

Федеральное агентство по образованию

Государственное образовательное учреждение
высшего профессионального образования
Санкт-Петербургский государственный технологический институт
(технический университет)

Кафедра систем автоматизированного проектирования и
управления

Чистякова Т.Б., Антипин Р.В., Новожилова И.В.

**ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ
ВЫСОКОГО УРОВНЯ.
КОНТРОЛЬНЫЕ РАБОТЫ**

Учебное пособие
для студентов заочной формы обучения

Санкт-Петербург
2008

Чистякова, Т.Б. Программирование на языках высокого уровня. Контрольные работы [Текст]: учебное пособие для студентов заочной формы обучения/ Т.Б. Чистякова, Р.В. Антипин, И.В. Новожилова. – СПб.: СПбГТИ(ТУ), 2008. – 104 с.

В учебное пособие включены задания для выполнения контрольных работ по изучению языка программирования высокого уровня C++ и созданию простых программ для объектов химической технологии в инструментальной среде разработки приложений на языке C++.

Учебное пособие предназначено для студентов заочной формы обучения по направлению подготовки 230100 «Информатика и вычислительная техника» и соответствует разделам рабочей программы курсов «Программирование на языках высокого уровня», «Технологии программирования», «Объектно-ориентированное программирование», «Лингвистическое и программное обеспечение САПР», «Алгоритмические языки и программирование».

Ил. 12, табл. 6, библиогр. назв. 6

Рецензенты:

Соколов Г.А., зав. кафедрой автоматизации процессов химической промышленности Санкт-Петербургского государственного технического института (технического университета), д-р техн. наук, проф.

Утверждено на заседании учебно-методической комиссии факультета информатики и управления

Рекомендовано к изданию РИСо СПбГТИ(ТУ)

© СПбГТИ(ТУ), 2008г.

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	4
ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КОНТРОЛЬНЫХ РАБОТ.....	5
КОНТРОЛЬНАЯ РАБОТА №1. ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ C++.....	6
КОНТРОЛЬНАЯ РАБОТА №2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ C++.....	13
КОНТРОЛЬНАЯ РАБОТА №3. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ.....	65
СПИСОК ЛИТЕРАТУРЫ.....	101

ВВЕДЕНИЕ

Высокоуровневые языки программирования разработаны для скорости и удобства использования программистом. Основная черта высокоуровневых языков – это абстракция, то есть введение смысловых конструкций, кратко описывающих такие структуры данных и операции над ними, описания которых на машинном коде (или другом низкоуровневом языке программирования) очень длинны и сложны для понимания. Наиболее распространёнными языками подобного типа являются C++, Visual Basic, Java, Python, Ruby, Perl, Pascal, PHP. Языкам высокого уровня свойственно умение работать с комплексными структурами данных. В большинство из них интегрирована поддержка строковых типов, объектов, операций файлового ввода-вывода и т. п.

В последнее десятилетие активное развитие получают языки программирования с объектно-ориентированными возможностями. Объектно-ориентированный подход к проектированию сложных программных систем стал безусловным стандартом, поскольку способствует лучшей управляемости проектом на всех стадиях разработки, тестирования и внедрения. Объектно-ориентированное программирование (ООП) – парадигма программирования, в которой основными концепциями являются понятия объектов и классов. ООП в настоящее время является абсолютным лидером в области прикладного программирования.

Одним из наиболее распространенных объектно-ориентированных языков высокого уровня является C++, созданный Бьерном Страуструпом в начале 1980-х годов. Язык C++ полностью унаследовал и расширил возможности языка Си, который используется в задачах системного программирования. К достоинствам языка C++ можно отнести кроссплатформенность; возможность работы на низком уровне с памятью, адресами, портами; возможность создания обобщенных алгоритмов для разных типов данных, их специализация, и вычисления на этапе компиляции, используя шаблоны.

В данном учебном пособии представлены задания на выполнение контрольных работ по изучению основных принципов объектно-ориентированного программирования; технологическим средствам разработки программного обеспечения; языка программирования C++; созданию простых программ для объектов химической технологии в инструментальной среде разработки приложений на языке C++.

ЗАДАНИЕ НА ВЫПОЛНЕНИЕ КОНТРОЛЬНЫХ РАБОТ

В учебном пособии составлены 3 контрольные работы. Студенту необходимо представить отчёт о выполненных контрольных работах в распечатанном виде и в электронном виде на любом носителе информации.

Отчёт должен включать: титульный лист, условие задачи, алгоритм решения (при необходимости), программу и результаты. Во время защиты контрольных работ студент должен подтвердить работоспособность программной реализации заданий. На титульном листе отчёта о выполнении контрольных работ необходимо указать фамилию, имя и отчество студента, номер учебной группы, номер контрольной работы, номер варианта.

Номер варианта соответствует номеру первой буквы фамилии студента согласно таблице 1.

