даны три числа a, b, c. Значение наибольшего из них присвоить переменной d.
12	По значению переменной X, означающему длину в километрах, метрах, миллиметрах или сантиметрах, напечатать длину в метрах.
13	Проверить, является ли дробь A/B правильной.
14	Для целого числа K от 1 до 9 напечатать фразу "мне K лет", учитывая при этом, что при некоторых значениях K слово "лет" надо заменить на слово "год" или "года".
15	Определить есть ли среди заданных целых чисел A, B, C, D хотя бы одно чётное.
16	По введенному числу (от 0 до 7) напечатать название цифры.

6. Написать программу, реализующую диалог на любую тему с использованием оператора **switch**.

7. Дополнительные задачи

№	Условие задачи
1	Разработать консольную программу-калькулятор для целых чисел с операциями %, /, +, -, *.

№	Условие задачи
2	На шахматном поле в (k, l) расположен слон. Угрожает ли он фигуре стоящей на (m, n) ? Предусмотреть обработку ошибочного ввода пользователя (выдавать диагностическое сообщение).
3	Можно ли с шахматного поля (k, l) одним ходом ферзя попасть на поле (m, n) ? Если нет, то выяснить, как это можно сделать за два хода.
4	Можно ли на прямоугольном участке застройки размером a на b метров разместить два дома размером v на p и q на s метров? Дома можно располагать только параллельно сторонам участка.
5	Может ли шар радиуса r пройти через ромбообразное отверстие с диагоналями p и q ?
6	Можно ли коробку размером $a*b*c$ упаковать в посылку размером $r*s*t$? «Углом» укладывать нельзя.
7	Четырехугольник ABCD задан координатами своих вершин на плоскости: $A(x_a, y_a)$, $B(x_b, y_b)$, $C(x_c, y_c)$, $D(x_d, y_d)$. Определить тип четырехугольника: прямоугольник, параллелограмм, трапеция, произвольный четырехугольник. Учесть погрешность вычислений. Замечание: для устранения дополнительных источников погрешности рекомендуется использовать аппарат векторной алгебры: коллинеарность, равенство и ортогональность векторов – сторон четырехугольника
8	Можно ехать на такси со скоростью v_1 км/ч и оплатой p_1 р/км либо идти пешком со скоростью v_2 км/ч бесплатно. Как с наименьшими затратами преодолеть путь s за время t , если это возможно? Каковы эти затраты? Замечание: рекомендуется рассмотреть «запредельные» случаи: когда времени слишком мало, чтобы успеть даже на такси, либо слишком много, так что и пешком можно с запасом успеть до отхода поезда.
9	На шахматной доске стоят черный король и белые ладья и слон. Проверить, есть ли угроза королю и если есть, то от кого именно.

№	Условие задачи
10	Два отрезка на плоскости заданы координатами своих концов. Определить, имеют ли эти отрезки общие точки. Замечание: необходимо рассмотреть различные случаи взаимной ориентации отрезков: на одной прямой, на параллельных или пересекающихся прямых.

Тест "Разветвляющиеся программы"

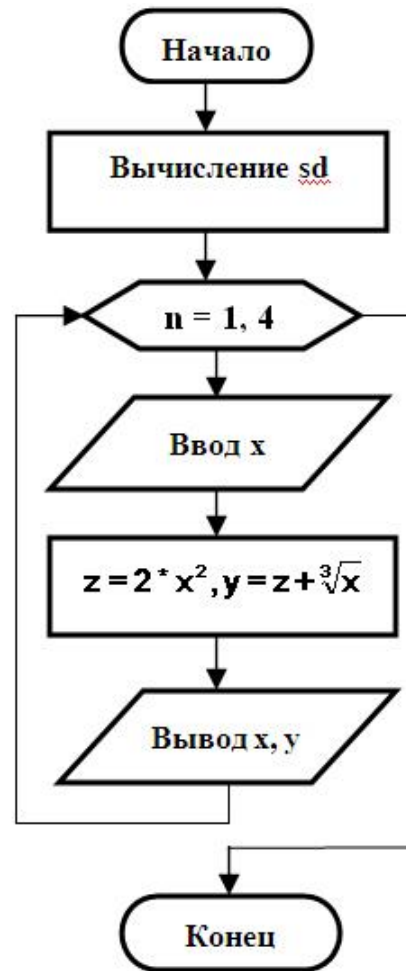
[В начало практикума](#)

Лабораторная работа № 5. Циклические программы

Цикл – это участок программы, в котором одни и те же вычисления реализуются неоднократно при различных значениях одних и тех же переменных (объектов).

Для организации циклов на языке C++ используются операторы: **for**, **while**, **do while**.

Задание	Краткие теоретические сведения	
<p>1. Выполнить программу с использованием оператора for, записанную в правой части.</p> <p>Изучить блок-схему алгоритма.</p>	<p>Пример. Вычислить значение y по формулам:</p> <p>z = 2 * x², y = z + ³√x ,</p> <p>где x = {4; 3,5; 7; 1}.</p> <p>В программе тело цикла for заключено в фигурные скобки. В переменной n находится номер значения x. Вычисляется выражение n = 1 и проверяется условие n <= 4. Если оно истинно, то выполняются операторы цикла.</p> <p>Затем вычисляется n = n + 1, вновь проверяется условие n <= 4 и, если оно истинно, то вновь выполняются операторы цикла.</p> <p>Так продолжается до тех пор, пока n не станет больше 4. Тогда управление передается оператору, следующему за телом цикла.</p> <p>В переменной sd вычисляется выражение (1 / 3), которое используется при вычислении корня третьей степени из x.</p> <p>Результаты выводятся в виде действительных чисел, занимают 5 символов, из которых 2 отводятся для записи дробной части.</p>	
	Блок-схема алгоритма	Программа



```

#include <iostream>
void main()
{
    setlocale(LC_CTYPE, "Russian");
    float z, y, x, sd;
    sd = 1.0 / 3.0;
    for (int n = 1; n <= 4; n++)
    {
        printf("Введите x ");
        scanf_s("%f", &x);
        z = 2 * pow(x, 2);
        y = z + pow(x, sd);
        printf("x = %5.2f\t", x);
        printf("y = %5.2f\n", y);
    }
}
  
```

2. Выполнить программу с использованием оператора **while**, записанную в правой части.

Опробовать способ прерывания работы программы с помощью клавиш **Ctrl – Break**.

Пример. Вычислить значение y по формулам: $z = 2 * x^2, y = z + \sqrt[3]{x}$, где $x = 3(0,1)4$ – меняется от 3 до 4 с шагом 0,1.

```
#include <iostream>
using namespace std;
void main()
{
    float z, y, x = 3;
    while (x < 4.1)
    {
        z = 2 * pow(x, 2);
        y = z + pow(x, (float)1 / 3);
        cout << "x=" << x << "\t";
        cout << " y=" << y << endl;
        x = x + 0.1;
    }
}
```

В этой программе тело цикла **while** заключается в фигурные скобки. Если условие в скобках ($x < 4.1$) истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным. Тогда управление передается оператору, следующему за телом цикла.

Если в программе будет отсутствовать оператор $x = x + 0.1$, то программа заикнется. Надо прервать выполнение программы, нажав **Ctrl – Break**, и исправить ошибку.

Запись **(float)1 / 3** – это приведение

типов в стиле C. Тип данных в скобках указывает, в какой тип необходимо преобразовать ближайший операнд.

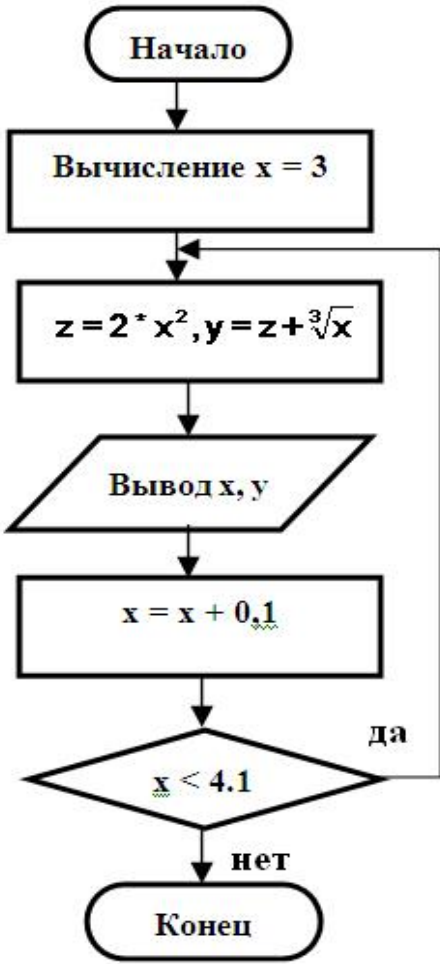
Например, выражение **(float)int / int = float / int** даст в результате значение с типом **float**.

3. Выполнить программу с использованием оператора **do while**, записанную в правой части. Проанализировать ее.

Пример. Вычислить значение y по формулам: $z = 2 * x^2, y = z + \sqrt[3]{x}$, где $x = 3(0,1)4$ – меняется от 3 до 4 с шагом 0,1.

Оператор цикла **do while** называется оператором цикла с постусловием.

Сначала выполняется тело цикла, затем проверяется условие, записанное в скобках ($x < 4.1$), если оно истинно, то выполняются операторы цикла до тех пор, пока оно не станет ложным. Тогда управление передается оператору, следующему за телом цикла.

Изучить алгоритма.	Изучить блок-схему	Программа
	<p style="text-align: center;">Блок-схема алгоритма</p>  <pre> graph TD Start([Начало]) --> CalcX[Вычисление x = 3] CalcX --> CalcZY["z = 2 * x², y = z + √³x"] CalcZY --> Output[/Вывод x, y/] Output --> IncX["x = x + 0.1"] IncX --> Decision{x < 4.1} Decision -- да --> CalcZY Decision -- нет --> End([Конец]) </pre>	<p style="text-align: center;">Программа</p> <pre> #include <stdio.h> #include <cmath> void main() { float z, y, x = 3; do { z = 2 * pow(x, 2); y = z + pow(x, (float)1 / 3); printf("x=%5.2f\t", x); printf("y=%5.2f\n", y); x = x + 0.1; } while (x < 4.1); } </pre>

<p>4. Выполнить программу, содержащую вложенный цикл. Записать условие задачи.</p> <p>Оформить вывод результатов, используя различные возможности операторов вывода.</p>	<pre>#include <stdio.h> void main() { int n, i, j; printf("Enter n: "); scanf_s("%d", &n); for (i = 1; i <= n; i++) { for (j = 1; j <= n; j++) printf("%5d", i * j); printf("\n"); } }</pre> <p>Пояснить результат.</p>	<p>Вместо #include <stdio.h> в заголовочной части записать:</p> <pre>#include <iostream> #include <iomanip> using namespace std;</pre> <p>Перед printf("\n"); разместить операторы:</p> <pre>if (i == 1) { cout<<endl; cout << setw(5*n)<<setfill('-') << '-'<<endl; }</pre>	

5. В таблице приведены формулы и три варианта исходных данных, по которым надо разработать три блок-схемы и три циклические программы с одними и теми же расчетными формулами.

При наличии ошибок из-за некорректных исходных данных выполнить вычисления с другими числами.

№	Формулы для вычислений	Исходные данные (for)	Исходные данные (while)	Исходные данные (двойной цикл)
1	$h = (10r - j) / (c^2 + e^{-m})$ $y = (h \cdot m - j^2) + (0,1c)^2$	$c = 2,1; r = 4 \cdot 10^{-4}$ $m = 7;$ $j = \{4,2; 0,3; 1,7\}$	$j = 0(0,1)1,7$	$j = \{9; 1,8; 15; -3\},$ $m = 1(0,5)2$
2	$y = \sqrt{i \cdot b - b^2 \cdot a}$ $z = y \cdot \operatorname{tg}(n/4) - e^{1+b}$	$a = 2 \cdot 10^{-3};$ $b = 8,5; n = 2;$ $i = \{2; 1; 8,3\}$	$i = 1(0,5)3$	$b = 2(0,5)3,$ $n = \{3; -6; 0,2; 2,8\}$
3	$z = \sqrt{a+1} - \operatorname{tg}(j+y)$ $q = e^{-j \cdot 0,1y} + (3 \cdot z)^2$	$y = -1,55; a = 6$ $j = \{8; -0,6; 1; 6,4\}$	$j = -1(0,2)1$	$a = 1(0,2)2,$ $y = \{3,3; -4; 0,9\}$
4	$y = (m \cdot j) / \operatorname{tg} a - e^{10m}$ $z = 2y \cdot b + \sqrt{a+b}$	$b = 2,5; m = 3$ $a = 1,4 \cdot 10^{-3}$ $j = \{0,7; -2; 2,9\}$	$j = b(0,1)3$	$m = \{7,3; -2; 0,8\},$ $y = \{0,3; 1; 0,7; 4\}$
5	$c = i/k - \sqrt{y}/0,4$ $d = e^{1-c} + 4,9(x^2 + 1)$	$i = \{0,9; 8,4; 2\}$ $y = 1,6 \cdot 10^{-4}$ $x = -1; k = 6$	$i = 0(0,5)3$	$x = 1(0,1)2,$ $y = \{1,3; -8; 0,2\}$
6	$v = u + b - 2\sqrt{0,7k + m}$ $w = 100 \cdot e^{-0,2u} - \ln(8,1u)$	$b = -5,4; m = 4$ $u = 0,05 \cdot 10^{-4}$ $k = \{6; 4; 0,3; -7\}$	$k = 3(0,2)4$	$b = 7(0,2)8,$ $m = \{-1,3; -2; 4,9\}$

№	Формулы для вычислений	Исходные данные (for)	Исходные данные (while)	Исходные данные (двойной цикл)
7	$z = t \cdot y^2 - \sqrt{i + x} \cdot \operatorname{tg} y$ $q = \sqrt{z^2 + 5z} \cdot \ln(y)$	$x = 0,9; y = 2;$ $t = 6,96 \cdot 10^{-5}$ $i = \{9; -1,4; 5\}$	$i = 1(0,2)2$	$t = 2(0,2)3,$ $x = \{0,7; -1; 9\}$
8	$w = 0,6j + e^{t \cdot j} \cdot (4y/n)^2$ $s = \sqrt{w - 0,1t} / (2 + n^2)$	$y = 5; t = -7,4;$ $n = 9; j = \{0,5; 8,4; 0,3\}$	$j = 0(0,1)2$	$y = \{0,1; -3; 0,5\},$ $j = 0,1(0,1)0,4$
9	$y = (s + c) / \ln(f) / e^{-s}$ $h = (y - m) / \ln(y)$	$s = 5,9; m = 6$ $f = 1,2 \cdot 10^{-3}; c = \{2; 0,5; -4\}$	$c = 0(0,1)0,9$	$s = \{-3; 0,8; 4\},$ $c = 0,2(0,1)0,5$
10	$y = s / \ln(5,2f) / (e^{-s} + 2)$ $v = (1 + m \cdot y - m \cdot k) / \ln(y)$	$s = 7,4; m = 10$ $f = 3,2 \cdot 10^{-4}; k = \{4; 0,5; 8\}$	$k = 0(0,5)4$	$k = \{0,9; -11; 0,5\},$ $m = 0,3(0,1)0,7$
11	$w = \operatorname{tg}(a/3) + e^{a/m}$ $r = 0,9\sqrt{w + j} + a^2 - 1 $	$a = -1,4; m = 16$ $j = \{0,5; 9,1; 5\}$	$j = 1,8(0,2)3$	$a = \{0,2; -4; 0,6\},$ $j = 0,1(0,1)0,4$
12	$y = (m - b) / (\sin a - e^a)$ $z = 3y + \sqrt{a - 4jb}$	$b = 0,5; m = 8$ $a = 2,4 \cdot 10^{-4}; j = \{1,7; 2,9; -8\}$	$j = 2(0,5)3$	$b = 2(0,5)3,$ $m = \{0,4; -1; 1,9\}$
13	$d = \sin(k/a) / \cos(m \cdot b)$ $c = d / (d^2 + 1) / (1 - e^k)$	$a = 8; m = 6$ $b = 5 \cdot 10^{-3}; k = \{1,6; 9,1; 8\}$	$k = a(-0,5)3$	$a = 2(0,2)2,8,$ $x = \{1,7; 5; -2\}$

№	Формулы для вычислений	Исходные данные (for)	Исходные данные (while)	Исходные данные (двойной цикл)
14	$y = a / (b + e^b) / (1 + j \cdot i)$ $t = \cos(y + 1) / \sqrt{ -2j }$	$b = 2; j = 6$ $a = 1,5 \cdot 10^{-8}; i = \{7; 4; 2; 6\}$	$i = 3(-0,1)2$	$b = 2,4(0,2)3,$ $a = \{0,1; -1; 3\}$
15	$z = \sqrt{t \cdot a + y} + 4e^{-2 \cdot j}$ $w = \ln(0,4y) / (7a - z)$	$a = -1; y = 0,4;$ $t = 5 \cdot 10^{-5}; j = \{5; 3; 1; 7; 3\}$	$j = 1(0,2)2$	$t = 3(0,2)4,$ $a = \{0,2; -1,6; 9,4\}$
16	$w = \cos(a / 3) + t + e^{a/m}$ $r = 0,7\sqrt{3w + j} + a^2 - 1 $	$a = -1,4; m = 6$ $t = 15 \cdot 10^{-5}; j = \{0,5; 9,1; 5\}$	$j = 1(0,1)2$	$m = 8(0,2)9,$ $x = \{0,6; -0,1; 5\}$

6. Дополнительные задания.

