

Государственное бюджетное профессиональное образовательное учреждение  
Астраханской области «Астраханский колледж вычислительной техники»

**КОМПЛЕКС**  
МЕТОДИЧЕСКИХ РЕКОМЕНДАЦИЙ  
по проведению лабораторных работ

по дисциплине Основам алгоритмизации и программированию

по специальности 10.02.03

Информационная безопасность

автоматизированных систем

*АКВТ.09.02.03.ЛР46.0000*

Листов: 81

## 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.

Для **оформления отчетов по лабораторным работам** следует использовать приложение **Word**. Каждая работа должна содержать название работы, условия задач в соответствии со своим вариантом, алгоритмы, тексты разработанных программ, результаты. Все работы сохраняются в **одном** файле.

## **ПРАВИЛА ПРОВЕДЕНИЯ ЛАБОРАТОРНЫХ РАБОТ**

Перед выполнением лабораторных работ по дисциплине студенты должны:

— строго выполнить весь объем домашней подготовки, т.е. ознакомиться с теоретическим материалом, указанный в описаниях соответствующих лабораторных работ;

— знать, что выполнению каждой работы предшествует проверка готовности студента, которая производится преподавателем;

— знать, что после выполнения работы группа студентов, которая назначается преподавателем на весь период работы, должна представить отчёт о проделанной работе с обсуждением полученных результатов и выводов;

ознакомиться с требованиями и процедурами выставления окончательной оценки (зачёта) по работе и порядок выполнения пропущенных работ по уважительным и неуважительным причинам.

## ОГЛАВЛЕНИЕ

- Лабораторная работа № 13. [Основные элементы языка C++. MS Visual Studio 2010. Ввод и вывод информации](#)
- Лабораторная работа № 14. [Разветвляющиеся программы](#)
- Лабораторная работа № 15. [Циклические программы. Отладка программ](#)
- Лабораторная работа № 16. [Решение инженерных задач на основе циклических программ. Указатели и ссылки](#)
- Лабораторная работа № 17. [Одномерные массивы. Многомерные массивы](#)
- Лабораторная работа № 18. [Обработка символьной информации](#)
- Лабораторная работа № 19. [Указатели как параметры и результаты функции](#)
- Лабораторная работа № 20. [Массивы и ссылки при работе с функциями](#)

## Лабораторная работа № 13. Основные элементы языка C++. MS Visual Studio 2010. Ввод и вывод информации

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, которые 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**);
- с 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 применяются *знаки операций*: +, −, \*, /.

Математические функции вычисляются с помощью стандартных встроенных функций.

Математическая запись	$\sin x$	$\cos x$	$e^x$	$ x $	$\operatorname{tg} x$	$\ln x$	$\sqrt{x}$	$x^y$
Запись на C++	<code>sin(x)</code>	<code>cos(x)</code>	<code>exp(x)</code>	<code>abs(x)</code>	<code>tan(x)</code>	<code>log(x)</code>	<code>sqrt(x)</code>	<code>pow(x, y)</code>

Программа начинается 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 2010** (а также в приложениях **Microsoft Visual Studio 2012, 2013** и др.), для запуска которого надо выполнить: **Пуск / Программы / Microsoft Visual Studio 2010 / Visual Studio 2010**.

Задание	Краткие теоретические сведения
<p>1. Создать новый проект с именем, которое состоит из фамилии студента и номера лабораторной работы.</p>	<p>Для создания нового проекта надо в окне <b>Начальная страница</b> выполнить <b>Создать проект</b>. Среда <b>Visual Studio</b> отобразит окно <b>Создать проект</b>. В окне следует развернуть узел <b>Visual C++</b>, выбрать <b>Win32</b> и на центральной панели выбрать <b>Консольное приложение Win32</b>. Затем в поле <b>Имя</b> надо ввести имя проекта, в поле <b>Расположение</b> указать путь размещения проекта, или выбрать его с помощью кнопки <b>Обзор</b>.</p> <p>После нажатия <b>ОК</b> откроется окно <b>Мастер приложений Win32</b>. Надо нажать кнопку <b>Далее</b>, в появившемся окне поставить галочку в поле <b>Пустой проект</b>. После нажатия кноп-</p>

	ки <b>Готово</b> на экране появится окно <b>Обозреватель решений</b> .	
2. Создать новый файл с программой под именем <b>main.cpp</b>	В контекстном меню окна <b>Обозреватель решений</b> выполнить <b>Добавить / Создать элемент</b> . В левой части появившегося окна выбрать <b>Код</b> , в центре выбрать <b>Файл C++</b> . В поле <b>Имя</b> задать имя нового файла, указать расширение <b>".cpp"</b> , нажать <b>Добавить</b> .	
3. Ввести программу, текст которой приведен справа, проанализировать ее.  Выполнить программу, ознакомиться с результатом.	<pre> /* Пример 1 */ #include &lt;iostream&gt; void main() {     float x = 3, y;     y = x * x + sin(x);     std::cout &lt;&lt; y; } </pre>	<p>Программа рассчитывает значение <math>y = x^2 + \sin(x)</math> для <math>x = 3</math>.</p> <p>Она начинается с комментария, который заключен внутри комбинации символов <code>/*</code> и <code>*/</code>.</p> <p>Строка <code>#include &lt;iostream&gt;</code> будет заменена содержимым файла <code>&lt;iostream.h&gt;</code>, который находится в стандартной библиотеке языка и отвечает за ввод-вывод данных на экран. В строке <code>float x=3, y;</code> определяется тип переменных <code>x</code> и <code>y</code> в виде действительных чисел. В переменную <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 <b>Сервис / Параметры / Текстовый редактор / C++</b>. Отметить пункт <b>Показывать номера строк</b>.</p> <p>Для компиляции программы следует выполнить <b>Построение / Компилировать</b> (или нажать клавиши <b>Ctrl – F7</b>). Запуск программы на выполнение: <b>Отладка / Запуск без отлад-</b></p>

	<b>ки (или Ctrl – F5).</b> Повторный вызов консольного окна на экран – также <b>Ctrl – F5</b> .	
4. В предыдущем проекте заменить текст программы на программу, записанную в правой части данного пункта, выполнить ее.	Исходные данные: $k = 4$ ; $a = 4.1$ ; $x = 5 \cdot 10^{-5}$ . Вычислить: $t = 2 \operatorname{tg}(k) / a - \ln( 3 + x ) + \exp(x)$ ; $u = \sqrt{t + 1} \cdot (\sin(x) - \cos(t))$ ;	<pre> <b>#include &lt;iostream&gt;</b> <b>void main()</b> {     <b>double t, u, k = 4, a = 4.1, x = 5e-5;</b>     <b>t = 2 * tan(k) / a * log(abs(3 + x)) + exp(x);</b>     <b>u = sqrt(t + 1) * (sin(x) * cos(t));</b>     <b>std::cout &lt;&lt; "t=" &lt;&lt; t;</b>     <b>std::cout &lt;&lt; "u=" &lt;&lt; u;</b> } </pre>

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

Задание	Краткие теоретические сведения
1. Выполнить программу, записанную справа, которая использует <i>потоковый</i> вывод данных. Проанализировать ее текст. Добавить в программу ввод и вывод переменных различных типов.	<b>setlocale (LC_CTYPE, "Russian");</b> используется для вывода кириллицы. <b>cout</b> – стандартный поток вывода (на экран дисплея); <b>cin</b> – стандартный поток ввода (с клавиатуры); <b>cerr, clog</b> – стандартный поток сообщений об ошибках; <b>endl</b> – перевод на новую строку. Если в программе записано <b>using namespace std;</b> (используя стандартное пространство имен), то перед каждым оператором вывода можно не писать <b>std::</b>



```
#include <iostream>
using namespace std;
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
```

// – начало комментария, конец которого определяется концом строки.

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**.

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;
}
```

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)** – 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;
```