Таблица – Распределение вариантов заданий

Первая буква фамилии студента	Номер варианта	Первая буква фамилии студента	Номер варианта
А	1	П	15
Б	2	Р	16
В	3	С	17
Г	4	Т	18
Д	5	У	19
Е, Ё	6	Ф	20
Ж	7	Х	21
З	8	Ц	22
И, Й	9	Ч	23
К	10	Ш, Щ	24
Л	11	Э	25
М	12	Ю	26
Н	13	Я	27
О	14	Пример решения	28

Приступая к выполнению контрольных работ, рекомендуется ознакомиться со следующими методическими материалами:

1) Чистякова, Т.Б. Программирование на языках высокого уровня. Базовый курс [Текст]: учебное пособие для студентов заочной формы обучения/ Т.Б. Чистякова, Р.В. Антипин, И.В. Новожилова. – СПб.: СПбГТИ(ТУ), 2008. – 227с.

2) Страуструп, Б. Язык программирования С++ / Бьерн Страуструп; Пер. с англ. С. Анисимова, М. Кононова; Под ред. Ф. Андреева, А. Ушакова. – Спец. изд. – М.: Бином; СПб.: Нев. диалект, 2001. – 1098 с.

КОНТРОЛЬНАЯ РАБОТА №1. ТЕОРЕТИЧЕСКИЕ ВОПРОСЫ ПО ПРОГРАММИРОВАНИЮ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ C++

Вариант №1

1) Что такое программирование и язык программирования? Перечислите основные языки программирования высокого уровня и их особенности.

2) Для чего применяется указатель на функцию, особенности использования?

3) Чем отличается конструктор копирования от обычного конструктора?

Вариант №2

1) Что такое парадигма программирования? Какие парадигмы программирования существуют?

2) В чем отличие передачи параметров по значению и по ссылке? В чем особенность использования указателя и ссылки в качестве параметров функции?

3) В каких случаях необходима перегрузка оператора присваивания?

Вариант №3

1) Что такое императивное программирование, какие парадигмы программирования к нему относятся?

2) Что такое исключения? Какие преимущества имеет механизм исключений относительно способа обработки ошибок в C++?

3) Какой порядок вызовов конструкторов при наследовании?

Вариант №4

1) Что такое декларативное программирование, какие парадигмы к нему относятся?

2) Что такое функция? Как объявляются функции?

3) Для чего используется множественное наследование? Как объявляется множественное наследование?

Вариант №5

1) Что такое массив? Как вычисляется размер массива? Какая связь массива с указателем?

2) Что такое класс (class), определение класса?

3) Для чего необходимы виртуальные базовые классы?

Вариант №6

- 1) Что такое структура (struct) с точки зрения структурного программирования? Какая существует связь структуры и класса (class)?
- 2) Для чего используются статические (static) и константные (const) поля классов?
- 3) Какой порядок вызовов деструкторов при наследовании?

Вариант №7

- 1) Что такое тип данных, какие существуют типы данных?
- 2) Для чего необходимы оператор выражения и составной оператор?
- 3) Чем абстрактный класс отличается от обычного класса?

Вариант №8

- 1) В чем отличия явной инициализации данных от неявной?
- 2) Для чего необходимы условный оператор и оператор безусловного перехода?
- 3) Какие существуют области видимости переменных?

Вариант №9

- 1) Что такое объединение (union), особенности его использования, вычисление размера объединения?
- 2) Для чего необходим оператор-переключатель?
- 3) Для чего используются чистые виртуальные функции?

Вариант №10

- 1) Какие скалярные типы данных существуют? Укажите размер и диапазон значений скалярных типов данных для вашей машины?
- 2) Что такое методы класса и как они объявляются?
- 3) Каким образом обеспечивается однократное включение заголовочных файлов?

Вариант №11

- 1) Что такое выражение, приоритеты, ассоциативность? Как можно изменить порядок вычисления выражений?
- 2) Что такое закрытые (private) члены класса, особенности их использования?
- 3) Зачем применяются виртуальные деструкторы?

Вариант №12

- 1) Какие существуют операции присваивания? Назовите их приоритет и ассоциативность?
- 2) Что такое открытые (public) члены класса, и в чём заключаются особенности их использования?
- 3) Какие классы используются для файлового ввода/вывода?

Вариант №13

- 1) Перечислите составные типы данных.
- 2) Какие существуют операторы цикла? Укажите особенности использования каждого оператора цикла?
- 3) Для чего используются виртуальные функции?

Вариант №14

- 1) Перечислите арифметические операции. Укажите их приоритет и ассоциативность.
- 2) Опишите оператор завершения и продолжения.
- 3) Как реализуется однократное наследование?

Вариант №15

- 1) Перечислите логические операции. Укажите их приоритет и ассоциативность.
- 2) Для чего используются и как объявляются шаблоны функций?
- 3) Какие основные функции используются для неформатированного доступа к файлу?