1. Шестизначный автобусный билет считается удачным, если сумма его цифр делится на 7. Могут ли два билета подряд быть удачными?

2. Торговая фирма в первый день работы реализовала товаров на Р тыс. руб., а затем ежедневно увеличивала выручку на 3%. Какой будет выручка фирмы в тот день, когда она впервые превысит заданное значение Q? Сколько дней придется торговать фирме для достижения этого результата?

3. О каждом учащемся класса известны его пол, год рождения, рост и вес. Определить, сколько в классе мальчиков и сколько девочек. Найти средний возраст мальчиков и девочек. Определить, верно ли, что самый высокий мальчик весит больше всех в классе, а самая маленькая девочка является самой юной среди девочек.

4. Фирма ежегодно на протяжении n лет закупала оборудование стоимостью соответственно s_1, s_2, \dots, s_n рублей в год (эти числа вводятся и обрабатываются последовательно). Ежегодно в результате износа и морального старения (амортизации) все имеющееся оборудование уценивается на p%. Какова общая стоимость накопленного оборудования за n лет?

5. Дана непустая последовательность ненулевых целых чисел, за которой следует 0. Определить, сколько раз в этой последовательности меняется знак.

6. Выполнить задание без хранения последовательностей. Дано натуральное k . Определить k -ю цифру последовательности: 1248163264..., в которой выписаны подряд степени 2.
7. Составить алгоритм решения ребуса $KOT + KOT = ТОК$ (различные буквы означают различные цифры, старшая - не 0).
8. Натуральное число, записанное в десятичной системе счисления, называется сверхпростым, если оно остается простым при любой перестановке своих цифр. Определить все сверхпростые числа до n .

Тест "Циклические программы"

[В начало практикума](#)

Лабораторная работа № 6. Отладка программ

Процесс отладки программы состоит из многократных попыток выполнения программы на компьютере, анализа полученных результатов и исправления ошибок. Отладка позволяет обнаружить ошибки, которые появляются во время выполнения программы, т. е. выявить ситуации, при которых программа работает, но вычисляет значения неправильно.

Задание	Краткие теоретические сведения	
<p>1. Записать условие задачи для программы, представленной в правой части.</p> <p>Выполнить прокрутки для программ из лабораторной работы № 5.</p>	<pre>#include <stdio.h> void main() { float y, x = 0; while (x <= 6) { if (x < 4) y = x; else if (x > 4) y = x * x; else y = 1; printf("x=%f\t", x); printf("y=%5.2f\n", y); x = x + 2; } }</pre> <p>Выполнение программы закончено.</p>	<p>Термином <i>прокрутка</i> обозначают выполнение программы вручную так, как если бы она выполнялась на компьютере.</p> <p>Запись прокрутки для данного примера может выглядеть следующим образом:</p> <p>x = 0; 0 <= 6; 0 < 4; y = x = 0; вывод x = 0 и y = 0; x = x + 2 = 0 + 2 = 2 2 <= 6; 2 < 4; y = x = 2; вывод x = 2 и y = 2; x = x + 2 = 2 + 2 = 4 4 <= 6; 4 = 4; y = 1; вывод x = 4 и y = 1; x = x + 2 = 4 + 2 = 6 6 = 6; 6 > 4; y = x * x = 6 * 6 = 36; вывод x = 6 и y = 36; x = x + 2 = 6 + 2 = 8 8 > 6</p>

<p>2. Выполнить программу из п. 1 с использованием средства <i>интерактивной</i> отладки Шаг с обходом.</p> <p>Внести изменения в программу с тем, чтобы появились ошибки (например, заменить == на = или убрать строчку x = x + 2; и т. п.). Проверить, как будет выполняться отладка по шагам с обходом.</p>	<p>Пошаговое выполнение приложения может выполняться без трассировки вызываемых методов (Отладка / Шаг с обходом или клавиша F10 или иконка на панели инструментов Отладка;</p> <p>Если выбрана отладка Шаг с обходом, то производится останов перед выполнением текущей строки исходного кода. Выполнение программы продолжается повторным нажатием клавиши F10. При этом Отладчик не заходит в используемые программой функции такие, как библиотечные, функции пользователя и т.п.</p> <p>Если поместить курсор мыши на имя переменной и удерживать его некоторое время, то появится всплывающее окно, в котором будет указано текущее значение переменной.</p> <p>После отладки необходимо вывести приложение из отладочного режима командой Отладка / Остановить отладку или Shift+F5 или одноименной кнопкой на панели инструментов Отладка.</p> <p>В окне Видимые можно поместить курсор в столбец Имя, в контекстном меню выполнить команду Добавить контрольное значение и ввести имя нужной переменной (или выделить нужную переменную и перетащить ее мышкой в поле Имя). В поле Значение отображается значение переменной (объекта).</p> <p>В окне Видимые можно изменять значения переменных, за которыми ведется наблюдение. Для этого надо сделать двойной щелчок на отображаемом значении переменной и ввести новое.</p> <p>В том случае, когда ясно, что отображаемое значение не верно, можно установить корректное значение и продолжить поиск ошибок. Это средство можно использовать также для пропуска первых шагов в цикле с большим количеством итераций.</p> <p>Если рядом с именем переменной стоит знак плюс, то для этой переменной может быть отображена дополнительная информация (массивы, указатели или объекты класса). Если</p>

	<p>нажать F10 два раза и щелкнуть на символе «+» возле имени переменной, то Отладчик отобразит значение, хранимое в памяти по адресу, содержащемуся в указателе.</p>
<p>3. Выполнить программу из п. 1 с использованием средства интерактивной отладки Шаг с заходом.</p>	<p>Пошаговое выполнение приложения может выполняться с трассировкой вызываемых методов (Отладка / Шаг с заходом или клавиша F11 или иконка на панели инструментов Отладка).</p> <p>Если выбрана отладка Шаг с заходом, то производится останов перед выполнением текущей строки исходного кода. Выполнение программы продолжается повторным нажатием клавиши F11. При этом Отладчик заходит в используемые программой функции (библиотечные, функции пользователя и т.п.)</p>
<p>3. Выполнить программу п.1 с использованием средств планируемой отладки, установив точки останова на операторах вывода и пошагово проследив вывод значений на консоль.</p> <p>Удалить точки останова.</p>	<p>При планируемой отладке используются <i>точки останова</i>. Точка останова назначается щелчком левой кнопки мыши в сером поле слева от строки программы (точка останова отмечается маркером в виде красного круга). Повторный щелчок на маркере точки останова приводит к ее отмене. Точку останова можно назначить также с помощью Отладка / Точка останова или клавишей F9. Она устанавливается на той строке, где помещен курсор.</p> <p>Затем приложение запускается в режиме отладки (F5)или командой меню Отладка / Продолжить. Приложение будет выполнено до точки останова.</p> <p>Продолжение – клавиша F5.</p> <p>Во время одной сессии отладки могут использоваться все три вышеописанных способа: шаг с заходом, шаг обходом и точки останова.</p>
<p>4. Создать исполняемый файл без отладочной информации Release.</p>	<p>После отладки и исправления всех ошибок можно построить приложение без отладочной информации. Для переключения в окончательную конфигурацию необходимо выбрать команду Построение / Диспетчер конфигураций. На экран будет выведено</p>

<p>Объяснить назначение папок и файлов решения проекта.</p> <p>Выполнить файл с расширением *.exe в папке Debug.</p>	<p>диалоговое окно установки активной конфигурации проекта. Надо выбрать Win32 Release и повторить построение проекта с помощью Построение / Перестроить проект.</p> <p>Каждая конфигурация проекта определяет папки, куда будут помещены файлы с промежуточными и окончательными результатами компиляции и компоновки. По умолчанию это папки Debug и Release, которые располагаются в папке проекта. Папка решения содержит: файл текущего примера решения *.sln, файл с информацией о проектах решения и опциях решения *.suo. В файле решения зафиксирован перечень проектов, входящих в решение. Во вложенной папке находятся файлы и папки проекта: файл *.vcproj в формате XML содержит перечень файлов, включенных в проект; файл *.cpp содержит исходный код программы на языке C++; *.obj –объектные файлы, содержащие машинный код исходных файлов проекта; *.pch – предварительно скомпилированный файл заголовков; *.pdb – файл с отладочной информацией, используемой при выполнении программы в режиме отладки; *.idb – файл с информацией, необходимой для перестройки всего решения и др.</p> <p>Папка Debug используется для хранения временных файлов. В этой папке размещаются файлы с программным кодом на промежуточном языке. В частности, файл *.exe содержит программный код приложения.</p>
--	---

5. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства интерактивной отладки **Шаг с обходом** и **Шаг с заходом**. При наличии ошибок из-за некорректных исходных данных выполнить вычисления с другими числами. В отчете представить результаты в окне Отладчика (окно **Видимые**).

№	Формулы для вычислений	Исходные данные	№	Формулы для вычислений	Исходные данные
1	$y = \sqrt{a + m^2 \cdot x^2} / (a + x) \cdot m$ $z = \begin{cases} y + 1, & y < 1 \\ \sin^2 y, & y \geq 1 \end{cases}$	$a = 1,774;$ $m = 5;$ $x = 3(0,2)5$	9	$s = w / (w^2 - j) \cdot \ln w $ $y = \begin{cases} s - e^{f/i}, & s \geq 3\sqrt{f} \\ s^2 / w, & s < 3\sqrt{f} \end{cases}$	$f = 1,2;$ $j = -7 \cdot 10^{-3};$ $w = 2(0,5)5$
2	$d = \begin{cases} a + b \cdot i, & i > 3b \\ \operatorname{tg} b - a \cdot i, & i \leq 3b \end{cases}$ $z = (d \cdot a / 4) / (3a \cdot b - e^{1+d} / 100)$	$b = -0,05;$ $a = 1,72;$ $i = -5(2)5$	10	$w = \begin{cases} \sqrt{0,2x} \cdot k, & x < m/2 \\ e^{2 \cdot x \cdot k}, & x \geq m/2 \end{cases}$ $v = \sqrt{w^3 + x - a } / \ln(1 + a)$	$a = 1,2; k = 3$ $x = 2,5;$ $m = 4(0,2)6$
3	$s = e^{-a \cdot x} - \ln(i / x \cdot m) / \ln^2 m$ $p = \begin{cases} (-i \cdot s)^2, & s > 2x \\ \sin(-6s), & s \leq 2x \end{cases}$	$x = \{8; 1,9; -4; 1\};$ $a = 105 \cdot 10^{-4};$ $m = 4; i = 7$	11	$d = e^{-x} + (x - a) / \ln x$ $b = \begin{cases} 6d \cdot x, & d \geq k \cdot \sqrt{a} \\ \sin(a / x), & d < k \cdot \sqrt{a} \end{cases}$	$a = 1,33 \cdot 10^{-3};$ $k = 6;$ $x = \{0,2; 1; -3; -6,1\}$
4	$x = \cos^2(y) / (j + 2a \cdot y)$ $z = \begin{cases} e^{-j}, & x \geq \sqrt{y} \\ (0,5y / j)^2, & x < \sqrt{y} \end{cases}$	$y = 2,7; a = -5,5 \cdot 10^{-4};$ $j = \{6; -8,2; 15,4; 2\}$	12	$c = 2 \sin(f / 2) + \ln t$ $d = \begin{cases} y \cdot e^{-2t} + f, & c \geq 3 \\ y - y^3, & c < 3 \end{cases}$	$f = -125 \cdot 10^{-6};$ $y = 1,7; t = 1(0,1)2;$
5	$p = e^{\sin(j/x)} \cdot \ln(x/y) \cdot x$ $q = \begin{cases} \sqrt{p/m}, & p \leq y^2 \\ \sqrt{2x/(j+p)}, & p > y^2 \end{cases}$	$y = 1,4 \cdot 10^{-3};$ $x = 0,2; m = 4;$ $j = 5(2)11$	13	$y = \cos^2 x^2 / x $ $z = \begin{cases} y , & y < a \cdot x \cdot n \\ \sqrt{1 + e^{-y}}, & y \geq a \cdot x \cdot n \end{cases}$	$a = 1,055; x = 0,6;$ $n = 6(2)14$

№	Формулы для вычислений	Исходные данные	№	Формулы для вычислений	Исходные данные
6	$d = i + 2t \cdot (1 + \sqrt{3a^2})$ $f = \begin{cases} t \cdot i, & d \geq t(t+i) \\ e^{t-d} + 9, & d < t(t+i) \end{cases}$	$a = -4,2;$ $t = \{5; 1; -3; 9; -1\}$ $i = 4$	14	$v = \begin{cases} \cos^2(j \cdot x), & 2s > x \cdot j \\ 2 \cdot \operatorname{tg}(j \cdot x), & 2s < x \cdot j \\ 5 - e^{x/2}, & 2s = x \cdot j \end{cases}$	$s = 5,2;$ $x = \{5; 0,4; -2,1; 6\}$ $j = 10,4 \cdot 10^4$
7	$x = \operatorname{tg}(a^2 - 1)/(d + 1)$ $y = \begin{cases} a \cdot k + d, & 3x < a \cdot c \\ \cos(a \cdot k) \cdot e^{a+1}, & 3x \geq a \cdot c \end{cases}$	$c = 3,7(0,1)5;$ $d = 51,9 \cdot 10^{-2};$ $a = 4; k = 1$	15	$b = 12s - e^{-s/2} \cdot (x - j)$ $z = \begin{cases} \sqrt{-2x \cdot j} + b, & b < 1,5 \\ 13x \cdot j + b, & b \geq 1,5 \end{cases}$	$x = -4 \cdot 10^{-3}; s = 1,1;$ $j = 4(0,5)7$
8	$t = a \cdot i / (a^2 - b) \cdot e^{-a}$ $x = \begin{cases} 4,8 \cdot 10^{-3} + i \cdot a, & t > 5c \\ a + i^2 \cdot t, & t \leq 5c \end{cases}$	$b = 3; i = 8(4)24$ $c = 6 \cdot 10^{-4}; a = 6$	16	$z = \begin{cases} \sin^2(t), & t > x \\ 4 \cdot (t + x), & t < x \\ 1 - e^{x-2}, & t = x \end{cases}$	$t = 1,5(0,1)2;$ $x = 1,9$

6. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства планируемой отладки.