<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>: <code>\n</code> – перемещает курсор в начальную позицию следующей строки; <code>\t</code> – перемещает курсор в следующую позицию табуляции экрана; <code>\r</code> – выполняет «возврат каретки», перемещая курсор к началу той же строки без перехода на следующую; <code>\b</code> – передвигает курсор только на одну позицию влево.</p> <p>Пример программы, выводящей слово <b>"Privet"</b>.</p> <table border="1" data-bbox="595 408 2074 753"> <tr> <td data-bbox="595 408 1205 753"> <pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main() {     printf("\n\t Privet\n");     printf("\n... Press key");     _getch(); }</pre> </td> <td data-bbox="1205 408 2074 753"> <p>Заголовочный файл с именем <b>stdio.h</b> может использоваться при форматированном вводе-выводе данных.</p> <p>Заголовочный файл <b>conio.h</b> поддерживает функцию <code>_getch()</code>, которая ожидает нажатия клавиши на клавиатуре.</p> <p>Функция <code>printf()</code> печатает информацию.</p> </td> </tr> </table>	<pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main() {     printf("\n\t Privet\n");     printf("\n... Press key");     _getch(); }</pre>	<p>Заголовочный файл с именем <b>stdio.h</b> может использоваться при форматированном вводе-выводе данных.</p> <p>Заголовочный файл <b>conio.h</b> поддерживает функцию <code>_getch()</code>, которая ожидает нажатия клавиши на клавиатуре.</p> <p>Функция <code>printf()</code> печатает информацию.</p>
<pre>#include &lt;stdio.h&gt; #include &lt;conio.h&gt; void main() {     printf("\n\t Privet\n");     printf("\n... Press key");     _getch(); }</pre>	<p>Заголовочный файл с именем <b>stdio.h</b> может использоваться при форматированном вводе-выводе данных.</p> <p>Заголовочный файл <b>conio.h</b> поддерживает функцию <code>_getch()</code>, которая ожидает нажатия клавиши на клавиатуре.</p> <p>Функция <code>printf()</code> печатает информацию.</p>		
<p>4. Внести изменения в программы лабораторной работы № 2 с тем, чтобы осуществлялся форматированный ввод и вывод данных.</p>	<p>Каждому аргументу функции <code>printf()</code> соответствует своя спецификация преобразования, которая начинается с символа процента (%), после которого стоит буква, указывающая тип данных: <code>%d</code> – десятичное целое число; <code>%f</code> – вещественное число типа <b>float</b> или <b>double</b>; <code>%c</code> – символ; <code>%s</code> – строка; <code>%p</code> – указатель; <code>%u</code> – беззнаковое целое число; <code>%o</code> – целые числа в восьмеричной системе счисления; <code>%x</code> – целые числа в шестнадцатеричной системе счисления; <code>%e</code> – вещественное число в экспоненциальной форме.</p> <p>В модификаторах формата после символа % можно указывать строку цифр, задающую минимальную ширину поля вывода, например: <code>%5d</code> (для целых), <code>%4.2f</code> (для вещественных – две цифры после запятой для поля шириной 4 символа). Если указанной ширины не хватает, происходит автоматическое расширение.</p>		

	<p>Функция <b>scanf()</b> позволяет вводить данные любых типов. В качестве аргументов используются указатели объектов <b>&amp;</b>. Например: <b>scanf(“ %d%f ”, &amp;x, &amp;y);</b></p> <p>Если нужно ввести значение строковой переменной, то использовать символ <b>&amp;</b> не нужно. Строка – массив символов, а имя массива эквивалентно адресу его первого элемента:</p> <pre><b>char name[20];</b> <b>scanf(“%s”, name);</b></pre> <p>В новых версиях <b>Visual Studio</b> компилятор C++ считает функцию <b>scanf</b> небезопасной и выдает ошибку во время компиляции. Можно использовать безопасную версию <b>scanf_s</b> либо добавить строку для отключения этой ошибки:</p> <pre><b>#define _CRT_SECURE_NO_WARNINGS</b></pre> <p>То же самое относится и к некоторым другим функциям, описанным ниже.</p>
<p>5. Написать программу, реализующую диалог, используя пример, записанный в правой части.</p>	<pre><b>#include &lt;windows.h&gt;</b> <b>void main()</b> {     <b>SetConsoleOutputCP(1251);</b>     <b>SetConsoleCP(1251);</b>     <b>char name[60];</b>     <b>puts("Как вас зовут? ");</b>     <b>gets_s(name);</b>     <b>printf("Привет, %s\n", name);</b> }</pre> <p>Функция <b>getchar()</b> вводит с клавиатуры единичный символ: <b>int letter; letter = getchar();</b></p> <p>Для корректного ввода и вывода русского текста в примере использованы функции: <b>SetConsoleOutputCP()</b> и <b>SetConsoleCP()</b>. В качестве параметра обеим функциям передается номер кодовой страницы, для кириллицы это 1251.</p> <p>Функция <b>gets()</b> (или <b>gets_s</b>) вводит строку в переменную, записанную в скобках.</p> <p>Функция <b>puts()</b> (или <b>puts_s</b>) осуществляет вывод символов на экран. Параметром <b>puts()</b> может быть строковая переменная:</p> <pre><b>char a[] = "Привет!"; puts (a);</b></pre> <p>Функция <b>putchar()</b> предназначена для вывода единичного символа на экран. Параметром функции может быть символ (например: <b>putchar ('H');</b>) или символьная переменная: <b>char letter; letter='G'; putchar (letter);</b></p>

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	$d = \operatorname{tg}(-x \cdot i) / \sqrt{x - z}$ $f = \sin(2d) / d$	$i = -6; x = 4,5;$ $z = 1,5 \cdot 10^{-6}$	11	$w = \operatorname{tg} 1 \cdot (1 + x) + z - e^a$ $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	$t = \operatorname{tg} x + r \cdot (1 - \ln x)$ $u = t/(x^3 + 1)/(1 - e^{k-4})$	$k = 6; r = 5 \cdot 10^{-7}$ $x = 0,095$
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	$t = (d \cdot c + a \cdot \sqrt{c - 1})$ $y = 0,5t / d + e^a$	$d = 0,5 \cdot 10^{-8}; c = -9;$ $j = -4; a = 1,5$

№	Формулы для вычислений	Исходные данные	№	Формулы для вычислений	Исходные данные
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	$y = \cos(5m) / \sin^2(0,4m)$ $w = 4z \cdot y - 7e^{-2 \cdot y}$	$m = 6;$ $z = 0,05 \cdot 10^{-5}$

Для ввода и вывода информации использовать *поточковый* и *форматированный* способы.

№ 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 <b>m</b> на две части, разность которых равна 5.

№ варианта	Условие задачи
3	<p>1. Нарисовать елочку, закрашенную введенным символом.</p> <p>2. Найти площадь прямоугольника, основание которого в <math>n</math> раз больше высоты, а площадь равна периметру.</p>
4	<p>1. Нарисовать круг по центру консольного окна, закрашенный введенным символом.</p> <p>2. Даны действительные числа <math>X, Y, Z</math>. Определить среднее арифметическое и среднее геометрическое их модулей.</p>
5	<p>1. Нарисовать треугольник по центру консольного окна, закрашенный введенным символом.</p> <p>2. Животновод в начале каждой зимы повышает отпускную цену на молоко на <math>p\%</math>, а каждым летом снижает на столько же процентов. Изменится ли цена на молоко и если да, то в какую сторону и на сколько, через <math>n</math> лет?</p>
6	<p>1. Нарисовать звезду, закрашенную введенным символом.</p> <p>2. Определить расстояние на плоскости между двумя точками с заданными координатами <math>M1(x1,y1)</math> и <math>M2(x2,y2)</math>.</p>
7	<p>1. Нарисовать трапецию по центру консольного окна, закрашенную введенными символами.</p> <p>2. Известна длина окружности. Найти площадь круга, ограниченного этой окружностью.</p>
8	<p>1. Нарисовать человечка, закрашенного введенным символом.</p> <p>2. Вычислить площадь треугольника по формуле Герона, если заданы его стороны.</p>
9	<p>1. Нарисовать разнонаправленные стрелки (<math>\rightarrow \uparrow \downarrow \leftarrow</math>), состоящие из введенного символа.</p>

№ варианта	Условие задачи
	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.
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 $\heartsuit$ , закрашенное введенным символом. 2. В бригаде, работающей на уборке сена, имеется $N$ косилок. Первая из них работала $M$ часов, а каждая следующая на 10 мин. Больше, чем предыдущая. Сколько часов работала вся бригада?
14	1. Нарисовать ромб $\blacklozenge$ , 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.



№ варианта	Условие задачи
16	<p>1. Нарисовать грибок, закрашенный введенным символом.</p> <p>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 <b>A</b> самолетов, <b>B</b> ракет и <b>C</b> спутников.</p>

## Лабораторная работа № 14. Разветвляющиеся программы

В разветвляющихся программах производится выбор направления вычислений в зависимости от выполнения некоторых условий.

Задание	Краткие теоретические сведения
<p>1. Выполнить программу, приведенную в правой части, несколько раз с различными значениями переменной <b>j</b>.</p> <p>Переделать программу с тем, чтобы ввод и вывод осуществлялся с помощью <i>поточковых</i> операторов ввода-вывода.</p>	<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <pre> #include &lt;iostream&gt; void main() {     setlocale(LC_CTYPE, "Russian");     float b, z, x = 4e-3, s = 1.1, j;     printf_s("Введите j ");     scanf_s("%f", &amp;j);     b = s + (5 * x + j);      if (b &lt; 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> </div> <p><i>Пример.</i> Вычислить значение <b>z</b> 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 <math>x = 4 \cdot 10^{-3}</math>; <math>s = 1,1</math>; <math>j = \{2; 6,8; 0,03; 55; \dots\}</math>.</p> <p>В программе для организации выбора используется условный оператор <b>if</b>. Формат оператора: <b>if</b> &lt;условие&gt; оператор-1; [<b>else</b> оператор-2;] или <b>if</b> &lt;условие&gt; оператор-1;</p> <p>Выполнение оператора <b>if</b> начинается с проверки условия. Если условие истинно, то выполняется оператор 1, если условие ложно – выполняется оператор 2.</p> <p>Если условие ложно и отсутствует оператор 2, то выполняется следующий за <b>if</b> оператор.</p> <p>При каждом повторении операторов цикла если выражение <math>b &lt; 1,5</math> истинно, то <b>z</b> вычисляется по первой формуле, в противном случае – по второй.</p>

2. Выполнить программу, записанную в правой части.

Изменить программу с тем, чтобы ввод и вывод осуществлялся с помощью *поточковых* операторов ввода-вывода.

```
#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);
}
```

*Пример.* Ввести с клавиатуры вещественные числа  $x$ ,  $y$ ,  $z$ , используя функции форматированного ввода.

Вычислить  $\max(x + y + z, xyz) \cdot \min(x, y, z)$ .

3. Выполнить программу, записанную в правой части, с использованием оператора выбора **switch**.

Записать условие задачи.

Для организации выбора из множества различных вариантов используется оператор выбора **switch**. Формат оператора:

```
switch (выражение)
{ [ case константное выражение 1 ]: [ список операторов 1 ];
.....
[ case константное выражение n ]: [ список операторов n ];
[ default: [ список операторов ]; ]
}
```

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

```
#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;
}
```

ет выполнение функции и возвращает системе значение **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** заверша-

В некоторых случаях приходится использовать оператор безусловного перехода **goto**, который передает управление оператору, помеченному 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 <b>m, n, p</b> . Подсчитать количество отрицательных чисел.
2	Определить, имеется ли среди трёх чисел <b>a, b</b> и <b>c</b> хотя бы одна пара равных между собой чисел.
3	Даны три числа <b>a, b</b> и <b>c</b> . 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 <b>A, B, C, D</b> хотя бы одно нечётное.
6	Дано натуральное трехзначное число <b>n</b> . Верно ли, что среди его цифр есть 0 или 9?
7	В переменную <b>Y</b> ввести номер года. Определить, является ли год високосным.
8	Дано натуральное четырехзначное число <b>n</b> . Верно ли, что все его цифры различны?
9	Проверить, является ли дробь <b>A / B</b> правильной.
10	Число делится на 3 тогда, когда сумма его цифр делится на 3. Проверить этот признак на примере заданного трехзначного числа.

№ варианта	Условие задачи
11	Даны три числа <b>a</b> , <b>b</b> , <b>c</b> . Значение наибольшего из них присвоить переменной <b>d</b> .
12	Есть натуральное двузначное число <b>n</b> . Верно ли, что среди его цифр есть 11 или 99?
13	Для натурального числа <b>K</b> напечатать фразу «мы нашли K грибов в лесу», согласовав окончание слова «гриб» с числом <b>K</b> .
14	Для целого числа <b>K</b> от 1 до 9 напечатать фразу «мне K лет», учитывая при этом, что при некоторых значениях <b>K</b> слово «лет» надо заменить на слово «год» или «года».
15	Определить есть ли среди заданных целых чисел <b>A</b> , <b>B</b> , <b>C</b> , <b>D</b> 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.

1. Разработать программу-калькулятор для целых чисел с операциями  $\%$ ,  $/$ ,  $+$ ,  $-$ ,  $*$ .

2. На шахматном поле в клетке  $(k, l)$  расположен слон. Угрожает ли он фигуре стоящей в клетке  $(m, n)$ ? 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. Возможно ли на прямоугольном участке застройки размером **a** на **b** метров разместить два дома размером **p** на **q** и **r** на **s** метров? Дома можно располагать только параллельно сторонам участка.

5. Пройдет ли шар радиуса **r** через ромбообразное отверстие с диагоналями **p** и **q**?

6. Можно ли коробку размером **a** x **b** x **c** упаковать в посылку размером **r** x **s** x **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.

8. Можно ехать на такси со скоростью  $v_1$  км/ч и оплатой  $p_1$  руб/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.

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.

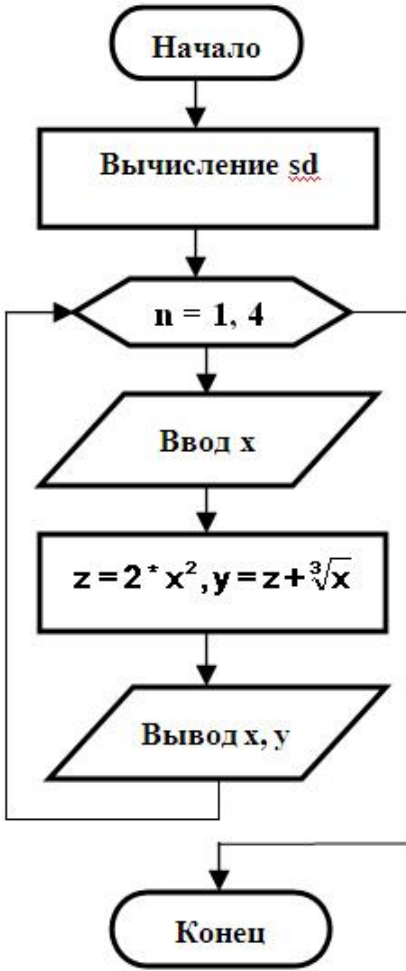
## Лабораторная работа № 15. Циклические программы. Отладка программ

*Цикл* – это участок программы, в котором одни и те же вычисления реализуются неоднократно при различных значениях некоторых переменных, являющихся параметрами цикла.

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

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



Блок-схема алгоритма	Программа
 <pre> graph TD     Start([Начало]) --&gt; Calc[Вычисление sd]     Calc --&gt; Loop{н = 1, 4}     Loop --&gt; Input[/Ввод x/]     Input --&gt; CalcBlock["z = 2 * x², y = z + √³x"]     CalcBlock --&gt; Output[/Вывод x, y/]     Output --&gt; Loop     Loop --&gt; End([Конец]) </pre>	<pre> #include &lt;iostream&gt; void main() {     setlocale(LC_CTYPE, "Russian");     float z, y, x, sd;     sd = 1.0 / 3.0;     for (int n = 1; n &lt;= 4; n++)     {         printf("Введите x ");         scanf_s("%f", &amp;x);         z = 2 * pow(x, 2);         y = z + pow(x, sd);         printf("x = %5.2f\t", x);         printf("y = %5.2f\n", y);     } } </pre>

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 **превратить** 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.

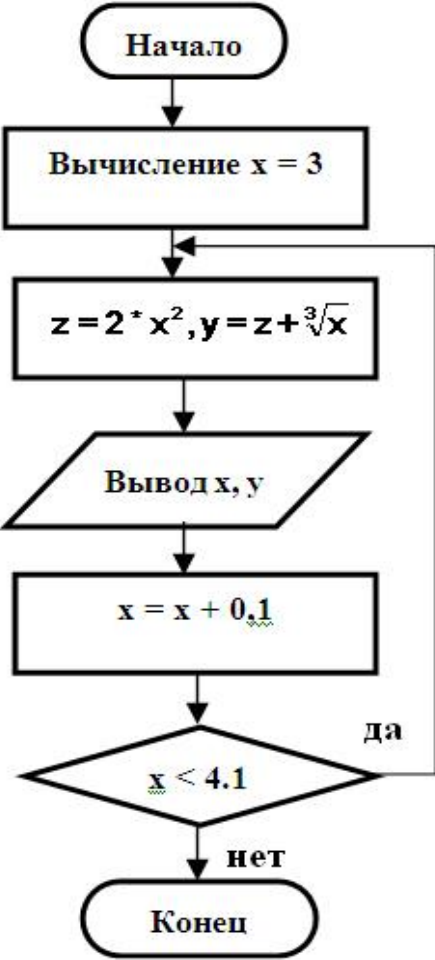
Например, выражение  $(float)int / int = float / int$  даст в результате значение с типом **float**.

3. Выполнить программу с использованием оператора **do while**, записанную в правой части. Проанализировать ее. Изучить блок-схему алгоритма.

*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.

Блок-схема алгоритма	Программа
 <pre> graph TD     Start([Начало]) --&gt; CalcX[Вычисление x = 3]     CalcX --&gt; CalcZY["z = 2 * x^2, y = z + √[3]{x}"]     CalcZY --&gt; Output[Вывод x, y]     Output --&gt; IncX["x = x + 0.1"]     IncX --&gt; Decision{x &lt; 4.1}     Decision -- да --&gt; CalcZY     Decision -- нет --&gt; End([Конец]) </pre>	<pre> #include &lt;stdio.h&gt; #include &lt;cmath&gt;  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 &lt; 4.1); } </pre>

4. Выполнить программу, содержащую вложенный цикл. Записать условие задачи.

Оформить вывод результатов, используя различные возможности операторов вывода.

```
#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");
    }
}
```

Вместо `#include <stdio.h>` в заголовочной части записать:

```
#include <iostream>
#include <iomanip>
using namespace std;
Перед printf("\n"); разместить операторы:
if (i == 1)
{ cout<<endl;
  cout << setw(5*n)<<setfill('-')<< '- '<<endl;
}
Пояснить результат.
```

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.

Средства отладки можно разделить на две группы: средства *интерактивной* отладки (позволяют выполнять программу по шагам); средства *планируемой* отладки (можно спланировать сценарий отладки на множестве шагов).