В отчете представить результаты в окне Отладчика (окно **Локальные**).

Создать исполняемый файл.

№ варианта	Условие задачи
1	Используя перебор значений вывести на экран в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр.

№ варианта	Условие задачи
2	Используя перебор значений найти сумму целых положительных чисел, кратных 3 и меньших 200.
3	Определить количество трехзначных натуральных чисел, сумма цифр которых равна n .
4	Вычислить наибольший общий делитель натуральных чисел a и b .
5	Построить первые N натуральных чисел, делителями которых являются только числа 2, 3 и 5.
6	Используя перебор значений вывести на экран в убывающем порядке все двузначные числа, в десятичной записи которых нет одинаковых цифр.
7	Даны натуральные числа m , n . Используя перебор значений получить все меньшие n натуральные числа, квадрат суммы цифр которых равен m .
8	Дано натуральное число n . Получить все его натуральные делители.
9	Вывести на экран в возрастающем порядке все четырехзначные числа, в десятичной записи которых нет одинаковых цифр.
10	Используя перебор значений найти сумму целых положительных чисел, кратных 5 и меньших 150.
11	Определить количество двузначных натуральных чисел, сумма цифр которых равна натуральному числу f .
12	Вычислить наибольший общий делитель натуральных чисел x , y и z .
13	Вывести первые 6 натуральных чисел, делителями которых являются числа 3 и 5.

№ варианта	Условие задачи
14	Используя перебор значений вывести на экран в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр.
15	Даны натуральные числа q и b . Используя перебор значений получить все меньшие q натуральные числа, квадрат суммы цифр которых равен b .
16	В заданной последовательности натуральных чисел (размер и значения вводятся пользователем) найти наибольший элемент из отрицательных.

7. Дополнительные задания.

1. Найти натуральное число, состоящее из трёх цифр, с возрастающими слева направо цифрами, являющееся полным квадратом. Число является полным квадратом, если квадратный корень из него есть простое число (число 121 – полный квадрат, т.к. $121=11*11$, а 11 – простое число).
2. Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50коп., 20коп., 5коп. и 2коп.
3. Имеются два сосуда. В первом сосуде находится C_1 литров воды, во втором – C_2 литров воды. Из первого сосуда переливают половину воды во второй сосуд, затем из второго переливают половину в первый сосуд, и так далее. Сколько воды окажется в обоих сосудах после 12 переливаний?
4. Три приятеля были свидетелями нарушения правил дорожного движения. Номер автомобиля – четырехзначное число – никто полностью не запомнил. Из показаний следует, что номер делится на 2, на 7 и на 11, в записи номера участвуют только две цифры, сумма цифр номера равна 30. Составить алгоритм и программу для определения номера автомашины.

5. Дана последовательность $a_n = n \sin n$. Найти сумму членов, номера которых записываются двузначными числами; сумму положительных членов из первых 100 членов; сумму тех из первых 100 членов, модули которых меньше 0,5.
6. Дано натуральное число n . Получить его каноническое разложение (разложение на простые множители).
7. Известно, что 1 января 2016 г. – пятница. Программа должна найти все «черные вторники» и «черные пятницы» 2016 и 2017 года (то есть – 13 числа).
8. Найти минимальное число, которое представляется суммой четырех квадратов натуральных чисел не единственным образом.

Пункт меню - Сборка

Тест "Отладка программ"

[В начало практикума](#)

Лабораторная работа № 7. Вычисление сумм, произведений, экстремумов

Задание	Краткие теоретические сведения
<p>1. Выполнить прокрутку и записать условие задачи для программы, представленной в правой части.</p> <p>Изменить программу так, чтобы вычислялась сумма квадратов четырех значений переменной a, произведение значений a.</p> <p>Произвести отладку всех вариантов.</p> <p>Написать программу вычисления факториала:</p> $5! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5$	<div style="display: flex; justify-content: space-between;"> <div style="width: 65%;"> <pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;">#include <iostream> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); float a, sum = 0; int i; const int size = 4; for (i = 1; i <= size; i++) { cout << "Введите a" << i << endl; cin >> a; sum = sum + a; } cout << "Ответ " << sum << endl; }</pre> </div> <div style="width: 30%;"> <p>Даны четыре значения переменной $\mathbf{a} = \{0,5; 44; -8; 11\}$. В процессе выполнения программы они вводятся с клавиатуры по одному.</p> </div> </div>

<p>2. Выполнить прокрутку и записать условие задачи для программы, представленной в правой части.</p> <p>Внести изменения в программу так, чтобы выводилось не только само значение переменной b, но и номер по порядку.</p>	<pre>#include <iostream> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); int COUNT = 5; float b, m = 9999; for (int i = 1; i <= COUNT; i++) { cout << "Введите b" << i << endl; cin >> b; if (b < m) m = b; } cout << "Ответ " << m << endl; }</pre>	<p>Даны пять значений переменной b = {9; -3,5; 54; -2,8; 40}.</p>
<p>3. Выполнить прокрутку и записать условия задач для программ, представленных в правой части.</p>	<pre>#include <stdio.h> #include <conio.h> void main() { char ch; //символ конца цикла float sv, x, sum = 0, count = 0; do { printf("Enter x:"); scanf_s("%f", &x); sum += x; count++; sv = sum / count; }</pre>	<pre>#include <stdio.h> #include <cmath> void main() { float sum = 0, a, t, p; for (int n = 2; n < 10; n++) { t = pow((float)n, log((float)n)); p = pow(log((float)n), (float)n); a = t / p; sum += a; }</pre>

	<pre> printf("sv=%1.3f\n", sv); printf("if continue input 'y' else 'n' "); ch = _getch(); } while (ch != 'n'); } </pre>	<pre> } printf("S=%f\n", sum); } </pre>
--	---	---

4. В соответствии со своим вариантом разработать программу по условию, приведенному в таблице ниже. Произвести отладку программы.

5. Написать программы еще для нескольких (лучше для всех) условий из этой же таблицы.

№	Формула для вычислений	Исходные данные	№	Формула для вычислений	Исходные данные
1	$h = d + \sum_{i=1}^n a_i^2$	$n = 5; d = 12,5 \cdot 10^{-4};$ $a = \{0,8; 4; -7; 2; 0,91\}$	9	$a_i = (e^{-x_i} \cdot \sin x_i) / (\sqrt{d + \cos x_i})$	$d = 5,5 \cdot 10^{-4};$ $x = \{0,7; 6; -7; 9; -4; 1\}$
2	$g = c \cdot \prod_{j=1}^m (b_j + 1)^2$	$m = 4; c = -0,045;$ $b = \{0,9; 0,5; -2; -0,1\}$	10	$q = \sum_{i=1}^n (x_i \cdot y_i)$	$n = 6;$ $x = \{3; -2; 0,7; -1; -2; 7\}$ $y = \{1; 5; -1,2; 6; 9; -4\}$
3	$z = 8 \cdot x_3 + \sum_{i=1}^m (x_i - 2)^2$	$m = 5;$ $x = \{-2; 6; 1,1; 2,7; 4\}$	11	$s = 2a + q \cdot \sin a$ $q = 4 \prod_{i=1}^n y_i / (i^2 + 1)$	$a = 5,45; n = 5;$ $y = \{2,1; 7,7; -4; 5; 9\}$
4	$f = k$ $g = f / \sum_{i=1}^5 (a_i / i)$	$k = 6(0,2)7;$ $a = \{2,3; 7; -2; -4; 9\}$	12	$z = \max\{x_i\}$ $y = z \cdot \sum_{i=1}^5 x_i^2$	$x = \{1; 2,7; 4,7; 6; 10\}$

№	Формула для вычислений	Исходные данные	№	Формула для вычислений	Исходные данные
5	$w_i = \begin{cases} a + v_i, & v_i > 0 \\ b / v_i, & v_i \leq 0 \end{cases}$	$a = 1,5; b = -4,15;$ $v = \{1; 1,5; -4; -1,9; 3\}$	13	$p = \min\{y_i\}$ $q = \prod_{i=1}^n (y_i - 5) + p$	$y = \{3; -2; 0,9; 0,5; 1\}$
6	$a = \sum_{i=1}^n x_i, \quad x_i > 0$ $b = \prod_{i=1}^n x_i, \quad x_i \leq 0$	$n = 5;$ $x = \{2,7; -5; 4; 3,5; -7\}$	14	$z = \begin{cases} \sin^2 c, & \sum_{i=1}^6 y_i > c \\ \cos^2 c, & \sum_{i=1}^6 y_i \leq c \end{cases}$	$y = \{4; -6; 3; -3; 9; 5\}$ $c = 10,1$
7	$c_i = \sqrt{a_i + b_i}$ $d = \sum_{i=1}^n (c_i / i)$	$n = 6;$ $a = \{0,5; 2; 2,5; 1; 0; 8\}$ $b = \{2,3; 4; 0,5; 2; 3; 9\}$	15	$q = t + \frac{x_1 + 1}{x_1} + \dots + \frac{x_6 + 1}{x_6}$	$t = 0,45;$ $x = \{1,1; 6,2; 3; -4; 6; 1\}$
8	$d = \sum_{i=1}^{n_1} a_i - c \cdot \sum_{i=1}^{n_2} (b_i - 1)^2$	$n_1 = 6; n_2 = 5; c = 0,7;$ $a = \{3; 12; -4; 6,2; 3; 0,4\}$ $b = \{19; 1; -24; 4,2; 8\}$	16	$y = \begin{cases} \sum_{i=1}^5 x_i^2 + t, & t > 2 \\ \cos(t^2), & t \leq 2 \end{cases}$	$x = \{8; -2,3; -8,4; 5,1; 9\}$ $t = 0,5(0,5)3$

6. Дополнительные задания.

1. Найти в последовательности n целых чисел и вывести значение суммы четных элементов.
2. Найти в последовательности n целых чисел и вывести порядковый номер первого отрицательного элемента.
3. Найти в последовательности n чисел с плавающей точкой количество элементов стоящих между минимальным и максимальным значениями.

4. В последовательности найти число чередований знака, то есть число переходов с минуса на плюс или с плюса на минус. Пример: в последовательности 0, -2, 0, -10, 2, -1, 0, 0, 3, 2, -3 четыре чередования (как известно, нуль не имеет знака).
5. Последовательность a_1, a_2, \dots, a_k называется пилообразной, если $a_1 < a_2 > a_3 < a_4 > \dots > a_k$ либо $a_1 > a_2 < a_3 > a_4 < \dots < a_k$. Проверить является ли заданная последовательность пилообразной.
6. Найти в последовательности n целых чисел сумму положительных и произведение отрицательных.
7. Найти в последовательности n целых чисел максимальный положительный и максимальный отрицательный элементы.
8. Найти в последовательности n натуральных чисел среднее арифметическое нечетных элементов.

Тест "Суммы, произведения, экстремумы"

[В начало практикума](#)

Лабораторная работа № 8. Решение инженерных задач на основе циклических программ

В инженерной практике часто встречаются задачи, для решения которых требуется использовать численные методы. Рассмотрим способы использования циклических алгоритмов для вычисления определенных интегралов (задачи определения площадей различных фигур, нахождения объемов тел вращения и др.) и для решения уравнений.

Задание	Краткие теоретические сведения
1. Изучить алгоритм вычисления определенного интеграла методом трапеций.	<p>Приближенное вычисление определенного интеграла основано на геометрическом смысле интеграла и сводится к приближенному вычислению площади, ограниченной графиком подынтегральной функции $f(x)$, прямыми $x = a = x_0$, $x = b = x_n$ и осью OX. Интервал $[a, b]$ делится на n равных частей длиной $h = (b - a) / n$. Тогда значениям $x_i = x_{i-1} + h, i = 1, 2, ..., n$ соответствуют значения $y_i = f(x_i)$.</p> <p>Согласно <i>методу трапеций</i> значение интеграла вычисляется как сумма площадей трапеций, высоты которых равны h, а основания соответственно y_0 и y_1 – для первой трапеции, y_1 и y_2 – для второй и т. д.</p> $\int_a^b f(x)dx \approx h \frac{y_0 + y_1}{2} + h \frac{y_1 + y_2}{2} + ... + h \frac{y_{n-1} + y_n}{2} = \frac{h}{2} (y_0 + 2(y_1 + y_2 + ... + y_{n-1}) + y_n)$ <p>Алгоритм вычисления интеграла по методу трапеций.</p> <ol style="list-style-type: none">1. Ввод a, b, n.2. Вычисление $h = (b - a) / n, x = a, s = 0$.3. Расчет $s = s + h \cdot (f(x) + f(x + h)) / 2, x = x + h$.4. Если $x > (b - h)$, то переход к пункту 5, иначе – переход к пункту 3.5. Вывод z.

<p>2. Изучить алгоритм вычисления определенного <i>интеграла</i> методом <i>парабол</i>.</p>	<p>При использовании <i>метода парабол</i> интервал $[a, b]$ делится на четное количество частей – $2n$. Тогда $h = (b - a) / (2n)$, $x_i = x_{i-1} + h$, $i = 1, 2, \dots, 2n$.</p> $\int_a^b f(x)dx \approx \frac{h}{3}(y_0 + 4(y_1 + y_3 + \dots + y_{2n-1}) + 2(y_2 + y_4 + \dots + y_{2n-2}) + y_{2n}).$ <p>Алгоритм метода парабол</p> <ol style="list-style-type: none"> 1. Ввод a, b, n. 2. Вычисление $h = (b - a) / (2n)$, $x = a + 2h$, $s1 = 0$, $s2 = 0$, $i = 1$. 3. Расчет $s2 = s2 + f(x)$, $x = x + h$, $s1 = s1 + f(x)$, $x = x + h$, $i = i + 1$. 4. Если $i < n$, то переход к пункту 3, иначе – переход к следующему пункту. 5. Вычисление значения интеграла: $z = \frac{h}{3}(f(a) + 4f(a + h) + 4s1 + 2s2 + f(b))$ <ol style="list-style-type: none"> 6. Вывод z. <p>Здесь $s1 = y_3 + y_5 + \dots + y_{2n-1}$, а $s2 = y_2 + y_4 + \dots + y_{2n-2}$.</p>
<p>3. Изучить способ отделения корней и метод касательных для решения уравнения.</p>	<p>Решение уравнения численными методами состоит из двух этапов: <i>отделение</i> корней, т. е. нахождение таких отрезков $[a, b]$ на оси ОХ, внутри которых имеется один корень; <i>вычисление</i> корней с заданной точностью.</p> <p>Отделить корни можно, построив, например, график функции $f(x)$ в приложении Excel. Из тех отрезков, на которых функция пересекает ось ОХ, рассмотрим один $[a, b]$.</p> <p>При использовании <i>метода касательных</i> для вычисления корня уравнения $f(x) = 0$ необходимо определить начальное приближение корня x_0: $x_0 = a$, если знаки $f(a)$ и $f''(a)$ совпадают, и $x_0 = b$, если знаки $f(b)$ и $f''(b)$ совпадают.</p>

	<p>Последовательные приближения корня рассчитываются по формуле</p> $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, n = 0, 1, 2, \dots$ <p>Вычисления продолжаются пока не будет выполнено условие: $x_{n+1} - x_n \leq e$, где e – требуемая точность вычисления корня.</p> <p><i>Алгоритм метода касательных</i></p> <ol style="list-style-type: none"> 1. Ввод значений a, b, e. 2. Вычисление начального приближения корня $x_1 = a$, если $f(a)f'(a) > 0$ или $x_1 = b$ в противном случае. 3. Вычисление $x = x_1$. 4. Определение очередного приближения корня по формуле $x_1 = x - \frac{f(x)}{f'(x)}$. 5. Если $x_1 - x > e$, то переход к пункту 3, в противном случае – переход к пункту 6. 6. Вывод значения корня x_1.
<p>4. Изучить метод <i>дихотомии</i> для решения уравнения.</p>	<p>Согласно <i>методу дихотомии</i> отрезок $[a, b]$ делится пополам. Из полученных двух отрезков для дальнейших вычислений выбирается тот, на концах которого функция $f(x)$ имеет разные знаки. Выбранный отрезок вновь делится пополам. Вычисления продолжаются до тех пор, пока величина последнего из полученных отрезков не станет меньше $2e$.</p> <p><i>Алгоритм метода дихотомии</i></p> <ol style="list-style-type: none"> 1. Ввод значений a, b, e. 2. Вычисление $x = (a + b)/2$. 3. Если $f(x)f(a) \leq 0$, то $b = x$, иначе – $a = x$. 4. Если $a - b > 2e$, то переход к пункту 2, иначе – переход к следующему пункту.