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 &lt;stdio.h&gt; void main() { float y, x = 0;   while (x &lt;= 6)   { if (x &lt; 4)     y = x;     else       if (x &gt; 4)         y = x * x;       else         y = 1;     printf("x=%f\t", x);     printf("y=%5.2f\n", y);     x = x + 2;   } }</pre>	<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 &lt;= 6; 0 &lt; 4; y = x = 0; вывод x = 0 и y = 0;  x = x + 2 = 0 + 2 = 2;  2 &lt;= 6; 2 &lt; 4; y = x = 2; вывод x = 2 и y = 2;  x = x + 2 = 2 + 2 = 4;  4 &lt;= 6; 4 = 4; y = 1; вывод x = 4 и y = 1;  x = x + 2 = 4 + 2 = 6;  6 = 6; 6 &gt; 4; y = x · x = 6 · 6 = 36; вывод x = 6 и y = 36;  x = x + 2 = 6 + 2 = 8;  8 &gt; 6. Выполнение программы закончено.</p>
<p>2. Выполнить программу из п. 1 с использованием средства <i>интерактивной отладки Шаг с обходом</i>.</p> <p>Внести изменения в программу с тем, чтобы появились ошибки (например, заменить == на = или убрать строчку <b>x = x + 2;</b> и т. п.).</p>	<p>Пошаговое выполнение приложения <b>без трассировки</b> вызываемых программ осуществляется с помощью команды <b>Отладка / Шаг с обходом</b> (или клавиша <b>F10</b> или иконка на панели инструментов <b>Отладка</b>).</p> <p>При выборе отладки <b>Шаг с обходом</b> производится останов перед выполнением текущей строки исходного кода.</p> <p>Выполнение программы продолжается <b>повторным</b> нажатием клавиши <b>F10</b>. При этом <b>Отладчик</b> 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,</p>	

<p>Проверить, как будет выполняться отладка по шагам с обходом.</p>	<p>то появится <b>всплывающее</b> окно, в котором будет указано текущее значение переменной.</p> <p>В режиме отладки появляется окно <b>Видимые</b>, в котором отображаются имена переменных. Здесь можно <b>изменить значение</b> некоторой переменной, сделав двойной щелчок на отображаемом значении и записав новое значение (это средство можно использовать для коррекции программы и для пропуска первых шагов в цикле с большим количеством итераций).</p> <p>Для контроля значений удобно использовать также окно <b>Контрольное значение</b> (предусмотрено до 4 таких окон). Для вызова окна надо выполнить <b>Отладка / Окна / Контрольные значения / Контрольное значение 1</b> (2, 3, 4) или в окне <b>Видимые</b> поместить курсор в столбец <b>Имя</b>, в контекстном меню выполнить команду <b>Добавить контрольное значение</b>. После появления окна надо ввести имя переменной (или выделить нужную переменную в тексте программы и перетащить ее мышкой в поле <b>Имя</b>).</p> <p>После отладки необходимо вывести приложение из отладочного режима командой <b>Отладка / Остановить отладку</b> или <b>Shift + F5</b> или одноименной кнопкой на панели инструментов <b>Отладка</b>.</p>
<p>3. Выполнить программу из п. 1 с использованием средства <i>интерактивной</i> отладки <b>Шаг с заходом</b>.</p> <p>Проанализировать изменения, происходящие в процессе отладки.</p>	<p>Пошаговое выполнение приложения с <b>трассировкой</b> вызываемых программ осуществляется с помощью команды <b>Отладка / Шаг с заходом</b> (или клавиша <b>F11</b>, или иконка на панели инструментов <b>Отладка</b>).</p> <p>При этом производится останов перед выполнением текущей строки исходного кода. Выполнение программы продолжается повторным нажатием клавиши <b>F11</b>. При этом <b>Отладчик</b> заходит в используемые программой функции (библиотечные, функции пользователя и т. п.)</p>
<p>4. Выполнить программу п.1 с использованием средств</p>	<p>При <b>планируемой</b> отладке используются <b>точки останова</b>. Точка останова назначается щелчком левой кнопки мыши в сером поле слева от строки программы (точка оста-</p>

<p><i>планируемой</i> отладки, установив точки останова на операторах вывода и пошагово проследив вывод значений на консоль.</p> <p>Удалить точки останова.</p>	<p>нова отмечается маркером в виде красного круга). Повторный щелчок на маркере точки останова приводит к ее отмене. Точку останова можно назначить также с помощью <b>Отладка / Точка останова</b> или клавишей <b>F9</b>. Она устанавливается на той строке, где помещен курсор.</p> <p>Затем приложение запускается в режиме отладки (<b>F5</b>) или командой меню <b>Отладка / Продолжить</b>. Приложение будет выполнено до точки останова.</p> <p>Продолжение – повторное нажатие клавиши <b>F5</b>.</p> <p>Для контроля значений можно использовать окно <b>Локальные</b> и окно <b>Контрольное значение</b>.</p> <p>Во время одной сессии отладки могут использоваться все три вышеописанных способа: шаг с заходом, шаг обходом и точки останова.</p>
<p>5. Создать исполняемый файл без отладочной информации <b>Release</b>.</p> <p>Объяснить назначение папок и файлов решения проекта.</p> <p>Выполнить файл с расширением <b>*.exe</b> в папке <b>Debug</b>.</p>	<p>После отладки и исправления всех ошибок можно построить приложение <b>без отладочной информации</b>. Для переключения в окончательную конфигурацию необходимо выбрать команду <b>Построение / Диспетчер конфигураций</b>. На экран будет выведено диалоговое окно установки активной конфигурации проекта.</p> <p>Надо выбрать <b>Win32 Release</b> и повторить построение проекта с помощью <b>Построение / Перестроить проект</b>.</p> <p>Каждая конфигурация проекта определяет папки, куда будут помещены файлы с промежуточными и окончательными результатами компиляции и компоновки. По умолчанию это папки <b>Debug</b> и <b>Release</b>, которые располагаются в папке проекта.</p> <p>Папка решения содержит файл текущего примера решения <b>*.sln</b>, файл с информацией о проектах решения и опциях решения <b>*.suo</b>. В файле решения зафиксирован перечень проектов, входящих в решение.</p> <p>Во вложенной папке находятся файлы и папки проекта: файл <b>*.vcproj</b> в формате <b>XML</b> содержит перечень файлов, включенных в проект; файл <b>*.cpp</b> содержит</p>

исходный код программы на языке C++; файлы \*.obj – это 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.

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.

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	$y = \sqrt{i \cdot b - b^2 \cdot a}$ $z = y \cdot \text{tg}(n/4) - e^{1+b}$	$a = 2 \cdot 10^{-3};$ $b = 8,5; n = 2;$ $i = \{2; 1; 8,3\}$	$i = 1(0,5)3$	$i = 2(0,5)3,$ $n = \{3; -6; 0,2; 2,8\}$



№	Формулы для вычислений	Исходные данные (for)	Исходные данные (while)	Исходные данные (двойной цикл)
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	$y = (m \cdot j) / \operatorname{tg} a - e^{10m}$ $z = 2y \cdot b + \sqrt{a+b}$	$b = 2,5; m = 3$ $a = 1,4 \cdot 10^{-3}$ $j = \{0,7; -2; 2,9\}$	$j = b(0,1)3$	$m = \{7,3; -2; 0,8\},$ $j = \{0,3; 1; 0,7; 4\}$
5	$c = i/k - \sqrt{y}/0,4$ $d = e^{1-c} + 4,9(x^2 + 1)$	$i = \{0,9; 8,4; 2\}$ $y = 1,6 \cdot 10^{-4}$ $x = -1; k = 6$	$i = 0(0,5)3$	$x = 1(0,1)2,$ $i = \{1,3; -8; 0,2\}$
6	$v = u + b - 2\sqrt{0,7k + m}$ $w = 100 \cdot e^{-0,2u} - \ln(8,1u)$	$b = -5,4; m = 4$ $u = 0,05 \cdot 10^{-4}$ $k = \{6; 4; 0,3; -7\}$	$k = 3(0,2)4$	$k = 7(0,2)8,$ $m = \{-1,3; -2; 4,9\}$
7	$z = t \cdot y^2 - \sqrt{i+x} \cdot \operatorname{tg} y$ $q = \sqrt{z^2} + 5z \cdot \ln(y)$	$x = 0,9; y = 2;$ $t = 6,96 \cdot 10^{-5}$ $i = \{9; -1,4; 5\}$	$i = 1(0,2)2$	$i = 2(0,2)3,$ $x = \{0,7; -1; 9\}$
8	$w = 0,6j + e^{t \cdot j} \cdot (4y/n)^2$ $s = \sqrt{w-0,1t} / (2+n^2)$	$y = 5; t = -7,4;$ $n = 9; j = \{0,5; 8,4; 0,3\}$	$j = 0(0,1)2$	$y = \{0,1; -3; 0,5\},$ $j = 0,1(0,1)0,4$

№	Формулы для вычислений	Исходные данные (for)	Исходные данные (while)	Исходные данные (двойной цикл)
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	$y = (m - b) / (\sin a - e^a)$ $z = 3y + \sqrt{a - 4jb}$	$b = 0,5; m = 8$ $a = 2,4 \cdot 10^4;$ $j = \{1,7; 2,9; 8\}$	$j = 2(0,5)3$	$j = 2(0,5)3,$ $m = \{0,4; -1; 1,9\}$
13	$d = \sin(k/a) / \cos(m \cdot b)$ $c = d / (d^2 + 1) / (1 - e^k)$	$a = 8; m = 6$ $b = 5 \cdot 10^3;$ $k = \{1,6; 9,1; 8\}$	$k = a(-0,5)3$	$a = 2(0,2)2,8,$ $k = \{1,7; 5; -2\}$
14	$y = a / (b + e^b) / (1 + j \cdot i)$ $t = \cos(y + 1) / \sqrt{ -2j }$	$b = 2; j = 6$ $a = 1,5 \cdot 10^8;$ $i = \{7; 4; 2; 6\}$	$i = 3(-0,1)2$	$b = 2,4(0,2)3,$ $i = \{0,1; -1; 3\}$

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

#### 6. Дополнительные задания.

1. Торговая фирма в первый день работы реализовала товаров на **P** 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?