	5. Вывод значения корня x .
5. Изучить способ нахождения корня уравнения в приложении Excel с помощью команды Подбор параметра .	<p>В приложении Excel для решения уравнения имеется команда Подбор параметра. Чтобы решить с помощью этой команды например уравнение $4 - x^2 + x = 0$, надо на рабочем листе, например в ячейке A1, записать начальное приближение корня (например, 2) , в ячейке B1 – само уравнение: $= 4 - A1^2 + A1$</p> <p>Выполнить Данные / Работа с данными / Анализ “что-если” / Подбор параметра.</p> <p>В появившемся окне задать следующие значения: в поле Установить в ячейке выбрать B1, в поле Значение ввести 0, в поле Изменяя значение ячейки – A1.</p> <p>После нажатия ОК в ячейке A1 будет корень уравнения.</p>

6. В соответствии со своим вариантом написать программы для вычисления определенного интеграла из таблицы, приведенной ниже, методом трапеций и методом парабол. Для всех вариантов принять $n = 200$. Сравнить результаты.

Номер вариан та	Подынтегральная функция $f(x)$	Пределы интегрирова ния	Номер вариан та	Подынтегральная функция $f(x)$	Пределы интегрирования
1	$x^3 - 3$	$a = 1, b = 3$	9	$x^3 + 3$	$a = 3, b = 6$
2	$\cos^3(x)$	$a = 4, b = 7$	10	$x^2 - 4$	$a = 4, b = 8$
3	$1 + x^3$	$a = 1, b = 6$	11	$\sin(x) + 1$	$a = 1, b = 3$
4	$e^x - 1 / x$	$a = 2, b = 3$	12	$e^x + 2$	$a = 5, b = 11$
5	$5 - x^2$	$a = 8, b = 12$	13	$2 + x^3$	$a = 8, b = 14$
6	$1 + x^3$	$a = 1, b = 5$	14	$x^4 + 4$	$a = 1, b = 4$
7	$e^x + 2$	$a = 5, b = 11$	15	$\sin^2(x) + 1$	$a = 2, b = 7$
8	$x^3 - 1$	$a = 0, b = 3$	16	$x^2 + 1 / x$	$a = 0, b = 4$

7. В соответствии со своим вариантом отделить корни уравнения из таблицы, приведенной ниже. Если корней несколько, то для одного из отрезков, содержащих корень (то есть при некоторых найденных значениях **a** и **b**), написать программы вычисления корня уравнения методом касательных и методом дихотомии. Точность вычислений принять равной $\epsilon = 0,0001$ для всех вариантов.

Найти корень уравнения с помощью приложения Excel.
Сравнить все результаты.

Номер вариан та	Подынтегральная функция $f(x)$	Номер вариан та	Подынтегральная функция $f(x)$
1	$x^3 + x - 3$	9	$x^3 + 3x - 1$
2	$\cos(x) + x - 7$	10	$x^3 + x - 4$
3	$x^3 + 2x - 1$	11	$\sin(x) + x^3$
4	$e^x - 3 - 1/x$	12	$e^x + 2x^2 - 3$
5	$2 - x^2 + x$	13	$2x + x^3 - 7$
6	$5x - 1 + x^3$	14	$x^3 + 2x - 4$
7	$e^x + x - 4$	15	$\sin(x) + 2 + x$
8	$x^3 + x - 2$	16	$x^2 + 4x - 2$

Контрольная работа

[В начало практикума](#)

Лабораторная работа № 9. Одномерные массивы

Массив – это группа связанных друг с другом элементов одного типа (**double**, **float**, **int** и т.п.) последовательно расположенных в памяти. Пример объявления массива в программе: **int a[6]**;

Нумерация элементов массива начинается с **нуля** и заканчивается **n – 1**, где **n** – число элементов массива.

Задание	Краткие теоретические сведения	
<p>1. Выполнить программу, записанную в правой части. Опробовать второй вариант генерации чисел, записанный в комментарии.</p> <p>Добавить в программу операторы вычисления суммы элементов массива A.</p> <p>Произвести отладку.</p>	<pre>#include <iostream> #include <ctime> void main() { setlocale(LC_CTYPE, "Russian"); using namespace std; const int N = 100; int i, sz, A[N]; int rmn = 0, rmx = 99; cout << "Введите размер массива "; cin >> sz; cout << "Массив A:" << endl; srand((unsigned)time(NULL)); for (i = 0; i < sz; i++) { A[i] = (int)((double)rand() / (double)RAND_MAX) * (rmx - rmn) + rmn); // или A[i] = rand() % rmx; - для [0, rmx) cout << A[i] << endl; } }</pre>	<p><u>Пример.</u> Сформировать одномерный статический массив целых чисел A, используя датчик случайных чисел (в диапазоне от 0 до 99).</p> <p>В программе массив заполняется случайными числами с помощью функции rand из стандартной библиотеки. Она генерирует целое число в диапазоне от 0 до RAND_MAX (символическая константа, равная 32767).</p> <p>При каждом запуске программы будут</p>

	<p>генерироваться одни и те же случайные числа.</p> <p>Чтобы числа были разными нужно использовать библиотечную функцию srand, которая задает начальное число для функции rand. Для получения начального числа считываются показания часов с помощью функции time, которая возвращает текущее время в секундах.</p> <p>При использовании функции time нужно включить в заголовок директивы #include <ctime> или #include <locale></p>
<p>2. Выполнить программу, записанную в правой части. Записать ее условие.</p> <p>Добавить в программу операторы вычисления среднего значения исходного массива.</p> <p>Произвести отладку.</p>	<div data-bbox="607 446 1594 1065"><pre>#include <iostream> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); int i, k, size = 4; float A[] = { 5, -4, 17.1, 9, 1 }; cout << "Введите номер элемента (от 0 до 4) "; cin >> k; for (i = k; i <= size; i++) A[i] = A[i + 1]; size--; for (i = 0; i <= size; i++) cout << A[i] << endl; }</pre></div> <div data-bbox="1594 446 2101 1065"><p>Инициализация массива означает присвоение начальных значений его элементам при объявлении.</p><p>Массивы можно инициализировать списком значений, заключенных в фигурные скобки, например в программе:</p><pre>float A[] = {5, -4, 17.1, 9, 1};</pre><p>Длина массива вычисляется компилятором по количеству значений, перечисленных в фигурных скобках.</p></div>

3. Выполнить прокрутку программы, приведенной в правой части, для любых исходных данных и записать условие задачи.

Опробовать программу. Реализовать условие задачи для массивов **A** и **B** разного размера.

Произвести отладку.

Пример. Даны два массива целых чисел **A** и **B** размера 5, элементы которых предварительно упорядочены по возрастанию. Сформировать массив **C** ...

Массив C не будет упорядоченным!

```
#include <stdio.h>
void main()
{
    const int size = 5;
    int a[size], b[size], c[size*2];
    int k = 0, l = 0, i = 0;
    printf("A:\n");
    for (int n = 0; n < size; n++)
        scanf_s("%d", &a[n]);
    printf("B:\n");
    for (int n = 0; n < size; n++)
        scanf_s("%d", &b[n]);
    do
    {
        if (a[k] <= b[l])
        {
            c[i++] = a[k++];
        }
        else
        {
            c[i++] = b[l++];
        }
    }
    if (k == size)
        for (; l < size; l++)
```

	<pre> c[i++] = b[l]; if (l == size) for (; k < size; k++) c[i++] = a[k]; } while (i < size*2); printf("\n"); for (i = 0; i < size*2; i++) printf("%d ", c[i]); printf("\n"); }</pre>	
--	--	--

<p>4. Выполнить программу, приведенную в правой части.</p> <p>Внести изменения с тем, чтобы вычислялся минимальный элемент массива.</p> <p>Произвести отладку.</p>	<pre> #include <locale> #include <iostream> void main() { setlocale(LC_ALL, "rus"); using namespace std; const int maxSize = 30; int n, i, a[maxSize], kmax = 0; cout << "Введите размер массива (не более 30)" << endl; cin >> n; if (n > 30) return; srand((unsigned)time(NULL)); for (i = 0; i < n; i++) { a[i] = rand() % 30; cout << a[i] << " "; } cout << endl; for (i = 1; i < n; i++) if (a[i] > a[kmax]) kmax = i; cout << "Максимальный элемент " << a[kmax] << endl; } </pre>	<p>Пример.</p> <p>Сформировать массив целых чисел в количестве не более 30. Размерность массива ввести с клавиатуры.</p> <p>Найти в массиве наибольший элемент.</p>
--	---	---

5. Сформировать одномерный массив целых чисел, используя датчик случайных чисел (диапазон от 0 до 99). Размер массива ввести с клавиатуры. В соответствии со своим вариантом написать программу по условию, представленному в таблице ниже. Составить блок-схему алгоритма.

№ варианта	Условие задачи
1	Удалить элемент с номером k . Добавить после каждого четного элемента массива элемент со значением 0.
2	Все четные элементы целочисленного массива $K(n)$ поместить в массив $L(n)$, а нечетные – в массив $M(n)$. Подсчитать количество тех и других.
3	Удалить элементы, индексы которых кратны 7. Добавить после каждого нечетного элемента массива элемент со значением 4.
4	Поменять местами минимальный и максимальный элементы массива.
5	В массиве C каждый третий элемент заменить полусуммой двух предыдущих. Дополнительный (рабочий) массив не использовать.
6	Удалить все элементы с заданным значением, если они имеются в массиве. Добавить перед каждым четным элементом массива элемент со значением 1.
7	Удалить из массива все элементы, совпадающие с его минимальным значением. Добавить в начало массива три элемента со значением, равным среднему арифметическому массива.
8	Найти максимальный элемент массива и заменить им нечетные по номеру элементы.
9	Найти в массиве элемент, наиболее близкий к среднему арифметическому суммы его элементов.
10	Найти в массиве элемент, если он существует, равный среднему арифметическому суммы трех его последних элементов.

№ варианта	Условие задачи
11	Удалить пять первых нечетных элементов массива. Добавить в конец массива три новых нулевых элемента
12	Найти минимальный элемент массива T и заменить им четные по номеру элементы.
13	В массиве A(n) каждый элемент, кроме первого, заменить суммой всех предыдущих.
14	В массиве несколько нулевых элементов. Найти первый и последний нулевые элементы. Вывести их индексы.
15	Удалить элементы, индексы которых кратны 3. Добавить после каждого отрицательного элемента массива элемент со значением 10.
16	В массиве найти первый минимальный и первый максимальный элементы. Вывести их индексы.

7. К номеру своего варианта прибавить 2 и написать программу для новых исходных данных (для вариантов с 15 по 16 перейти к вариантам с 1 по 2). Представить результаты в окне Отладчика.

8. Дополнительные задания.

1. Имеются результаты n ежедневных измерений количества выпавших осадков. За какую из недель (отрезок времени длиной 7 дней), считая с начала периода измерений, выпало наибольшее количество осадков?
2. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами.
3. Подсчитать количество пар соседних элементов массива с одинаковыми значениями.
4. Найти в массиве наибольшее число подряд идущих одинаковых элементов.

5. В массиве $M(k)$ много совпадающих элементов. Найти количество различных элементов в нем (не упорядочивая массива).
6. Определить, содержится ли в заданном массиве хотя бы одно число Фибоначчи.
7. Разделить массив на две части, поместив в первую элементы, большие среднего арифметического их суммы, а во вторую - меньшие (части не сортировать).
8. В массиве $A(2n+1)$, не содержащем одинаковых элементов, найти средний по величине элемент, то есть такой, что в массиве A ровно n элементов меньше его и столько же элементов больше его. Массив A сохранить (не сортировать), дополнительных массивов не использовать

Тест "Одномерные массивы"

[В начало практикума](#)

Лабораторная работа № 10. Битовые операции

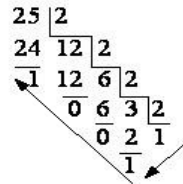
При организации вычислительных процессов в компьютерах используются десятичная система счисления (с/с), двоичная, восьмеричная и шестнадцатеричная.

Пусть надо перевести число $a_1...a_n$, где $a_1, . . . , a_n$ – цифры этого числа, из системы счисления с основанием k в десятичную систему счисления. Данное число можно представить в виде: $a_1 \cdot k^{n-1} + a_2 \cdot k^{n-2} + ... + a_n \cdot k^0$. Вычисление суммы даст нужный результат.

Пример перевода из двоичной в десятичную систему счисления:

$$110011_{(2 \text{ с/с})} = 1 \cdot 10^{101} + 1 \cdot 10^{100} + 0 \cdot 10^{11} + 0 \cdot 10^{10} + 1 \cdot 10^1 + 1 \cdot 1^0_{(2 \text{ с/с})} = \\ = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0_{(10 \text{ с/с})} = 32 + 16 + 2 + 1 = 51_{(10 \text{ с/с})}.$$

При переводе из десятичной системы счисления в двоичную, исходное число делится на основание с/с, т. е. на число 2, фиксируется остаток от деления и частное. Затем частное нужно снова разделить на основание с/с и зафиксировать остаток от деления. Процесс деления частных продолжать до тех пор, пока частное не станет меньше основания с/с. Все полученные в процессе деления остатки от деления и последнее частное будут образовывать цифры нужного результата в обратном порядке.



Например:

$$25_{(10 \text{ с/с})} = 11001_{(2 \text{ с/с})} = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 25_{(10 \text{ с/с})}.$$

Десятич- ная с/с	Двоич- ная с/с	Восьме- рич- ная с/с	Шестнад- цатерич- ная с/с
0	0	0	0
1	1	1	1
2	10	2	2
3	11	3	3
4	100	4	4
5	101	5	5
6	110	6	6
7	111	7	7
8	1000	10	8
9	1001	11	9
10	1010	12	A
11	1011	13	B
12	1100	14	C
13	1101	15	D
14	1110	16	E
15	1111	17	F
16	10000	20	10
17	10001	21	11

На языке C/C++ предусмотрены битовые операции для работы с отдельными битами. Их **нельзя** применять к переменным **вещественного** типа. Основные битовые операции:

- **AND (и)** – **&** – если какой-то бит в одном из операндов равен 0, то результирующий бит тоже будет равен 0;
- **OR (или)** – **|** – если какой-то бит в одном из операндов равен 1, то результирующий бит тоже будет 1;
- **XOR (исключающее или)** – **^** – результирующий бит равен 1, если сравниваемые биты различны;
- **NOT (не)** – **~** – меняются все биты на противоположные;
- **сдвиг влево** – **<<** – удваивается значение; – **сдвиг вправо** – **>>** – значение уменьшается в два раза.