2. Фирма ежегодно на протяжении **n** лет закупала оборудование стоимостью соответственно **s1, s2, ..., sn** 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?

3. Дана последовательность ненулевых целых чисел, которая заканчивается числом 0. Определить, сколько раз в этой последовательности меняется знак.

4. Дано натуральное число **k**. Определить **k**-ю цифру последовательности: 1248163264 ..., в которой выписаны подряд степени 2.

5. Составить алгоритм решения ребуса КОТ + КТО = ТОК (различные буквы означают различные цифры, старшая буква – не 0).

6. Натуральное число, записанное в десятичной системе счисления, называется сверхпростым, если оно остается простым при любой перестановке своих цифр. Определить все сверхпростые числа от 1 до **n**.

6. В соответствии со своим вариантом написать программу и отладить ее для задачи, представленной в таблице ниже. Опробовать средства *интерактивной* отладки **Шаг с обходом**.

При наличии ошибок из-за некорректных исходных данных выполнить вычисления с другими числами. Вызвать окно **Контрольное значение 1**, поместить туда имена переменных и проконтролировать их значения в процессе выполнения программы. В отчете представить результаты в окне **Отладчика** (окно **Видимые** и окно **Контрольное значение 1**).

Для той же программы представить результаты в окнах **Отладчика** для интерактивной отладки **Шаг с заходом**.

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

№	Формулы для вычислений	Исходные данные	№	Формулы для вычислений	Исходные данные
5	$p = e^{\sin(j/x)} \cdot \ln(x/y) \cdot x$ $q = \begin{cases} \sqrt{p/m}, & p \leq y^2 \\ \sqrt{2x/(j+p)}, & p > y^2 \end{cases}$	$y = 1,4 \cdot 10^{-3};$ $x = 0,2; m = 4;$ $j = 5(2)11$	10	$y = \cos^2 x^2 /  x $ $z = \begin{cases}  y , & y < a \cdot x \cdot n \\ \sqrt{1 + e^{-y}}, & y \geq a \cdot x \cdot n \end{cases}$	$a = 1,055; x = 0,6;$ $n = 6(2)14$
11	$d = i + 2t \cdot (1 + \sqrt{3a^2})$ $f = \begin{cases} t \cdot i, & d \geq t(t+i) \\ e^{t-d} + 9, & d < t(t+i) \end{cases}$	$a = -4,2;$ $t = \{5; 1; -3; 9; -1\}$ $i = 4$	14	$v = \begin{cases} \cos^2(j \cdot x), & 2s > x \cdot j \\ 2 \cdot \operatorname{tg}(j \cdot x), & 2s < x \cdot j \\ 5 - e^{x/2}, & 2s = x \cdot j \end{cases}$	$s = 5,2;$ $x = \{5; 0,4; -2,1; 6\}$ $j = 10,4 \cdot 10^4$
12	$x = \operatorname{tg}(a^2 - 1) / (d + 1)$ $y = \begin{cases} a \cdot k + d, & 3x < a \cdot c \\ \cos(a \cdot k) \cdot e^{a+1}, & 3x \geq a \cdot c \end{cases}$	$c = 3,7(0,1)5;$ $d = 51,9 \cdot 10^{-2};$ $a = 4; k = 1$	15	$b = 12s - e^{-s/2} \cdot (x - j)$ $z = \begin{cases} \sqrt{-2x \cdot j} + b, & b < 1,5 \\  13x \cdot j  + b, & b \geq 1,5 \end{cases}$	$x = -4 \cdot 10^{-3}; s = 1,1;$ $j = 4(0,5)7$
13	$t = a \cdot i / (a^2 - b) \cdot e^{-a}$ $x = \begin{cases} 4,8 \cdot 10^{-3} + i \cdot a, & t > 5c \\ a + i^2 \cdot t, & t \leq 5c \end{cases}$	$b = 3; i = 8(4)24$ $c = 6 \cdot 10^{-4}; a = 6$	16	$z = \begin{cases} \sin^2(t), & t > x \\ 4 \cdot (t + x), & t < x \\ 1 - e^{x-2}, & t = x \end{cases}$	$t = 1,5(0,1)2;$ $x = 1,9$

7. 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** и окно **Контрольное значение 1**).

Создать *исполняемый файл* и ознакомиться с файлами в папках проекта.

№ варианта	Условие задачи
1	Используя перебор значений, вывести на экран в возрастающем порядке все трехзначные числа, в десятичной записи которых нет одинаковых цифр.
2	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 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.
14	Используя перебор значений, вывести на экран в возрастающем порядке все трехзначные числа, в де-

№ варианта	Условие задачи
	сятичной записи которых нет одинаковых цифр.
15	Даны натуральные числа <b>q</b> и <b>b</b> . Используя перебор значений, получить все меньшие <b>q</b> натуральные числа, квадрат суммы цифр которых равен <b>b</b> .
16	В заданной последовательности натуральных чисел (размер и значения вводятся пользователем) найти наибольший элемент из отрицательных.

8. 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 – простое число (число 121 – полный квадрат, т. к.  $121=11 \cdot 11$ , а 11 – простое число).
2. Составить алгоритм, определяющий, сколько существует способов набора одного рубля при помощи монет достоинством 50 коп., 20 коп., 5 коп. и 2 коп.
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 т. д. 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. Найти минимальное число, которое представляется суммой четырех квадратов натуральных чисел не единственным образом.

## Лабораторная работа № 16. Решение инженерных задач на основе циклических программ. Указатели и ссылки

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



<p>2. Изучить алгоритм вычисления определенного интеграла методом парабол.</p>	<p>5. Вывод <b>z</b>.</p> <p>При использовании <i>метода парабол</i> интервал <b>[a, b]</b> делится на четное количество частей – <b>2n</b>. Тогда <math>\mathbf{h} = (\mathbf{b} - \mathbf{a}) / (2 \cdot \mathbf{n})</math>; <math>\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{h}</math>; <math>\mathbf{i} = 1, 2, \dots, 2\mathbf{n}</math>.</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"> <li>1. Ввод <b>a, b, n</b>.</li> <li>2. Вычисление <math>\mathbf{h} = (\mathbf{b} - \mathbf{a}) / (2 \cdot \mathbf{n})</math>; <math>\mathbf{x} = \mathbf{a} + 2\mathbf{h}</math>; <math>\mathbf{s1} = 0</math>; <math>\mathbf{s2} = 0</math>; <math>\mathbf{i} = 1</math>.</li> <li>3. Расчет <math>\mathbf{s2} = \mathbf{s2} + \mathbf{f}(\mathbf{x})</math>; <math>\mathbf{x} = \mathbf{x} + \mathbf{h}</math>; <math>\mathbf{s1} = \mathbf{s1} + \mathbf{f}(\mathbf{x})</math>; <math>\mathbf{x} = \mathbf{x} + \mathbf{h}</math>; <math>\mathbf{i} = \mathbf{i} + 1</math>.</li> <li>4. Если <math>\mathbf{i} &lt; \mathbf{n}</math>, то переход к пункту 3, иначе – переход к следующему пункту.</li> <li>5. Вычисление значения интеграла:</li> </ol> $z = \frac{h}{3}(f(a) + 4f(a+h) + 4s1 + 2s2 + f(b))$ <p>6. Вывод <b>z</b>.</p> <p>Здесь <math>\mathbf{s1} = y_3 + y_5 + \dots + y_{2n-1}</math>, а <math>\mathbf{s2} = y_2 + y_4 + \dots + y_{2n-2}</math>.</p>
<p>3. Изучить графический способ отделения корней и <i>метод касательных</i> для решения уравнения.</p>	<p><b>Решение уравнения</b> численными методами состоит из двух этапов: <i>отделение</i> корней, т. е. нахождение таких отрезков <b>[a, b]</b> на оси <b>OX</b>, внутри которых имеется один корень; <i>вычисление</i> корней с заданной точностью.</p> <p>Графический метод отделения корней. Отделить корни можно, построив, например, график функции <b>f(x)</b> в приложении <b>Excel</b>. Из тех отрезков, на которых функция пересекает ось <b>OX</b>, рассмотрим один <b>[a, b]</b>.</p> <p>При использовании <i>метода касательных</i> для вычисления корня уравнения <math>\mathbf{f}(\mathbf{x}) = 0</math> необ-</p>

ходимо 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.

Последовательные приближения корня рассчитываются по формуле

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}, \quad n = 0, 1, 2, \dots$$

Вычисления продолжаются пока не будет выполнено условие:  $|x_{n+1} - x_n| \leq e$ , где  $e$  – требуемая точность вычисления корня.

Алгоритм метода касательных:

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  $x_1$ .

4. Изучить метод *дихотомии* для решения уравнения.