Задание	Краткие теоретические сведения	
1. Изучить использование битовых операций и маскирования числа, опробовав программу, записанную в правой части, с различными исходными числами.	<pre>#include <iostream> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); unsigned int value; int i; const unsigned int mask = 1 << 31; cout << "Введите целое число "; cin >> value; cout << "Двоичный вид: "; for (i = 1; i <= 32; i++) { putchar(mask & value ? '1' : '0'); value <= 1; if (i % 8 == 0) putchar(' '); } }</pre>	<p><u>Пример</u> программы, печатающей тридцатидвухразрядное двоичное представление целого числа, введенного с клавиатуры.</p> <p>Здесь используется маскирование всех битов числа, за исключением текущего, выводимого на печать. Поскольку представление содержит 32 бита, то маска имеет вид: 10000000 00000000 00000000 00000000, т.е. 1 << 31.</p> <p>Последовательно применяется маска и сдвигается число на разряд влево.</p>

2. Выполнить программу, записанную в правой части. Ознакомиться с результатом. Опробовать программу, изменяя различные биты различных чисел.

```
#include <iostream>
using namespace std;
void main()
{
    setlocale(LC_CTYPE, "Russian");
    int A = 150; char tmp[33];
    _itoa_s(A, tmp, 2);
    cout << " Число A: " << tmp << endl;
    _itoa_s(0x24, tmp, 2);
    cout << " Маска для A: " << tmp << endl;
    _itoa_s(A | 0x24, tmp, 2);
    cout << " Результат: " << tmp << endl <<
endl;
}
```

Пример. Установить в единицу каждый **третий** бит числа A считая справа.

Здесь для вывода двоичного представления числа используется стандартная функция: _itoa_s (число ввода, строка вывода, основание с/с).

3. В программе, записанной в правой части, используются различные битовые операции.

Внести изменения в программу с тем, чтобы проверялось число на кратность четырем.

```
#include <iostream>
using namespace std;
void main()
{
    setlocale(LC_CTYPE, "Russian");
    int A, i; char tmp[33];
    cout << "Введите число ";
    cin >> A;
    _itoa_s(A, tmp, 2);

    cout << "Число в двоичном виде = " << tmp << endl;
    if ((A & 7) == 0)
        cout << "Число кратно 8" << endl;
    else
        cout << "Число не кратно 8" << endl;
}
```

Пример. Пусть имеется некоторое целое число. Вывести его двоичное представление и проверить, кратно ли оно восьми.

<p>4. В правой части приведен пример программы, демонстрирующей использование битовых операций.</p> <p>Проанализировать текст программы и написать пояснения.</p>	<pre> #include <iostream> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); char tmp[33]; int A, B, maskA = 14; int maskB = ~maskA >> 1; cout << "Первое число="; cin >> A; cout << "Второе число="; cin >> B; _itoa_s(A, tmp, 2); cout << "A=" << tmp << endl; _itoa_s(B, tmp, 2); cout << "B=" << tmp << endl; _itoa_s(maskA, tmp, 2); cout << "Маска для A: " << tmp << endl; _itoa_s((A & maskA) >> 1, tmp, 2); cout << "Выделенные биты A: " << tmp << endl; _itoa_s(maskB, tmp, 2); cout << "Маска для B: " << tmp << endl; _itoa_s(B & maskB, tmp, 2); cout << " Очищены биты в B: " << tmp << endl; _itoa_s(((B & maskB) ((A & maskA) >> 1)), tmp, 2); cout << " Результат B=" << tmp << endl; } </pre>	<p><u>Пример.</u> Извлечь 3 бита числа А, начиная со второго справа и вставить их в число В, начиная с первого.</p>
---	---	---

5. В соответствии со своим вариантом разработать программы, использующие битовые операции для решения задач, представленных в таблице.

Результаты одной из программ представить в Отладчике.

№ варианта	Условия задач
1	<ol style="list-style-type: none"> 1. Ввести целое A и посчитать, сколько нулей в числе начиная с третьего бита по 13, включая эти биты. 2. Инвертировать в числе A n битов вправо от позиции p, заменить ими m битов числа B, начиная с позиции q.
2	<ol style="list-style-type: none"> 1. Извлечь 5 битов числа A, начиная со второго и вставить их в число B, начиная с третьего бита. 2. Установить в 1 в числе A n битов вправо от позиции p.
3	<ol style="list-style-type: none"> 1. Ввести целое число A. Инвертировать все биты с 2 по 14, включая эти биты. Вывести результат. 2. Инвертировать в числе A n битов влево от позиции p, заменить ими m битов числа B, начиная с позиции q.
4	<ol style="list-style-type: none"> 1. Используя битовые операции проверить, кратно ли четырем число A. 2. Установить в 1 в числе A n битов влево от позиции p.
5	<ol style="list-style-type: none"> 1. Определить, насколько в числе A больше значащих битов, равных единице, чем битов, равных нулю. 2. Установить в 1 в числе A n битов вправо от позиции p, заменить ими m битов числа B, начиная с позиции q.
6	<ol style="list-style-type: none"> 1. Установить в единицу каждый второй значащий бит целого числа A. 2. Извлечь 3 бита числа A, начиная с позиции n, и вставить в число B, начиная с позиции m.
7	<ol style="list-style-type: none"> 1. Извлечь 4 бита числа A, начиная с пятого, и добавить их к числу B справа. 2. Установить в 1 в числе A n битов влево от позиции p, заменить ими m битов числа B, начиная с позиции q.
8	<ol style="list-style-type: none"> 1. Установить в ноль каждый третий значащий бит целого числа A. 2. Извлечь 3 бита числа A, начиная с позиции n, и вставить в число B, начиная с позиции m.
9	<ol style="list-style-type: none"> 1. Извлечь 5 битов числа A, начиная с третьего и вставить их в число B, начиная со 2. 2. Установить в 0 в числе A n битов вправо от позиции p.
10	<ol style="list-style-type: none"> 1. Вывести 6 бит целого числа A, начиная со 2-ого. 2. Инвертировать в числе A n битов влево от позиции p.

№ варианта	Условия задач
11	1. Используя битовые операции проверить, кратно ли шестнадцати число А. 2. Установить в 0 в числе А n битов влево от позиции p, заменить ими m битов числа В, начиная с позиции q.
12	1. Ввести целое число А. Инвертировать все биты с 4 по 8, включая эти биты. Вывести полученное число. 2. Установить в 0 в числе А n битов вправо от позиции p, заменить ими m битов числа В, начиная с позиции q.
13	1. Ввести целое число А. Извлечь 2 бита числа А, начиная с пятого и вставить их в число В, начиная также с пятого бита. 2. Инвертировать в 1 в числе А n битов вправо от позиции p.
14	1. Ввести целое число А и посчитать, сколько единиц в числе с 5 по 10 бит, включая эти биты. 2. Извлечь 3 бита числа А, начиная с позиции n, и вставить в число В, начиная с позиции m, заменить ими m битов числа В, начиная с позиции q.
15	1. Используя битовые операции проверить, кратно ли двум число А. 2. Установить в 0 в числе Аn битов влево от позиции p.
16	1. Ввести целое число А. Извлечь 3 бита числа А, начиная со второго и вставить их в число В, начиная с первого бита. 2. Установить в 12 бита числа А, начиная с четвертого.

6. К номеру своего варианта прибавить 1 и написать программы для новых исходных данных (для варианта 16 перейти к варианту 1).

4-ый вопрос – нумерация с нуля идет

Тест "Битовые операции"

[В начало практикума](#)

Лабораторная работа № 11. Указатели и ссылки

Задание	Краткие теоретические сведения	
1. Изучить работу с указателями, выполнив программы, записанные в правой части.	<p>Указатель – это переменная, значением которой является адрес ячейки памяти.</p> <p>В памяти компьютера указатель обычно занимает 4 байта (однако он зависит от конкретной реализации компилятора, размер указателей различных типов также может быть различным.)</p> <p>Пусть объявлена переменная и указатель на нее: int a = 0; int *ptr; Можно в указатель поместить адрес переменной и напечатать: ptr = &a; cout << ptr;</p> <p>Здесь & – операция получения адреса переменной, * – операция разыменования (получения значения переменной, на которую указывает указатель).</p>	
	Вывод значений переменных и указателей	Суммирование значений массива a
	<pre>#include <iostream> using namespace std; void main() { int a = 10, *pa, b = 20, *pb; pa = &a; cout << &a << " " << a << endl; cout << pa << " " << *pa << endl; pb = &b; cout << pb << " " << *pb << endl; *pa = *pb; cout << &a << " " << a << endl; }</pre>	<pre>#include <iostream> void main() { using namespace std; float a, *pa, s = 0; int i; pa = &a; for (i = 1; i <= 4; i++) { cout << "a="; cin >> a; s+= *pa; } cout << "s=" << s << endl; }</pre>

	Вывод адресов переменных производится в шестнадцатеричной системе счисления.		
<p>2. В правой части записаны фрагменты программ с использованием указателя на константу, константного указателя, константного указателя на константу.</p> <p>Убрать ошибочные операторы, дописать операторы вывода и выполнить программы на компьютере.</p>	Указатель на константу	Константный указатель	Константный указатель на константу
	<pre>{ int a = 9; const int *pa = &a; *pa = 12; // ошибка (нельзя менять значение переменной a) int b = 20; pa = &b; }</pre>	<pre>{ int a = 99; int* const pa = &a; int b = 44; pa = &b; // ошибка (константный указатель менять нельзя) *pa = 12; }</pre>	<pre>{ int a = 99; const int* const pa = &a; *pa = 33; // ошибка (нельзя менять содержимое переменной a) int b = 44; pa = &b; // ошибка (константный указатель на константу менять нельзя) }</pre>
<p>3. Написать программу генерации элементов массива A из случайных чисел, их вывода и определения максимального элемента массива двумя способами (с указателями и без них).</p>	<p>Имя массива A без индекса является указателем-константой (не изменяется на протяжении всей работы программы), т. е. адресом первого по счету элемента массива A[0].</p> <p>Цикл, в котором генерируется и выводится массив A, содержащий size элементов (случайные числа от 0 до size-1), можно реализовать тремя способами:</p> <ol style="list-style-type: none">for (i = 0; i < size; i++) { A[i] = rand()%99; cout << A[i] << endl; }for (i = 0; i < size; i++) { *(A + i) = rand()%99; cout << *(A + i) << endl;		

	<pre> } 3. int size, *pk; for (pk = A; pk < A + size; pk++) { *pk = rand() % 99; cout << *pk << endl; }</pre>				
<p>4. Изучить отличия ссылок от указателей. Выполнить программы, записанные в правой части.</p>	<p>Ссылку (ссылочный тип) можно рассматривать как альтернативное имя переменной или как указатель, который всегда разыменовывается.</p> <p>Например: int kol; int &pal = kol;</p> <p>Здесь & – оператор ссылки, означающий, что следующее за ним имя является именем переменной ссылочного типа, pal – альтернативное имя для kol.</p> <table><tr><th>Использование ссылки</th><th>Переход от ссылки к указателю</th></tr><tr><td><pre>#include <stdio.h> void main() { int V = 1; printf("V = %d\n", V); int &rV = V; rV = 5; printf("V = %d\n", V); }</pre></td><td><pre>#include <stdio.h> void main() { int V = 1; printf("V = %d\n", V); int &rV = V; rV = 5; int *pV = &rV; printf("V = %d\n", *pV); }</pre></td></tr></table>	Использование ссылки	Переход от ссылки к указателю	<pre>#include <stdio.h> void main() { int V = 1; printf("V = %d\n", V); int &rV = V; rV = 5; printf("V = %d\n", V); }</pre>	<pre>#include <stdio.h> void main() { int V = 1; printf("V = %d\n", V); int &rV = V; rV = 5; int *pV = &rV; printf("V = %d\n", *pV); }</pre>
Использование ссылки	Переход от ссылки к указателю				
<pre>#include <stdio.h> void main() { int V = 1; printf("V = %d\n", V); int &rV = V; rV = 5; printf("V = %d\n", V); }</pre>	<pre>#include <stdio.h> void main() { int V = 1; printf("V = %d\n", V); int &rV = V; rV = 5; int *pV = &rV; printf("V = %d\n", *pV); }</pre>				

	<p>Между ссылкой и указателем существуют два основных отличия: ссылка обязательно должна быть инициализирована в месте своего определения; всякое изменение ссылки преобразует не ее, а тот объект, на который она ссылается.</p>	
<p>5. Выполнить программу, приведенную в правой части, которая разработана с использованием указателей.</p> <p>Внести изменения с тем, чтобы программа стала содержать ошибки. Исследовать их с помощью отладки.</p>	<pre>#include <iostream> void main() { setlocale(LC_CTYPE, "Russian"); using namespace std; int i, k, sz = 5; float A[] = { 5, -4, 17.1, 9, 1 }; cout << "Введите номер элемента (от 0 до 4) " << endl; cin >> k; cout << endl; for (i = k; i < sz - 1; i++) *(A + i) = *(A + i + 1); sz--; for (i = 0; i < sz; i++) cout << *(A + i) << endl; }</pre>	<p>Пример. Пусть имеется массив A. Значения элементов массива инициализируются в программе. Удалить элемент с номером, который вводится с клавиатуры.</p>

6. В соответствии со своим вариантом написать программу для условия задачи из таблицы, представленной ниже, с использованием указателей. Результаты показать в Отладчике.

№ варианта	Условие задачи
1	Ввести целое число N. Выделить из этого числа цифры, кратные m, и записать их в одномерный массив.
2	Даны массивы A и B, состоящие из n элементов. Построить массив S, каждый элемент которого равен сумме соответствующих элементов массивов A и B.
3	Заданы два массива A(n) и B(n). Подсчитать в них количество элементов, меньших значения t, и первым на печать вывести массив, имеющий наибольшее их количество.
4	Даны две последовательности $x[1] \dots x[n]$ и $y[1] \dots y[k]$ целых чисел. Найти максимальную длину последовательности, являющейся подпоследовательностью обеих последовательностей, т.е. найти максимальную последовательность, которая содержит члены каждой последовательности.
5	Дан массив A из n элементов и B из m элементов. Содержится ли наибольший элемент массива A в массиве B?
6	Даны две последовательности $x[1] \dots x[n]$ и $y[1] \dots y[k]$ целых чисел. Выяснить, является ли вторая последовательность подпоследовательностью первой, т. е. можно ли из первой вычеркнуть некоторые члены так, чтобы осталась вторая.
7	Задан одномерный числовой массив A из n элементов и число k. Найти номера всех элементов массива, которые равны, больше и меньше k.
8	Дан массив x, содержащий k элементов, и y, содержащий n элементов. Найти их "пересечение" т.е. массив z, содержащий их общие элементы, причем кратность каждого элемента в массиве z равняется минимуму из его кратностей в массивах x и y.

№ варианта	Условие задачи
9	Даны N положительных целых чисел, которые не делятся ни на какие простые числа, кроме 2 и 3. Требуется удалить из массива минимально возможное количество чисел так, чтобы из любых двух оставшихся одно делилось на другое.
10	Заданы два массива по 10 целых чисел в каждом. Найти наименьшее среди чисел первого массива, которое не входит во второй массив (считая, что хотя бы одно такое число есть).
11	Вводится последовательность из n натуральных чисел. Необходимо определить наименьшее натуральное число, отсутствующее в последовательности.
12	Заданы два массива A и B, каждый из n элементов. Подсчитать количество таких k, для которых: $A[k] = B[k]$, $A[k] > B[k]$ и $A[k] < B[k]$.
13	Дан массив x, содержащий n элементов. Найти количество различных чисел среди элементов этого массива.
14	Даны два массива x и y. Найти количество одинаковых элементов в этих массивах, т. е. количество пар $x[i] = y[j]$ для некоторых i и j.
15	Дан массив A размера n, не содержащий нулевых элементов. Преобразовать массив A так, чтобы вначале шли положительные элементы, а затем отрицательные. Дополнительные массивы не использовать.
16	Дан массив x, содержащий k элементов, массив y, содержащий n элементов, и число q. Найти сумму вида $x[i] + y[j]$, наиболее близкую к числу q.

7. К номеру своего варианта прибавить 1 и написать программу для новых исходных данных (для варианта 16 перейти к варианту 1).

Тест "Указатели и ссылки"

[В начало практикума](#)

Лабораторная работа № 12. Обработка символьной информации

Для представления символьной информации можно использовать символы, символьные переменные и символьные литералы. Строка символов хранится в памяти как *массив*. Каждый элемент массива содержит символ, при этом последним символом является '\0'. Поэтому при объявлении массива надо указывать размерность на единицу больше, чем количество символов. Имя массива без индекса является **указателем-константой**.

Задание	Краткие теоретические сведения	
<p>1. Изучить способы преобразования символов, выполнив программы в правой части.</p> <p>Выполнить прокрутки программ и записать условия.</p> <p>Опробовать программы с различным текстом.</p>	<pre>#include <stdio.h> void main() { int num = 5; char symbol, new_symbol=' '; symbol = num + '0'; printf("%c\n", symbol); if (symbol >= '0' && symbol <= '9') num = symbol - '0'; printf(" %d\n", num); symbol = 'b'; if (symbol >= 'a' && symbol <= 'z') new_symbol = symbol - 'a' + 'A'; printf(" %c\n", new_symbol); }</pre> <p>Заменить в программе пятую, шестую и седьмую строчки на следующие:</p>	<pre>#include <stdio.h> void main() { char str[] = "Text"; int count = 0; char *pstr; pstr = &str[0]; if (str) while (*pstr++) ++count; printf("%d\n", count); }</pre> <p>В данной программе указатель может содержать нулевое значение, поэтому перед операцией разыменования его следует проверять.</p>

	<pre>char *pc; pc = &symbol; *pc = num + '0'; printf("%c\n", *pc);</pre> Объяснить результат.													
2. Выполнить программу, приведенную в правой части. Объяснить принцип использования стандартной библиотеки.	Пусть имеется адрес файла в сети, например: http://belstu.by/p1/p2/file1.htm Определить имя последней папки (каталога).	<table><tr><th>Функция</th><th>Краткое описание функции</th></tr><tr><td>strcmp</td><td>int strcmp(const char *str1, const char *str2); Сравнивает строки str1 и str2. Если str1 < str2, то результат <0, если str1 = str2, то результат = 0, если str1 > str2, то результат >0.</td></tr><tr><td>strcpy</td><td>char* strcpy(char*s1, const char *s2); Копирует байты из строки s1 в строку s2</td></tr><tr><td>strdup</td><td>char *strdup (const char *str); Выделяет память и переносит в нее копию строки str.</td></tr><tr><td>strlen</td><td>int strlen (const char *str); Вычисляет длину строки str</td></tr><tr><td>strncat</td><td>char *strncat(char *s1, const char *s2, int kol);</td></tr></table>	Функция	Краткое описание функции	strcmp	int strcmp(const char *str1, const char *str2); Сравнивает строки str1 и str2. Если str1 < str2, то результат <0, если str1 = str2, то результат = 0, если str1 > str2, то результат >0.	strcpy	char* strcpy(char*s1, const char *s2); Копирует байты из строки s1 в строку s2	strdup	char *strdup (const char *str); Выделяет память и переносит в нее копию строки str.	strlen	int strlen (const char *str); Вычисляет длину строки str	strncat	char *strncat(char *s1, const char *s2, int kol);
Функция	Краткое описание функции													
strcmp	int strcmp(const char *str1, const char *str2); Сравнивает строки str1 и str2. Если str1 < str2, то результат <0, если str1 = str2, то результат = 0, если str1 > str2, то результат >0.													
strcpy	char* strcpy(char*s1, const char *s2); Копирует байты из строки s1 в строку s2													
strdup	char *strdup (const char *str); Выделяет память и переносит в нее копию строки str.													
strlen	int strlen (const char *str); Вычисляет длину строки str													
strncat	char *strncat(char *s1, const char *s2, int kol);													

	<pre> printf(" '\/' "); else { char s[20]; strncpy_s(s, &str[index_first + 1], index_last - index_first - 1); s[index_last - index_first - 1] = 0; cout<<"katalog:"<< s<<"\n"; } } </pre>		Приписывает kol символов строки s1 к строке s2
		strncpy	char *strncpy(char *s1, const char *s2, int kol); Копирует kol символов строки s2 в строку s1
		strnset	char *strnset(char *str, int c, int kol); Заменяет первые kol символов строки s1 символом c
		atoi	int atoi(char *str); Преобразует строку в целое число
		atof	float atof(char *str); Преобразует строку в вещественное число

<p>3. В правой части записаны два варианта решения задачи. Выполнить программы и объяснить различия между ними.</p> <p>Внести изменения в программу с тем, чтобы проверялось не только количество скобок, но и правильность их</p>	<p><u>Пример</u> проверки соответствия количества открывающих и закрывающих круглых скобок в строке.</p>
--	--

расстановки (первой в тексте была открывающая скобка).

```
#include <stdio.h>
void main()
{
    char s[256];
    int i, count;

    puts("Enter string: ");
    gets_s(s);
    for (count = i = 0; s[i] != 0; i++)
    {
        if (s[i] == '(') count++;
        if (s[i] == ')') count--;
    }
    if (!count)
        puts("Ok\n");
    else
        puts("Not Ok\n");
}
```

```
#include <stdio.h>
void main()
{
    char s[256];
    int count;
    char *ps;

    puts("Enter string: ");
    gets_s(s);
    for (count = 0, ps = s; *ps != 0; ps++)
    {
        if (*ps == '(') count++;
        if (*ps == ')') count--;
    }
    if (count == 0)
        puts("Ok\n");
    else
        puts("Not Ok\n");
}
```

5. Выполнить задание из таблицы ниже двумя способами: используя индексы и используя указатели. При написании программ не использовать стандартные операции и функции для строк символов.

№ варианта	Условие задачи
1	Написать программу, реализующую вставку в строку n символов, начиная с позиции k .

№ варианта	Условие задачи
2	Написать программу, реализующую выделение подстроки $S1$ длиной k с позиции номер n из строки.
3	В строке есть два символа *. Получить все символы между первым и вторым символом *.
4	Написать программу, которая удаляет в строке все буквы b в тексте, написанном латинскими буквами.
5	Исключить из строки группы символов, расположенные между скобками вместе со скобками. Предполагается, что нет вложенных скобок.
6	В строке есть символы *. Преобразовать строку следующим образом: удалить все символы *, и повторить каждый символ, отличный от *.
7	Преобразовать строку: после каждой буквы a добавить символ !
8	Написать программу, которая осуществляет сравнение двух строк и выводит сообщение о том, какие символы совпадают.
9	Написать программу, реализующую вставку подстроки $S1$ длиной k в строку S с позиции номер n .
10	Написать программу, которая записывает строку в обратном порядке.
11	Вывести текст, составленный из последних букв всех слов.
12	Зашифровать введенную с клавиатуры строку, поменяв местами первый символ со вторым, третий с четвертым и т. д.
13	Отредактировать заданное предложение, удаляя из него все слова с чётными номерами.

№ варианта	Условие задачи
14	Найти самое длинное и самое короткое слово в заданном предложении.
15	Из предложения удалить все символы, совпадающие с символом, введенным с клавиатуры.
16	Из текста удалить те его части, которые заключены в кавычки (вместе с кавычками).

7. К номеру своего варианта прибавить 3 и написать программу для новых исходных данных (для вариантов с 14 по 16 перейти к вариантам с 1 по 3).
8. Дополнительные задания.
 1. В заданной последовательности слов найти все слова, имеющие заданное окончание.
 2. В заданном предложении указать слово, в котором доля гласных (А, Е, I, О, U — строчных или прописных) максимальна.
 3. В имеющемся словаре найти группы слов, записанных одними и теми же буквами и отличающиеся только их порядком.
 4. Из заданного предложения удалить те слова, которые уже встречались в предложении раньше.
 5. Отредактировать заданное предложение, удаляя из него все слова с нечетными номерами и переворачивая слова с четными номерами. Пример: HOW DO YOU DO -> OD OD.
 6. Преобразовать строку таким образом, чтобы в ее начале были записаны слова, содержащие только цифры, потом слова, содержащие только буквы, а затем слова, которые содержат и буквы и цифры.
 7. Даны два предложения. Найти самое короткое из слов первого предложения, которого нет во втором предложении.
 8. Найти самое длинное симметричное слово заданного предложения

В начало практикума

Лабораторная работа № 13. Многомерные массивы

Многомерными называются массивы, имеющие два и более индексов, которые заключаются в квадратные скобки. Имя массива без индекса является **указателем-константой** на начало массива.

Задание	Краткие теоретические сведения	
<p>1. Выполнить программу, записанную в правой части. Внести изменения в программу с тем, чтобы инициализировался другой массив, например: D[2][4]. Осуществить вывод массива в виде матрицы.</p>	<pre>#include <iostream> void main() { const int c_i = 3; const int c_j = 2; int a[c_i][c_j] = { { 1,2 }, { 3,4 }, { 5,6 } }; int i, j; for (i = 0; i < c_i; i++) for (j = 0; j < c_j; j++) std::cout << "\n a[" << i << ", " << j << "]" << "=" << a[i][j]; }</pre>	<p><u>Пример</u> программы, которая инициализирует массив и выводит его элементы на экран.</p>

<p>2. Выполнить программы, записанные в правой части для одной и той же задачи, условие которой надо определить. Одна из программ использует указатели. Изменить в программе форму обращения к элементам массива через указатели.</p>	<pre>#include <iostream> void main() { const int n = 3, m = 2; int i, j, h = 0; int arr[n][m] = { 0, 1, 2, 3, 4, 5 }; for (i = 0; i < n; i++) for (j = 0; j < m; j++) h += arr[i][j]; std::cout << h; }</pre>	<pre>#include <iostream> void main() { const int n = 3, m = 2; int i, j, h = 0; int C[n][m] = { 0,1,2,3,4,5 }; for (i = 0; i < n; i++) for (j = 0; j < m; j++) h += (*(C + i) + j); std::cout << h; }</pre>	<p>При объявлении двумерного массива, например, int arr[4][3]; в памяти выделяется участок для хранения значений массива. Для</p> <p>данного примера – четыре указателя на строки. Для доступа к элементам массива arr через указатели можно использовать следующую форму записи: *(arr + 1) + 2 или *(arr[1] + 2) или *(arr + 1)[2]</p>
<p>3. Изучить способы работы с двумерными массивами, выполнив программу в правой части и записав ее условие.</p> <p>Внести изменения в программу с тем, чтобы определялись минимальные элементы</p>	<pre>#include <iostream> void main() { setlocale(LC_CTYPE, "Russian"); using namespace std; const int n = 2, m = 4; int B[n][m]; int i, j, row = 0, colum = 0; cout << "Введите элементы массива" << endl; for (i = 0; i < n; i++) for (j = 0; j < m; j++) cin >> B[i][j]; }</pre>		

каждой строки с помощью указателей.	<pre> int min = B[0][0]; for (int i = 0; i < n; i++) for (int j = 0; j < m; j++) if (min > B[i][j]) { min = B[i][j]; colum = i; row = j; } cout << " Исходный массив:" << endl; for (int i = 0; i < n; i++) { cout << "\n"; for (int j = 0; j < m; j++) cout << "B[" << i << ", " << j << "] =" << B[i][j] << "\t"; } cout << endl; cout << "Минимальный элемент B[" << colum << ", " << row << "] =" << min << endl; } </pre>
-------------------------------------	---

4. Выполнить задание из таблицы ниже двумя способами: используя индексы и используя указатели.

№ варианта	Условие задачи
1	Найти наибольший элемент матрицы A(N, M), а также номера строки и столбца, на пересечении которых он находится.

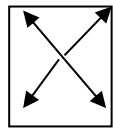
№ варианта	Условие задачи
2	В каждой строке заданной матрицы $A(N, M)$ вычислить сумму, количество и среднее арифметическое положительных элементов.
3	Для заданной целочисленной матрицы $A(N, M)$ определить, является ли сумма её элементов чётным числом.
4	Дана матрица $A(N, M)$. Найти количество элементов этой матрицы, больших среднего арифметического всех её элементов.
5	Дана целочисленная матрица $A(N, M)$. Вычислить сумму и произведение тех её элементов, которые при делении на два дают нечётное число.
6	В заданной матрице $A(N, M)$ поменять местами столбцы с номерами P и Q .
7	Дана матрица $A(N, M)$. Поменять местами её наибольший и наименьший элементы.
8	Даны две целочисленные матрицы $A(N, M)$ и $B(N, M)$. Подсчитать количество тех пар (a_{ij}, b_{ij}) , для которых: а) $a_{ij} < b_{ij}$; б) $a_{ij} = b_{ij}$; в) $a_{ij} > b_{ij}$.
9	Дана матрица $A(N, N)$. Переписать элементы её главной диагонали в одномерный массив $Y(N)$ и разделить их на максимальный элемент главной диагонали.
10	Дана матрица $B(n, m)$. Вычислить произведение чётных положительных элементов матрицы,
11	Найти наибольший элемент главной диагонали матрицы $A(N, N)$ и вывести на печать всю строку, в которой он находится.
12	Дана целочисленная матрица $A(N, M)$. Вычислить сумму и произведение нечётных отрицательных элементов матрицы, удовлетворяющих условию $ a_{ij} < i$.

№ варианта	Условие задачи
13	Найти наименьший элемент главной диагонали матрицы $C(N, N)$ и вывести на печать столбец, в котором он находится.
14	Дана матрица $A(N, N)$ и целое число m . Преобразовать матрицу по правилу: строку с номером M сделать столбцом с номером M , а столбец с номером M сделать строкой с номером M
15	В заданном массиве $A(N, N)$ вычислить две суммы элементов, расположенных выше и ниже главной диагонали.
16	Найти наиболее близкий к среднему арифметическому всех элементов матрицы $B(N, N)$, а также номера строки и столбца, на пересечении которых он находится.

5. К номеру своего варианта прибавить 2 и написать программу для новых исходных данных (для вариантов с 15 по 16 перейти к вариантам с 1 по 2). Опробовать работу программы.

6. Дополнительные задания.

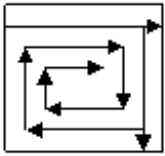
1. Дана квадратная матрица порядка $2n$, элементы которой формируются случайным образом и находятся в пределах от -10 до 10 . Получить новую матрицу, переставляя ее блоки размера $n \times n$ в соответствии со схемой.