Согласно *методу дихотомии* 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  $2e$ .

Алгоритм метода дихотомии:

1. Ввод значений  $a, b, e$ .
2. Вычисление  $x = (a + b) / 2$ .
3. Если  $f(x)f(a) \leq 0$ , то  $b = x$ , иначе –  $a = x$ .

	<p>4. Если <math> a - b  &gt; 2e</math>, то переход к пункту 2, иначе – переход к следующему пункту.</p> <p>5. Вывод значения корня <math>x</math>.</p>		
<p>5. Изучить способ нахождения корня уравнения в приложении Excel с помощью команды <b>Подбор параметра</b>.</p>	<p>В приложении Excel для решения уравнения имеется команда <b>Подбор параметра</b>. Чтобы решить с помощью этой команды, например, уравнение <math>4 - x^2 + x = 0</math>, надо на рабочем листе, например, в ячейке <b>A1</b>, записать начальное приближение корня (например, 2) , в ячейке <b>B1</b> – само уравнение: <math>= 4 - A1^2 + A1</math></p> <p>Выполнить <b>Данные / Работа с данными / Анализ “что-если” / Подбор параметра</b>.</p> <p>В появившемся окне задать следующие значения: в поле <b>Установить в ячейке</b> выбрать <b>B1</b>, в поле <b>Значение</b> ввести <b>0</b>, в поле <b>Изменяя значение ячейки</b> – <b>A1</b>.</p> <p>После нажатия <b>ОК</b> в ячейке <b>A1</b> будет корень уравнения.</p>		
<p><b>Задание</b></p>	<p><b>Краткие теоретические сведения</b></p>		
<p>1. Изучить работу с указателями, выполнив программы, записанные в правой части.</p> <p>Использовать средства отладки.</p> <p>Записать условия задач.</p>	<p><b>Указатель</b> – это переменная, значением которой является <b>адрес ячейки памяти</b>, в которой содержится значение другой переменной.</p> <p>В памяти компьютера указатель обычно занимает 4 байта.</p> <p>Пусть объявлена переменная и указатель на нее: <b>int a = 0; int *ptr;</b></p> <p>Можно в указатель поместить адрес переменной и напечатать: <b>ptr = &amp;a; cout &lt;&lt; ptr;</b></p> <p>Здесь <b>&amp;</b> – операция получения адреса переменной, <b>*</b> – операция разыменования (получения значения переменной, на которую указывает указатель).</p> <table border="1" data-bbox="618 1058 2051 1219"> <tr> <td data-bbox="618 1058 1319 1219"> <pre>#include &lt;iostream&gt; using namespace std; void main()</pre> </td> <td data-bbox="1319 1058 2051 1219"> <pre>#include &lt;iostream&gt; void main() {   using namespace std;</pre> </td> </tr> </table>	<pre>#include &lt;iostream&gt; using namespace std; void main()</pre>	<pre>#include &lt;iostream&gt; void main() {   using namespace std;</pre>
<pre>#include &lt;iostream&gt; using namespace std; void main()</pre>	<pre>#include &lt;iostream&gt; void main() {   using namespace std;</pre>		

```

{   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;
}

```

```

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;
}

```

При использовании средств отладки в окне **Видимые** или **Локальные** или **Контрольное значение** рядом с именем переменной может стоять знак плюс. Это означает, что для этой переменной может быть отображена дополнительная информация, которая отображается после щелчка по символу «+» и представляет собой значение, хранимое в памяти по адресу, содержащемуся в указателе.

Для изучения расположения данных в памяти при использовании указателей полезно проконтролировать содержимое памяти компьютера с помощью **Отладка / Окна / Память / Память 1** (2, 3, 4).

Вывод адресов переменных и содержимого переменных производится в *шестнадцатеричной* системе счисления.

2. В правой части записаны фрагменты программ с использованием указателя на константу, константного указателя, константного

Указатель на константу	Константный указатель	Константный указатель на константу
<pre> {   int a = 9;     const int *pa = &amp;a; </pre>	<pre> {   int a = 99;     int* const pa = &amp;a; </pre>	<pre> {   int a = 99;     const int* const pa = &amp;a; </pre>

<p>указателя на константу.</p> <p>Убрать ошибочные операторы, дописать операторы вывода и выполнить программы на компьютере.</p>	<pre><b>*pa = 12; // ошибка</b> <b>(нельзя менять значение</b> <b>переменной a)</b> <b>int b = 20;</b> <b>pa = &amp;b;</b> <b>}</b></pre>	<pre><b>int b = 44;</b> <b>pa = &amp;b; // ошибка</b> <b>(константный указатель</b> <b>менять нельзя)</b> <b>*pa = 12;</b> <b>}</b></pre>	<pre><b>*pa = 33; // ошибка (нельзя менять</b> <b>содержимое переменной a)</b> <b>int b = 44;</b> <b>pa = &amp;b; // ошибка (константный</b> <b>указатель на константу менять нельзя)</b> <b>}</b></pre>		
<p>3. Написать программу генерации элементов массива <b>A</b> из случайных чисел, их вывода и определения максимального элемента массива двумя способами (с указателями и без них).</p>	<p>Имя массива <b>A</b> без индекса является <b>указателем-константой</b> (не изменяется на протяжении всей работы программы), т. е. адресом первого по счету элемента массива <b>A[0]</b>.</p> <p>Цикл, в котором генерируется и выводится массив <b>A</b>, содержащий <b>size</b> элементов (случайные числа от 0 до <b>n - 1</b>), можно реализовать тремя способами:</p> <p>1. <b>for ( i = 0; i &lt; n; i++)</b>    2. <b>for ( i = 0; i &lt; n; i++)</b>    3. <b>int n, *pk;</b>  <b>A[i] = rand()%99;</b>            <b>*(A + i) = rand()%99;</b>    <b>for ( pk = A; pk &lt; A + n; pk++)</b>  <b>*pk = rand() % 99;</b></p>				
<p>4. Изучить отличия ссылок от указателей. Выполнить программы, записанные в правой части.</p>	<p><b>Ссылку</b> (ссылочный тип) можно рассматривать как <i>альтернативное</i> имя переменной или как указатель, который всегда разыменовывается. Например: <b>int kol; int &amp;pal = kol;</b></p> <p>Здесь <b>&amp;</b> – оператор ссылки, означающий, что следующее за ним имя является именем переменной ссылочного типа, <b>pal</b> – альтернативное имя для <b>kol</b>.</p> <table border="1" data-bbox="660 1103 1890 1148"> <tr> <td data-bbox="660 1103 1263 1148">Использование ссылки</td> <td data-bbox="1263 1103 1890 1148">Переход от ссылки к указателю</td> </tr> </table>			Использование ссылки	Переход от ссылки к указателю
Использование ссылки	Переход от ссылки к указателю				

```
#include <stdio.h>
void main()
{   int V = 1;
    printf("V = %d\n", V);
    int &rV = V;
    rV = 5;
    printf("V = %d\n", V);
}
```

```
#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);
}
```

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.

5. Выполнить программу, приведенную в правой части, которая разработана с использованием указателей.

Внести изменения с тем, чтобы программа стала содержать ошибки. Исследовать их с помощью отладки.

```
#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;
}
```

*Пример.* Пусть имеется массив А. Значения элементов массива инициализируются в программе. Удалить элемент с номером, который вводится с клавиатуры.

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.

№	Подынтегральная функция f(x)	Пределы интегрирования	№	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 (т. е. 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.

№	Функция $f(x)$	№	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	$x^2 + 4x - 2$



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 <b>N</b> . Выделить из этого числа цифры, кратные <b>m</b> , и записать их в одномерный массив.
2	Даны массивы <b>A</b> и <b>B</b> , состоящие из <b>n</b> элементов. Построить массив <b>S</b> , каждый элемент которого равен сумме соответствующих элементов массивов <b>A</b> и <b>B</b> .
3	Заданы два массива <b>A(n)</b> и <b>B(n)</b> . Подсчитать в них количество элементов, меньших значения <b>t</b> , 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 <b>A</b> из <b>n</b> элементов и <b>B</b> из <b>m</b> элементов. Содержится ли наибольший элемент массива <b>A</b> в массиве <b>B</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 <b>A</b> из <b>n</b> элементов и число <b>k</b> . Найти номера всех элементов массива, которые равны, больше и меньше <b>k</b> .
8	Дан массив <b>x</b> , содержащий <b>k</b> элементов, и <b>y</b> , содержащий <b>n</b> элементов. Найти их «пересечение» т.е. массив

№ варианта	Условие задачи
	<b>z</b> , содержащий их общие элементы, причем кратность каждого элемента в массиве <b>z</b> равняется минимуму из его кратностей в массивах <b>x</b> и <b>y</b> .
9	Даны <b>N</b> 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 <b>n</b> натуральных чисел. Необходимо определить наименьшее натуральное число, отсутствующее в последовательности.
12	Заданы два массива <b>A</b> и <b>B</b> , каждый из <b>n</b> элементов. Подсчитать количество таких <b>k</b> , для которых: <b>A[k] = B[k]</b> , <b>A[k] &gt; B[k]</b> и <b>A[k] &lt; B[k]</b> .
13	Дан массив <b>x</b> , содержащий <b>n</b> элементов. Найти количество различных чисел среди элементов этого массива.
14	Даны два массива <b>x</b> и <b>y</b> . Найти количество одинаковых элементов в этих массивах, т. е. количество пар <b>x[i] = y[j]</b> для некоторых <b>i</b> и <b>j</b> .
15	Дан массив <b>A</b> размера <b>n</b> , не содержащий нулевых элементов. Преобразовать массив <b>A</b> так, чтобы вначале шли положительные элементы, а затем отрицательные. Дополнительные массивы не использовать.
16	Дан массив <b>x</b> , содержащий <b>k</b> элементов, массив <b>y</b> , содержащий <b>n</b> элементов, и число <b>q</b> . Найти сумму вида <b>x[i] + y[j]</b> , наиболее близкую к числу <b>q</b> .