2. Латинским квадратом порядка n называется квадратная таблица размером $n \times n$, каждая строка и каждый столбец которой содержит все числа от 1 до n . Для заданного n в матрице $L(n, n)$ построить латинский квадрат порядка n .

3. Для заданной матрицы размером 6 на 6 найти такие значения k , что k -я строка матрицы совпадает с k -м столбцом. Найти сумму элементов в тех строках, которые содержат хотя бы один отрицательный элемент.

4. Получить целочисленную квадратную матрицу порядка n , элементами которой являются числа $1, 2, \dots, n^2$, расположенные по спирали.

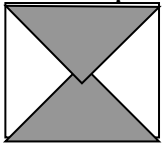


5. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции (2,2), следующий по величине – в позиции (3,3) и т. д., заполнив таким образом всю главную диагональ.

6. Получить квадратную матрицу порядка n

1	2	3	...	3	2	1
0	1	2	...	2	1	0
			...			
0	1	2	...	2	1	0
1	2	3	...	3	2	1

7. Дана действительная матрица $A(N, N)$. Найти сумму и \max значение среди элементов, расположенных в заштрихованной части матрицы



8. В трехмерном массиве $K(f, m, n)$, состоящем из нулей и единиц, хранится сеточное изображение некоторого трехмерного тела. Получить в двумерных массивах три проекции (тени) этого тела.

Тест "Многомерные массивы"

В начало практикума

Лабораторная работа № 14. Динамические массивы и функции пользователя

Динамическая память – это область памяти, которая выделяется во время работы программы. После выделения и использования памяти ее надо освобождать, что позволит эффективно расходовать память. Формирование динамических массивов можно организовать двумя способами: с использованием функций (язык C); с использованием операторов **new** и **delete** (язык C++).

Задание	Краткие теоретические сведения		
1. Изучить формирование динамического массива с использованием стандартных функций, выполнив программу на языке C, записанную справа. Написать пояснения к программе.	В таблице представлены функции работы с динамической памятью и программа, в которой вычисляется сумма: $s = n \cdot x_1 + (n - 1) \cdot x_2 + \dots + 2 \cdot x_{(n-1)} + x_n$. Значения x_1, x_2, \dots, x_n положительны, их количество не определено. Признаком конца ввода чисел является ввод отрицательного числа.		
	Функция	Прототип и краткое описание	Пример программы
	malloc	void *malloc(unsigned s) – возвращает указатель на начало области динамической памяти длиной в s байт, при неудачном завершении возвращает NULL	<pre>#include <stdio.h> #include <stdlib.h> void main() { int *N = (int*)malloc(0); int count = 0, i, sum=0 printf("Enter numbers:\n"); do { N = (int*)realloc(N, sizeof(int)*++count); scanf_s("%d", &N[count - 1]); } while (N[count - 1] > 0); count--;</pre>

	realloc	void *realloc(void *p, unsigned s) – изменяет размер блока ранее выделенной динамической памяти до размера s байт, p -адрес начала изменяемого блока, при неудачном завершении возвращает NULL	<pre>for (i = 0; i < count; i++) sum += (count - i) * N[i]; printf("Result is: %d\n", sum); free(N);</pre>
	free	void *free(void p) – освобождает ранее выделенный участок динамической памяти, p - адрес первого байта	
<p>2. Изучить способы выделения динамической памяти для <i>одномерного массива</i>, выполнив программу на языке C++, записанную в правой части.</p> <p>Опробовать работу программы с разными значениями вводимых символов.</p>	<p>Операция new позволяет выделить и сделать доступным свободный участок памяти, размеры которого соответствуют типу данных, определяемому именем типа. Оператор возвращает указатель на начало выделенного блока памяти.</p> <p>Операция delete освобождает участок памяти, ранее выделенный операцией new. Аргументом оператора delete выступает адрес первой ячейки блока, который необходимо освободить.</p> <p><u>Пример.</u> Пусть необходимо в строке символов, которые вводятся с клавиатуры, подсчитать количество повторений символа, который также вводится с клавиатуры. Количество символов в строке при каждом повторении программы разное.</p>		

```

#include <iostream>
using namespace std;
void main()
{
    setlocale(LC_CTYPE, "Russian");
    char *pStr; // указатель для адреса массива
    char c, *pc;
    int j, length, count = 0;

    cout << "Введите размер строки ";
    cin >> length;
    pStr = new char[length + 1];
           //выделение динам.памяти
    cout << "Введите строку ";
    cin >> pStr;
    cout << "Введите символ ";
    cin >> c;

    pc = &c;
    for (j = 0; j < length; j++)
        if (pStr[j] == *pc) count++;
    cout << "Повторений символа: " << count << endl;

    delete[] pStr; // освобождение динам. памяти
}

```

В этой программе при выделении динамической памяти количество элементов символьного массива `s` увеличено на 1, так как в конце массива должен присутствовать признак конца строки (нуль байт).

3. В программе, записанной справа, демонстрируется использование динамической памяти при работе с *двумерным массивом*.

Выполнить программу с различными размерами исходного массива.

```
#include <ctime>
#include <iostream>
using namespace std;
void main()
{
    int size, mult = 1, sum = 0;
    int **A;
    cout << "Enter size ";
    cin >> size;
    srand(time(0));
    A = new int*[size];
    for (int i = 0; i < size; i++)
    {
        A[i] = new int[size];
        for (int j = 0; j < size; j++)
            A[i][j] = 1 + rand() % 10;
    }
    for (int **ptr = A; ptr != A + size; ++ptr)
    {
        for (int* it = *ptr; it != *ptr + size; ++it)
        {
            cout << "\t" << *it;
            if (*it % 2 == 0)
            {
                mult *= *it;
                sum += *it;
            }
        }
        cout << endl;
    }
}
```

	<p><u>Пример.</u> Дана целочисленная квадратная матрица случайных чисел. Вычислить сумму и произведение ее четных элементов.</p>
--	--

	<pre>cout << "sum: " << sum << endl; cout << "mult: " << mult << endl; for (int k = 0; k < size; k++) delete[] A[k]; delete[] A; }</pre>		
<p>4. Изучить способы организации работы с функциями, выполнив программы, записанные в правой части.</p>	<p>В любой программе C/C++ должна быть функция с именем main (главная функция или основная программа) с которой начинается решение задачи. Могут также присутствовать <i>функции пользователя</i> (подпрограммы). Функция пользователя может быть записана как до основной программы, так и после. В последнем случае вызов функции следует предварять объявлением (прототипом) этой функции.</p> <p><u>Пример</u> программы, в которой происходит проверка пароля (например, пароля qwerty123) в двух вариантах: без использования функции и с использованием функции check_pass.</p> <table><tr><td><pre>#include <iostream> #include <string> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); string valid_pass = "qwerty123"; string user_pass; cout << "Введите пароль: "; getline(cin, user_pass); if (user_pass == valid_pass)</pre></td><td><pre>#include <iostream> #include <string> using namespace std; void check_pass(string password); //прототип функции void main() { setlocale(LC_CTYPE, "Russian"); string user_pass; cout << "Введите пароль: "; getline(cin, user_pass); check_pass(user_pass); }</pre></td></tr></table>	<pre>#include <iostream> #include <string> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); string valid_pass = "qwerty123"; string user_pass; cout << "Введите пароль: "; getline(cin, user_pass); if (user_pass == valid_pass)</pre>	<pre>#include <iostream> #include <string> using namespace std; void check_pass(string password); //прототип функции void main() { setlocale(LC_CTYPE, "Russian"); string user_pass; cout << "Введите пароль: "; getline(cin, user_pass); check_pass(user_pass); }</pre>
<pre>#include <iostream> #include <string> using namespace std; void main() { setlocale(LC_CTYPE, "Russian"); string valid_pass = "qwerty123"; string user_pass; cout << "Введите пароль: "; getline(cin, user_pass); if (user_pass == valid_pass)</pre>	<pre>#include <iostream> #include <string> using namespace std; void check_pass(string password); //прототип функции void main() { setlocale(LC_CTYPE, "Russian"); string user_pass; cout << "Введите пароль: "; getline(cin, user_pass); check_pass(user_pass); }</pre>		

	<pre> cout << "Доступ разрешен" << endl; else cout << "Неверный пароль!" << endl; } </pre>	<pre> void check_pass(string password) { string valid_pass = "qwerty123"; if (password == valid_pass) cout << "Доступ разрешен" << endl; else cout << "Неверный пароль" << endl; } </pre>
<p>5. В программе, записанной справа, разработан интерфейс с помощью оператора switch, который позволяет производить выбор между двумя функциями.</p> <p>Написать условие задачи.</p> <p>Изменить функцию matrix() с тем, чтобы выделялся и освобождался не одномерный динамический массив размерности n * m, а</p>	<pre> #include <iostream> using namespace std; void massiv(); void matrix(); int main(void) { setlocale(LC_CTYPE, "Russian"); int c; do { cout << endl; cout << "Введите" << endl; cout << "1-одномерный массив" << endl; cout << "2-работа с матрицей" << endl; cout << "3-выход" << endl; cin >> c; switch (c) </pre>	

двумерный динамический
массив (см. п.3).

```
{
    case 1: massiv(); break;
    case 2: matrix(); break;
    case 3: break;
}
} while (c != 3);
}

void massiv()
{
    int size=0, i, n;
    float *M, sum = 0, avar, mn;
    cout << "Введите размер массива ";
    cin >> size;

    M = new float[size];
    for (i = 0; i < size; i++)
    {
        cout << "Введите " << i + 1 << " элемент ";
        cin >> *(M + i);
        sum += *(M + i);
    }
    avar = sum / size;
    mn =abs( *M - avar);
    n = 0;
    for (i = 0; i < size; i++)
        if (abs(*(M + i) - avar ) < mn)
        {
            mn = abs(*(M + i) - avar);
        }
    }
```

```

        n = i;
    }
    cout << "Первое число, близкое к среднему арифм.=" << *(M + n);
    cout << endl;
    delete[] M;
}

void matrix()
{
    int *A, row, column, i, j, max_column, max_row, max;
    cout << "Введите число строк матрицы ";
    cin >> row;
    cout << "Введите число столбцов ";
    cin >> column;
    A = new int[row * column];
    for (i = 0; i < row; i++)
        for (j = 0; j < column; j++)
        {
            cout << "Введите " << i + 1 << " элемент " << j + 1 << "-й строки ";
            cin >> *(A + i * column + j);
        }
    max = A[0];
    max_row = 0;
    max_column = 0;
    for (i = 0; i < row; i++)
        for (j = 0; j < column; j++)
            if (*(A + i * column + j) > max)
            {
                max = *(A + i * column + j);
                max_row = i + 1;
            }
}

```

	<pre> max_colum = j + 1; } cout << "Максимальный элемент A[" << max_row << ", " << max_colum << "]= " << max; cout << endl; delete[] A; } </pre>
--	--

6. В соответствии со своим вариантом написать программы с использованием динамических массивов для условий задач из таблицы. Начальные размерности массивов ввести с клавиатуры. Объединить написанные программы, разработав интерфейс с помощью оператора **switch**, используя пример п. 5.

№ варианта	Условия задач
1	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить количество отрицательных элементов массива и сумму модулей элементов, расположенных после минимального по модулю элемента.</p> <p>2. Найти в матрице первый столбец, все элементы которого положительны. Знаки элементов предыдущего столбца изменить на противоположные.</p>
2	<p>1. Задан массив A из n элементов. Проверить, есть ли в нём отрицательные элементы. Если есть, найти наибольшее k, при котором $A[k] < 0$.</p> <p>2. Дана вещественная матрица размером 5x4. Переставляя ее строки и столбцы, добиться того, чтобы наибольший элемент (один из них) оказался в верхнем левом углу.</p>

№ варианта	Условия задач
3	<p>1. В одномерном массиве, состоящем из p вещественных элементов, вычислить произведение отрицательных элементов массива и сумму положительных элементов массива, расположенных до максимального элемента.</p> <p>2. Дана матрица размером 4×4. Найти сумму наименьших элементов ее нечетных строк и наибольших элементов ее четных строк.</p>
4	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить номер минимального элемента массива и сумму элементов массива, расположенных между первым и вторым отрицательными элементами.</p> <p>2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, не содержащих ни одного нулевого элемента.</p>
5	<p>1. В одномерном массиве, состоящем из p вещественных элементов, вычислить количество элементов массива, равных 0, и сумму элементов массива, расположенных после минимального элемента.</p> <p>2. Дана целочисленная прямоугольная матрица. Определить количество столбцов, содержащих хотя бы один нулевой элемент и номер строки, в которой находится самая длинная серия одинаковых элементов.</p>
6	<p>1. В одномерном массиве, состоящем из p вещественных элементов, вычислить номер минимального по модулю элемента массива и сумму модулей элементов массива, расположенных после первого отрицательного элемента.</p> <p>2. Дана целочисленная квадратная матрица. Определить произведение элементов в тех строках, которые не содержат отрицательных элементов и максимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.</p>

№ варианта	Условия задач
7	<p>1. Задан массив A из n элементов. Найти количество элементов этого массива, больших среднего арифметического всех его элементов.</p> <p>2. Дана целочисленная прямоугольная матрица. Определить количество строк, не содержащих ни одного нулевого элемента и максимальное из чисел, встречающихся в заданной матрице более одного раза.</p>
8	<p>1. В одномерном массиве, состоящем из k целых элементов, вычислить количество положительных элементов массива и сумму элементов массива, расположенных после последнего элемента, равного нулю.</p> <p>2. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.</p>
9	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить произведение положительных элементов массива и сумму элементов массива, расположенных до минимального элемента.</p> <p>2. Найти в матрице первую строку, все элементы которой положительны, и сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.</p>
10	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму положительных элементов массива и произведение элементов массива, расположенных между максимальным по модулю и минимальным по модулю элементами.</p> <p>2. Найти в матрице первую строку, все элементы которой отрицательны. Увеличить все элементы матрицы на значение первого элемента найденной строки.</p>
11	<p>1. Задан массив A из n элементов. Проверить, есть ли в нём элементы, равные нулю. Если есть, найти наименьшее k, при котором $A[k]=0$.</p>

№ варианта	Условия задач
	2. Для заданной матрицы размером 8 на 8 найти такие k , что k -я строка матрицы совпадает с k -м столбцом.
12	<p>1. Задан массив A из n элементов. Подсчитать, сколько раз встречается в нем максимальное число.</p> <p>2. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.</p>
13	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить сумму элементов массива с нечетными номерами и сумму элементов массива, расположенных между первым и последним отрицательными элементами.</p> <p>2. Найти в матрице первую строку, все элементы которой равны нулю. Все элементы столбца с таким же номером уменьшить вдвое.</p>
14	<p>1. В массиве, состоящем из вещественных элементов, вычислить количество элементов массива, больших C, и произведение элементов массива, расположенных после максимального по модулю элемента.</p> <p>2. Проверить, есть ли в матрице хотя бы одна строка, содержащая элемент, равный нулю, и найти ее номер. Уменьшить все элементы матрицы на значение первого элемента найденной строки.</p>
15	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить количество элементов массива, меньших C, и сумму целых частей элементов массива, расположенных после последнего отрицательного элемента.</p> <p>2. Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые.</p>

№ варианта	Условия задач
16	<p>1. В одномерном массиве, состоящем из n вещественных элементов, вычислить минимальный элемент массива и сумму элементов, расположенных между первым и последним положительными элементами.</p> <p>2. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое.</p>

Тест "Динамические массивы"

[В начало практикума](#)

Лабораторная работа № 15. Указатели как параметры и результаты функции

Формальные параметры функции представляют собой локальные переменные, которым в момент вызова присваиваются значения фактических параметров. Формальные параметры внутри функции могут как угодно изменяться – это не затрагивает соответствующих фактических параметров в основной программе.