## Лабораторная работа № 17. Одномерные массивы. Многомерные 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 А.</p> <p>Произвести отладку.</p>	<pre> #include &lt;iostream&gt; #include &lt;ctime&gt; void main() {   setlocale(LC_CTYPE, "Russian");     using namespace std;     const int N = 100;     int i, sz, A[N];   int rmn = 0, rmx = 99;     cout &lt;&lt; "Введите размер массива ";     cin &gt;&gt; sz;     cout &lt;&lt; "Массив А:" &lt;&lt; endl;     srand(((unsigned)time(NULL)));     for (i = 0; i &lt; sz; i++)     {   A[i] = (int)((((double)rand() /         (double)RAND_MAX) * (rmx - rmn) + rmn);         // или A[i] = rand() % rmx;         cout &lt;&lt; A[i] &lt;&lt; endl;     } } </pre>	<p><i>Пример.</i> Сформировать одномерный статический массив целых чисел <b>A</b>, используя датчик случайных чисел (в диапазоне от 0 до 99).</p> <p>В программе массив заполняется случайными числами с помощью функции <b>rand</b> из стандартной библиотеки, которая генерирует целое число в диапазоне от 0 до <b>RAND_MAX</b> (символическая константа, равная 32767).</p> <p>При каждом запуске программы будут генерироваться одни и те же случайные числа.</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.

При использовании функции **time** нужно включить в заголовок директивы **#include <ctime>** или **#include <locale>**

2. Выполнить программу, записанную в правой части. Записать ее условие.

Добавить в программу операторы вычисления среднего значения исходного массива.

Произвести отладку.

```
#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;
}
```

Инициализация массива означает присвоение начальных значений его элементам при объявлении.

Массивы можно инициализировать списком значений, заключенных в фигурные скобки, например в программе:

```
float A[ ] = {5, -4, 17.1, 9, 1};
```

Длина массива вычисляется компилятором по количеству значений, перечисленных в фигурных скобках.

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

Опробовать программу для массивов **A** и **B** разного размера.

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

```
#include <stdio.h>
void main()
{ const int size = 5;
  int A[size], B[size], C[size*2]; int k = 0, j = 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[j])  C[i++] = A[k++]; else  C[i++] = B[j++];
      if (k == size)
          for (; j < size; j++)
              C[i++] = B[j];

      if (j == 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");
}
```

4. Выполнить программу, приведенную в правой части.

Внести изменения с тем, чтобы вычислялся минимальный элемент массива.

Произвести отладку.

```
#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;
}
```

*Пример.* Сформировать массив целых чисел в количестве не более 30. Размерность массива ввести с клавиатуры.

Найти в массиве наибольший элемент.

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: <b>D[2][4]</b>. Осуществить вывод массива в виде матрицы.</p>	<pre><b>#include &lt;iostream&gt;</b> <b>void main()</b> <b>{</b>     <b>const int c_i = 3;</b>     <b>const int c_j = 2;</b>     <b>int a[c_i][c_j] = { { 1,2 },{ 3,4 },{ 5,6 } };</b>     <b>int i, j;</b>     <b>for (i = 0; i &lt; c_i; i++)</b>         <b>for (j = 0; j &lt; c_j; j++)</b>             <b>std::cout &lt;&lt; "\n a[" &lt;&lt; i &lt;&lt; ", " &lt;&lt; j &lt;&lt; "] = " &lt;&lt; a[i][j];</b> <b>}</b></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>	<p>Пусть объявлен двумерный массив, например:</p> <pre><b>int A[4][3];</b></pre> <p>В памяти выделяется участок для хранения значения переменной <b>A</b>, которая является указателем на первый элемент массива указателей (для данного примера – массив из четырех указателей). Каждый из этих четырех указателей содержит адреса первых элементов строк матрицы <b>A[4][3]</b>.</p> <p>Для доступа к элементам массива <b>A</b> через указатели можно использовать следующую форму записи: <b>*(A + 1) + 2</b> или <b>*(A[1] + 2)</b> или <b>*(A + 1)[2]</b></p>	

	<pre> #include &lt;iostream&gt; void main() {     const int n = 3, m = 2;     int i, j, h = 0;     int A[n][m] = { 0, 1, 2, 3, 4, 5 };     for (i = 0; i &lt; n; i++)         for (j = 0; j &lt; m; j++)             h += A[i][j];     std::cout &lt;&lt; h; } </pre>	<pre> #include &lt;iostream&gt; 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 &lt; n; i++)         for (j = 0; j &lt; m; j++)             h += (*(C + i) + j);     std::cout &lt;&lt; h; } </pre>
<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 каждой строки с помощью указателей.</p>	<pre> #include &lt;iostream&gt; 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 &lt;&lt; "Введите элементы массива" &lt;&lt; endl;     for (i = 0; i &lt; n; i++)         for (j = 0; j &lt; m; j++)             cin &gt;&gt; B[i][j];     int min = B[0][0];     for (int i = 0; i &lt; n; i++)         for (int j = 0; j &lt; m; j++)             if (min &gt; B[i][j]) </pre>	



```

        {   min = B[i][j];
          column = 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[" << column << ", " << row << "] = " << min <<
endl;
}

```

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 <b>k</b> . Добавить после каждого четного элемента массива элемент со значением 0.
2	Все четные элементы целочисленного массива <b>K(n)</b> поместить в массив <b>L(n)</b> , а нечетные – в массив

№ варианта	Условие задачи
	<b>M(n)</b> . 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 <b>C</b> 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.
11	Удалить пять первых нечетных элементов массива. Добавить в конец массива три новых нулевых элемента.
12	Найти минимальный элемент массива <b>T</b> и заменить им четные по номеру элементы.
13	В массиве <b>A(n)</b> каждый элемент, кроме первого, заменить суммой всех предыдущих.
14	В массиве несколько нулевых элементов. Найти первый и последний нулевые элементы. Вывести их

№ варианта	Условие задачи
	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.

6. 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**.

7. 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 массив).
6. Разделить массив на две части, поместив в первую элементы, большие среднего арифметического их суммы, а во вторую - меньшие (части не сортировать).

7. В массиве  $A(2n + 1)$ , не содержащем одинаковых элементов, найти средний по величине элемент, т. е. такой, при котором в массиве  $A$  ровно  $n$  элементов меньше его и столько же элементов больше его. Массив  $A$  сохранить (не сортировать), дополнительных массивов не использовать

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.
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) $a_{ij} > b_{ij}$ .

№ варианта	Условие задачи
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$ .
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.

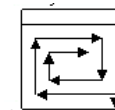
### 8.

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.

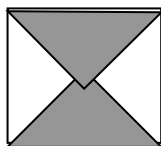


5. Путем перестановки элементов квадратной вещественной матрицы добиться того, чтобы ее максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции  $(2,2)$ , следующий по величине – в позиции  $(3,3)$  и т. д., заполнив таким образом всю главную диагональ.

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.

## Лабораторная работа № 18. 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 &lt;stdio.h&gt; void main() {   int num = 5; char symb, new_symbol = ' ';     symb = num + '0';     printf("%c\n", symb);     if (symb &gt;= '0' &amp;&amp; symb &lt;= '9') num = symb - '0';     printf(" %d\n", num);     symb = 'b';     if (symb &gt;= 'a' &amp;&amp; symb &lt;= 'z')         new_symbol = symb - 'a' + 'A';     printf(" %c\n", new_symbol); }</pre> <p>Заменить третью, четвертую и пятую строчки на:</p> <pre>int num = 5; char *pc; pc = &amp;symb; *pc = num + '0'; printf("%c\n", *pc);</pre> <p>Объяснить результат.</p>	<pre>#include &lt;stdio.h&gt; void main() {     char str[] = "Text";     int count = 0;     char *pstr;     pstr = &amp;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>



2. Выполнить программу, приведенную в правой части.

Объяснить принцип использования функций стандартной библиотеки.