Если фактический параметр должен быть изменен в подпрограмме-функции, то формальный параметр надо определить как указатель. Тогда фактический параметр должен быть передан с использованием операции &.

Функция в качестве результата может возвращать указатель. Указатель-результат функции может ссылаться не только на отдельную переменную, но и на массив.

Задание	Краткие теоретические сведения
1. Выполнить программы, записанные в правой части и изучить использование указателя как формального параметра функции.	<p>Если внутри функции изменить фактический параметр, переданный из основной программы в функцию, то это изменение в основной программе не будет отражено.</p> <p><u>Пример.</u> Вычислить значения $s = 1! / x + 2! / x + \dots + n! / x$ для $x = -2(0.5)^0$. В процессе вычислений заменить $x = 0$ на $x = 1$.</p> <p>В таблице ниже приведены два варианта программы для одной и той же задачи.</p> <p>При выводе результатов в программе слева видно, что хотя функция считает значение s при $x = 1$, в основной программе по-прежнему $x = 0$.</p> <p>Чтобы эту ошибку исправить, надо формальный параметр x определить как указатель. Тогда фактический параметр x должен быть явно передан в виде указателя с использованием операции & в программе справа.</p>

```

#include <iostream>
using namespace std;

float ff(int size, float x)
{
    int F = 1;
    float sum = 0;
    for (int i = 1; i <= size; i++)
    {
        if (x == 0.0)
        {
            x = 1;
            cout << "x =" << x
                << endl;
        }
        F = F * i;
        sum = sum + F / x;
    }
    return sum;
}

int main()
{
    int size; float x = -2.0;
    cout << "size =";
    cin >> size;
    do
    {
        cout << "x =" << x << " s="
            << ff(size, x) << endl;

```

```

#include <iostream>
using namespace std;
float ff(int size, float *x)
{
    int F = 1;
    float sum = 0;
    for (int i = 1; i <= size; i++)
    {
        if (*x == 0.0)
        {
            *x = 1;
            cout << "x =" << *x
                << endl;
        }
        F = F * i;
        sum = sum + F / (*x);
    }
    return sum;
}

int main()
{
    int size; float x = -2.0;
    cout << "size =";
    cin >> size;
    do
    {
        cout << "x =" << x << " s="
            << ff(size, &x) << endl;
        x = x + 0.5;

```

	<pre> x = x + 0.5; } while (x <= 0.0); }</pre>	<pre>} while (x <= 0.0); }</pre>
<p>2. Изучить использование указателя как формального параметра функции при работе с массивом символьных данных, опробовав работу программы в правой части.</p>	<pre>#include <stdio.h> #include <ctype.h> void letters(char *str); int main(void) { char s[80]; puts("Input char"); gets_s(s); letters(s); return 0; } void letters(char *str) { for (int t = 0; str[t]; ++t) { str[t] = toupper(str[t]); printf("%c", str[t]); } }</pre>	<p><u>Пример</u> программы, осуществляющей вывод букв в верхнем регистре.</p> <p>Здесь функция toupper выполняет преобразование строчных букв в прописные. Для ее использования нужна директива #include <ctype.h></p>
<p>3. Выполнить программы, записанные в</p>	<p>При передаче массива в функцию его имя преобразуется в указатель, и копия указателя на начало массива передается в функцию по значению.</p>	

<p>правой части и изучить один из способов передачи массива в функцию, а также использование указателя как результата выполнения функции.</p>	<p><u>Пример.</u> Пусть имеется массив В. С использованием функции определить минимальный элемент массива в подпрограмме и в основной программе увеличить его значение на 1.</p> <p>Ниже даны два варианта программы. Программа в левой части выполниться не сможет. Использование указателя как <i>результата выполнения</i> функции в правой части позволяет решить задачу.</p> <div> <div> <pre> #include <iostream> int pfmin(int p, int n) { int pmin; for (pmin = p; n > 0; p++, n--) if (p < pmin) pmin = p; return pmin; } void main() { int B[5] = { 4, 8, 2, 6, 4 }; (pfmin(B, 5))++; for (int i = 0; i < 5; i++) printf("%d ", B[i]); } </pre> </div> <div> <pre> #include <iostream> int *pfmin(int *p, int n) { int *pmin; for (pmin = p; n > 0; p++, n--) if (*p < *pmin) pmin = p; return pmin; } void main() { int B[5] = { 4, 8, 2, 6, 4 }; (*pfmin(B, 5))++; for (int i = 0; i < 5; i++) printf("%d ", B[i]); } </pre> </div> </div>
<p>4. Программа вычисления интеграла справа демонстрирует применение указателя на функцию, который используется для передачи</p>	<p>Программа вычисления интеграла $\int_0^{10} (2x + 5)dx$ методом правых прямоугольников</p> <pre> #include <iostream> float integ(float(*) (float), float, float, float); float f(float); //прототипы </pre>

имени функции **f** в качестве
формального параметра
функции **integ**.

```
int main()
{
    float z;
    z = integ(f, (float)0.0, (float)10.0, (float)0.01);
    std::cout << "Integral=" << z;
}

float integ(float(*f) (float), float a, float b, float h)
{
    float x, s = 0.0;
    x = a + h;
    while (x <= b)
    {
        s += h*f(x);
        x = x + h;
    }
    return s / 2;
}

float f(float x)
{
    return (2 * x + 5);
}
```

5. В соответствии со своим вариантом написать программу решения уравнения из лабораторной работы № 8. Операторы метода вычисления корня оформить в виде функции пользователя, уравнение также записать в виде функции пользователя. В основной программе предусмотреть ввод исходных данных, обращение к функциям и вывод результатов. Использовать *указатель на функцию*.

6. К номеру своего варианта прибавить 1 и написать программу с использованием динамических массивов для условий из лабораторной работы № 12. Внести изменения в программу с тем, чтобы продемонстрировать использование *указателей как формальных параметров* функции и как *результатов выполнения* функции.

Тест "Функции"

[В начало практикума](#)

Лабораторная работа № 16. Массивы и ссылки при работе с функциями

Задание	Краткие теоретические сведения	
1. Изучить использование <i>ссылок как формальных параметров</i> , выполнив программы, записанные в правой части	<p>При передаче фактического параметра по <i>ссылке</i> передаётся адрес объекта и, соответственно, работа внутри функции происходит не с копией, а с оригиналом объекта. Чтобы параметр передавался по ссылке, достаточно в прототипе функции поставить знак & после типа параметра.</p> <p>При передаче параметра функции в виде ссылки нет необходимости в разыменовании указателей.</p>	
	<u>Пример</u> программы обмена значениями переменных с использованием <i>указателей</i>	<u>Пример</u> программы обмена значениями переменных с использованием <i>ссылок</i>
	<pre>#include <iostream> void swap(int *a, int *b) { int temp = *a; *a = *b; *b = temp; } void main() { int a = 10, b = 100; swap(&a, &b); std::cout << a << ' ' << b; //a=100, b=10 }</pre>	<pre>#include <iostream> void swap(int& a, int& b) { int temp = a; a = b; b = temp; } void main() { int a = 10, b = 100; swap(a, b); std::cout << a << ' ' << b; //a=100, b=10 }</pre>

<p>2. Изучить использование <i>ссылки</i> как <i>результата выполнения функции</i>, выполнив программу, записанную в правой части.</p> <p>Опробовать работу программы с разными значениями массива A.</p>	<p><u>Пример</u>. Массив A содержит набор значений. Необходимо в подпрограмме определить минимальный элемент и изменить его на другое значение в основной программе.</p>

```

#include <iostream>

double &dmin(double A[], int size)
{
    int i, j = 0;
    for (i = 1; i < size; i++)
        if (A[j] > A[i])
            j = i;
    return A[j]; //возвращается ссылка
}

void main()
{
    double s;
    const int size = 5;
    double A[] = { 5, 4.1, 3, 0.2, 11 };
    s = dmin(A, size);
    std::cout << s<< std::endl;
    for (int i = 0; i < size; i++)
        std::cout << " " << A[i];
    std::cout << std::endl;
    dmin(A, size) = 1.0; // изменение минимума на 1.0
    for (int i = 0; i < size; i++)
        std::cout << " " << A[i]:

```

3. Ознакомиться с использованием функций с результатом логического

Пример. Имеется массив целых чисел **A**. Определить, содержит ли он число, которое вводится с клавиатуры.

```
#include <iostream>
```

типа, опробовав работу программы в правой части.

```
using namespace std;

bool is_elem(int *pA, int n, int iV)
{
    bool bf = false;
    for (int i = 0; i < n; i++)
        if (pA[i] == iV)
        { bf = true;
          break;
        }
    return bf;
}

void main()
{
    setlocale(LC_CTYPE, "Rus");
    bool t;  int k;
    int A[] = { 5, 4, 3, 2, 11 };
    cout << "Введите число ";
    cin >> k;
    t = is_elem(A, sizeof(A)/sizeof(int), k);
    if (t == true)
        cout << "Число есть в массиве ";
    else
        cout << "Числа нет в массиве ";
}
```

Результат, возвращаемый функцией **bool is_elem ()**, это логическая переменная, которая может принимать два значения – **true** (если искомый элемент есть в массиве) и **false** (если элемент отсутствует).

<p>4. Выполнить программу, содержащую функцию с <i>переменным числом параметров</i>. Записать условие задачи.</p>	<pre>#include <iostream> int sum(int n, ...) { int *p = &n; int sum = 0; for (int i = 1; i <= n; i++) sum += *(++p); return sum; } void main() { std::cout << sum(6, 4, 5, 1, 2, 3, 0)<<std::endl; std::cout << sum(2, 34, 4445); }</pre>	<p>При вызове функции с <i>переменным числом параметров</i> задается любое требуемое число аргументов. В объявлении и определении такой функции переменное число аргументов задается многоточием в конце списка формальных параметров или списка типов аргументов. При этом должен быть указан как минимум один обычный параметр.</p>
---	--	---

5. В соответствии со своим вариантом написать программы по условиям задач из таблицы ниже. Программа первого задания должна использовать динамические массивы, функции и ссылки.

Программа второго задания должна содержать функцию пользователя с переменным числом параметров и не менее трех обращений к ней с различным количеством параметров.

№ вариант а	Условие задачи
1	<p>1. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.</p> <p>2. Написать функцию fmin с переменным числом параметров, которая находит минимальное из чисел типа int.</p>
2	<p>1. Дана целочисленная прямоугольная матрица. Определить количество строк, содержащих хотя бы один нулевой элемент.</p> <p>2. Написать функцию fsum с переменным числом параметров, которая находит сумму чисел типа int по формуле: $S=a1*a2+a2*a3+a3*a4+ \dots$</p>
3	<p>1. Найти в матрице первую строку, все элементы которой положительны, и сумму этих элементов. Уменьшить все элементы матрицы на эту сумму.</p> <p>2. Написать функцию fmax с переменным числом параметров, которая находит минимальное из чисел типа int или из чисел типа double, тип параметров определяется с помощью первого параметра функции.</p>
4	<p>1. Проверить, есть ли в матрице хотя бы одна строка, содержащая положительный элемент, и найти ее номер. Знаки элементов предыдущей строки изменить на противоположные.</p> <p>2. Написать функцию days с переменным числом параметров, которая находит количество дней, прошедших между двумя датами (параметрами функции являются даты в формате «дд.мм.гг»).</p>
5	<p>1. Найти в матрице первую строку, все элементы которой равны нулю. Все элементы столбца с таким же номером уменьшить вдвое.</p>

№ вариант а	Условие задачи
	2. Написать функцию kvadr с переменным числом параметров, которая определяет количество чисел, являющихся точными квадратами (2, 4, 9, 16,...) типа int .
6	<p>1. Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые.</p> <p>2. Написать функцию as с переменным числом параметров, которая находит сумму чисел типа int по формуле: $S=a1*a2-a2*a3+a3*a4-....$</p>
7	<p>1. Найти в матрице первый столбец, все элементы которого отрицательны, и среднее арифметическое этих элементов. Вычесть полученное значение из всех элементов матрицы.</p> <p>2. Написать функцию mn с переменным числом параметров, которая находит минимальное из чисел типа int или из чисел типа double, тип параметров определяется с помощью первого параметра функции.</p>
8	<p>1. Проверить, есть ли в матрице хотя бы один столбец, содержащий отрицательный элемент, и найти его номер. Уменьшить элементы найденного столбца вдвое.</p> <p>2. Написать функцию prost с переменным числом параметров, которая находит все простые числа из нескольких интервалов. Интервалы задаются границами a и b. С помощью этой функции проверить несколько интервалов.</p>
9	1. Дана матрица размером 4 x4. Найти сумму наименьших элементов ее нечетных строк и наибольших элементов ее четных строк.

№ вариант а	Условие задачи
	2. Во введенном тексте подсчитать количество символов в слове максимальной длины (слова разделяются пробелами) с помощью функции с переменным числом параметров.
10	<p>1. Дана целочисленная прямоугольная матрица. Определить максимальное из чисел, встречающихся в заданной матрице более одного раза.</p> <p>2. Написать функцию, которая находит в строке самое первое (по алфавиту) слово. С ее помощью реализовать размещение слов в выходной строке в алфавитном порядке.</p>
11	<p>1. Дана целочисленная квадратная матрица. Определить сумму элементов в тех строках, которые не содержат отрицательных элементов.</p> <p>2. Написать функцию с переменным числом параметров, которая находит минимальное значение матрицы. С ее помощью найти минимальные значения в n матрицах.</p>
12	<p>1. Проверить, есть ли в матрице хотя бы одна строка, содержащая отрицательный элемент, и найти ее номер. Все элементы столбца с таким же номером уменьшить вдвое.</p> <p>2. Написать функцию, проверяющую есть ли отрицательные элементы в заданном одномерном массиве размерностью n. Удалить из массива все отрицательные элементы, удаленный элемент заполняется нулем и переносится в конец массива.</p>
13	<p>1. Дана целочисленная квадратная матрица. Определить минимум среди сумм элементов диагоналей, параллельных главной диагонали матрицы.</p> <p>2. Написать функцию для вычисления суммы элементов квадратной матрицы, которые расположены ниже главной диагонали. С ее помощью найти максимальное значение такой суммы в n матрицах.</p>

№ вариант а	Условие задачи
14	<p>1. Найти в матрице первый столбец, все элементы которого положительны. Знаки элементов предыдущего столбца изменить на противоположные.</p> <p>2. Написать и протестировать функцию compr, которая «сжимает» строку, удаляя из нее все пробелы. С ее помощью сжать различные строки.</p>
15	<p>1. Проверить, все ли столбцы матрицы содержат хотя бы один положительный элемент. Если нет, то в первом столбце, не удовлетворяющем условию, заменить отрицательные элементы их модулями.</p> <p>2. Написать функцию с переменным числом параметров для перевода чисел из десятичной системы счисления в двоичную. С помощью этой функции перевести различные числа из десятичной системы счисления в двоичную.</p>
16	<p>1. Проверить, все ли строки матрицы содержат хотя бы один отрицательный элемент. Если да, то изменить знаки всех элементов матрицы на обратные.</p> <p>2. Выяснить, есть ли во введенном тексте слова, начинающиеся с буквы A, и сколько таких слов. С помощью этой функции проверить несколько строк.</p>

5 вопрос два правильных, не корректно

2 вопрос – не корректен надо прототип

3 вопрос - не корректно

1- Не корректно

Тест "Массивы и ссылки при работе с функциями"

В начало практикума