Пусть имеется адрес файла в сети, например: <http://belstu.by/p1/p2/file1.htm>  
 Определить имя последней папки (каталога).

```
# include <cstring>
#include <iostream>
using namespace std;
void main()
{   char str[50];
    int i, index_last, index_first, j;
    cout<<"Input string:"; cin>> str;
    for (i = strlen(str); i >= 0; i--)
        if (str[i] == '\\') break;
    index_last = i;
    for (j = (index_last - 1); j >= 0; j--)
        if (str[j] == '\\') break;
    index_first = j;
    if (index_last == index_first)
        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";
    }
}
```

Функция	Краткое описание функции
<b>strcmp</b>	<b>int strcmp(const char *str1, const char *str2);</b> Сравнивает строки str1 и str2. Если str1 < str2, то результат <0, если str1 = str2, то результат = 0, если str1 > str2, то результат >0.
<b>strcpy</b>	<b>char* strcpy(char*s1, const char *s2);</b> Копирует байты из строки s1 в строку s2
<b>strlen</b>	<b>int strlen (const char *str);</b> Вычисляет длину строки str
<b>strncat</b>	<b>char *strncat(char *s1, const char *s2, int kol);</b> Приписывает kol символов строки s1 к строке s2
<b>strncpy</b>	<b>char *strncpy(char *s1, const char *s2, int kol);</b> Копирует kol символов строки s2 в строку s1
<b>strnset</b>	<b>char *strnset(char *str, int c, int kol);</b> Заменяет первые kol символов строки s1 символом c
<b>atoi</b>	<b>int atoi(char *str);</b> Преобразует числовые символы в целое число
<b>atof</b>	<b>float atof(char *str);</b> Преобразует числовые символы в вещественное число

3. В правой части записаны два варианта решения задачи. Выполнить программы и объяснить различия между ними.

Внести изменения в программу с тем, чтобы проверялось не только количество скобок, но и правильность их расстановки (первой в тексте была открывающая скобка).

Пример проверки соответствия количества открывающих и закрывающих круглых скобок в строке.

```
#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 <b>n</b> символов, начиная с позиции <b>k</b> .
2	Написать программу, реализующую выделение подстроки <b>S1</b> длиной <b>k</b> с позиции номер <b>n</b> 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>b</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 <b>a</b> добавить символ <b>!</b>
8	Написать программу, которая осуществляет сравнение двух строк и выводит сообщение о том, какие символы совпадают.
9	Написать программу, реализующую вставку подстроки <b>S1</b> длиной <b>k</b> в строку <b>S</b> с позиции номер <b>n</b> .
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 предложении указать слово, в котором доля гласных (А, Е, I, О, U – строчные или прописные) максимальна.
  3. В имеющемся словаре найти группы слов, записанных одними и теми же буквами и отличающиеся только порядком расположения.
  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 преобразовать в 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.

## Лабораторная работа № 19. 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
<p>1. Выполнить программы, записанные в правой части и изучить использование <i>указателя как формального параметра</i> функции.</p>	<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 <math>s = 1! / x + 2! / x + \dots + n! / x</math> для <math>x = -2(0.5)0</math>. В процессе вычислений заменить <math>x = 0</math> на <math>x = 1</math>.</p> <p>В таблице ниже приведены два варианта программы для одной и той же задачи. При выводе результатов в программе слева видно, что хотя функция считает значение <math>s</math> при <math>x = 1</math>, в основной программе по-прежнему <math>x = 0</math>.</p> <p>Чтобы эту ошибку исправить, надо формальный параметр <math>x</math> определить как <i>указатель</i>. Тогда фактический параметр <math>x</math> должен быть явно передан в виде указателя с использованием операции <b>&amp;</b> 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;
        x = x + 0.5;
    }
}

```

```

#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>    } while (x &lt;= 0.0); }</pre>	<pre>    } while (x &lt;= 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 &lt;stdio.h&gt; #include &lt;ctype.h&gt; 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>Пример программы, осуществляющей вывод букв в верхнем регистре.</p> <p>Здесь функция <b>toupper</b> выполняет преобразование строчных букв в прописные. Для ее использования нужна директива <b>#include &lt;ctype.h&gt;</b></p>
<p>3. Выполнить программы, записанные в правой части и изучить один из способов передачи массива в функцию, а также</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><u>Пример.</u> Пусть имеется массив <b>V</b>. С использованием функции определить минимальный элемент массива в подпрограмме и в основной программе увеличить его значение на 1.</p>	

использование указателя как результата выполнения функции.

Ниже даны два варианта программы. Программа в левой части выполниться не сможет. Использование указателя как результата выполнения функции в правой части позволяет решить задачу.

```
#include <iostream>
int pmin(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 };
    (pmin(B, 5))++;
    for (int i = 0; i < 5; i++)
        printf("%d ", B[i]);
}
```

```
#include <iostream>
int *pmin(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 };
    (*pmin(B, 5))++;
    for (int i = 0; i < 5; i++)
        printf("%d ", B[i]);
}
```

4. Программа вычисления интеграла справа демонстрирует применение указателя на функцию, который используется для передачи имени функции **f** в качестве формального параметра функции **integ**.

Программа вычисления интеграла  $\int_0^{10} (2x + 5)dx$  методом правых прямоугольников

```
#include <iostream>
float integ(float(*) (float), float, float, float); //прототип
float f(float); //прототип
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*.

## Лабораторная работа № 20. Массивы и ссылки при работе с функциями

Задание	Краткие теоретические сведения	
<p>1. Изучить использование <i>ссылок как формальных параметров</i>, выполнив программы, записанные в правой части</p>	<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 <b>&amp;</b> после типа параметра.</p> <p>При передаче параметра функции в виде ссылки нет необходимости в разыменовании указателей.</p>	
	<p><u>Пример</u> программы обмена значениями переменных с использованием <i>указателей</i></p>	<p><u>Пример</u> программы обмена значениями переменных с использованием <i>ссылок</i></p>
	<pre>#include &lt;iostream&gt; void swap(int *a, int *b) {     int temp = *a;     *a = *b;     *b = temp; } void main() {     int a = 10, b = 100;     swap(&amp;a, &amp;b);     std::cout &lt;&lt; a &lt;&lt; ' ' &lt;&lt; b; //a=100, b=10 }</pre>	<pre>#include &lt;iostream&gt; void swap(int&amp; a, int&amp; b) {     int temp = a;     a = b;     b = temp; } void main() {     int a = 10, b = 100;     swap(a, b);     std::cout &lt;&lt; a &lt;&lt; ' ' &lt;&lt; b; //a=100, b=10 }</pre>

2. Изучить использование *ссылки* как *результата выполнения функции*, выполнив программу, записанную в правой части.

Опробовать работу программы с разными значениями массива **A**.

Пример. Массив **A** содержит набор значений. Необходимо в подпрограмме определить минимальный элемент и изменить его на другое значение в основной программе.

```
#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* типа, опробовав работу программы в правой части.

```
#include <iostream>
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_STYPE, "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 ";
}
```

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

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><code>#include &lt;iostream&gt; int sum(int n, ...) {     int *p = &amp;n;     int sum = 0;     for (int i = 1; i &lt;= n; i++)         sum += *(++p);     return sum; } void main() {     std::cout &lt;&lt; sum(6, 4, 5, 1, 2, 3, 0)&lt;&lt;std::endl;     std::cout &lt;&lt; sum(2, 34, 4445); }</code></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.

№ варианта	Условие задачи
1	1. Дана целочисленная прямоугольная матрица. Определить номер первого из столбцов, содержащих хотя бы один нулевой элемент.

№ варианта	Условие задачи
	2. Написать функцию <b>fmin</b> с переменным числом параметров, которая находит минимальное из чисел типа <b>int</b> .
2	<p>1. Дана целочисленная прямоугольная матрица. Определить количество строк, содержащих хотя бы один нулевой элемент.</p> <p>2. Написать функцию <b>fsum</b> с переменным числом параметров, которая находит сумму чисел типа <b>int</b> по формуле: <math>S=a1*a2+a2*a3+a3*a4+ \dots</math></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 <b>fmax</b> с переменным числом параметров, которая находит минимальное из чисел типа <b>int</b> или из чисел типа <b>double</b>, 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 <b>days</b> 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> <p>2. Написать функцию <b>kvadr</b> с переменным числом параметров, которая определяет количество чисел, являющихся точными квадратами (2, 4, 9, 16,...) типа <b>int</b>.</p>
6	1. Проверить, все ли строки матрицы содержат хотя бы один нулевой элемент. Если нет, то заменить значения всех отрицательных элементов матрицы на нулевые.

№ варианта	Условие задачи
	<p>2. Написать функцию <b>as</b> с переменным числом параметров, которая находит сумму чисел типа <b>int</b> по формуле: <math>S=a1*a2-a2*a3+a3*a4-\dots</math></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 <b>mn</b> с переменным числом параметров, которая находит минимальное из чисел типа <b>int</b> или из чисел типа <b>double</b>, 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 <b>prost</b> с переменным числом параметров, которая находит все простые числа из нескольких интервалов. Интервалы задаются границами <b>a</b> и <b>b</b>. С помощью этой функции проверить несколько интервалов.</p>
9	<p>1. Дана матрица размером 4 x4. Найти сумму наименьших элементов ее нечетных строк и наибольших элементов ее четных строк.</p> <p>2. Во введенном тексте подсчитать количество символов в слове максимальной длины (слова разделяются пробелами) с помощью функции с переменным числом параметров.</p>
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>

№ варианта	Условие задачи
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 <math>n</math> 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 <math>n</math>. 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 <math>n</math> матрицах.</p>
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 <b>compr</b>, 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</p>



№ варианта	Условие задачи
	счисления в двоичную. С помощью этой функции перевести различные числа из десятичной системы счисления в двоичную.
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 <b>A</b>, и сколько таких слов. С помощью этой функции проверить несколько строк.</p>