Вариант №16

- 1) Перечислите операции сравнения. Укажите их приоритет и ассоциативность.
- 2) Что такое защищенные (protected) члены класса? В чем заключаются особенности их использования?
- 3) Как реализуется открытое наследование? В чем его особенность?

Вариант №17

- 1) Перечислите битовые операции. Укажите их приоритет и ассоциативность.
- 2) По каким параметрам возможна перегрузка функций?
- 3) Как создаются многомерные динамические массивы с применением операций (new[], delete[])?

Вариант №18

- 1) Что такое указатель? В чем особенности его использования?
- 2) Укажите отличие встраиваемых функции от обычных.
- 3) Какие существуют классы памяти?

Вариант №19

- 1) Опишите операции для работы с динамической памятью (new, delete).
- 2) Что такое конструктор класса, для чего он используется?
- 3) Как реализуется защищенное наследование? Укажите его особенности.

Вариант №20

- 1) Опишите тернарную условную операцию, ее приоритет и ассоциативность.
- 2) Что означает ключевое слово this и для чего оно используется?
- 3) Как объявляется шаблон класса?

Вариант №21

- 1) Что такое ссылка? В чем заключаются особенности ее использования?
- 2) Опишите особенности функций с параметрами по умолчанию и функций с переменным количеством параметров?
- 3) Для чего используется ключевое слово void?

Вариант №22

- 1) В чем заключается концепция объектно-ориентированного программирования? Что такое полиморфизм?
- 2) Опишите рекурсивные функции, особенности их применения. Укажите альтернативный вариант реализации рекурсивных вычислений без использования рекурсивных функций.
- 3) Как реализуется закрытое наследование и в чём его особенности?

Вариант №23

- 1) Для чего используется приведение типов? Укажите особенности const_cast и reinterpret_cast.
- 2) Что такое деструктор класса и для чего он используется?
- 3) Как создаются битовые поля? Для чего используется безымянное битовое поле?

Вариант №24

- 1) Опишите перечисление (enum), особенности его использования. Какие преимущества имеет перечисление перед целочисленными константами?
- 2) Что такое дружественная функция и для чего она используется?
- 3) Для чего используются манипуляторы ввода/вывода?

Вариант №25

- 1) Опишите явное и неявное приведение типов. Укажите особенности static_cast, dynamic_cast.
- 2) Что такое дружественный класс, для чего он используется?
- 3) Для чего используется специализация шаблонов классов?

Вариант №26

- 1) В чем заключается концепция объектно-ориентированного программирования? Что такое инкапсуляция?
- 2) В чем особенность использования массивов в качестве параметров функции?
- 3) Для чего используются функции форматированного ввода/вывода: width, fill, precision?

Вариант №27

- 1) В чем заключается концепция объектно-ориентированного программирования? Что такое наследование?
- 2) Опишите операцию получения размера (sizeof), а также её использование применительно к скалярным типам данных, массивам и объектам?
- 3) Что такое перегрузка операторов? Какие операции нельзя перегружать?

Вариант №28

- 1) Что такое оператор языка программирования C++? Перечислите операторы, существующие в C++.
- 2) Для чего используется спецификатор volatile?
- 3) Для чего используется спецификатор mutable?

Пример выполнения контрольной работы № 1

Вариант № 28

1) Оператор языка программирования C++ – это мельчайшая независимая часть программы. Оператор соответствует предложению естественного языка, но завершается точкой с запятой (;). Выражение C++ (например, `ival + 5`) становится простым оператором, если после него поставить точку с запятой. Составной оператор – это последовательность простых операторов, заключенная в фигурные скобки. По умолчанию операторы выполняются в порядке записи. Операторы языка C++ делятся на три группы: операторы-выражения, получающиеся из произвольных выражений добавлением точки с запятой; пустые операторы и блоки; операторы, начинающиеся с ключевого слова.

Основные операторы C++:

<code>;</code>	Пустой оператор, при выполнении не имеет никакого эффекта;
<code>e;</code>	Оператор-выражение, вычисляет выражение <code>e</code> ;
<code>{ }</code>	Составной оператор или блок, выполняет операторы в блоке один за другим. Переменные, определенные в блоке, разрушаются по окончании блока;
<code>if(условие) оператор1 else оператор2</code>	Вычисляет <i>условие</i> и выполняет элемент <i>оператор1</i> , если <i>условие</i> имеет значение true, в противном случае выполняет <i>оператор2</i> ;
<code>while(условие) оператор</code>	Проверяет <i>условие</i> и выполняет <i>оператор</i> до тех пор, пока условие сохраняет значение true;
<code>do оператор while (условие);</code>	Выполняет <i>оператор</i> , а затем проверяет <i>условие</i> . Выполняет <i>оператор</i> до тех пор, пока <i>условие</i> не примет значение false;
<code>for(оператор1; условие; выражение) оператор2</code>	Выполняет <i>оператор1</i> один раз при входе в цикл, а затем проверяет <i>условие</i> . Если <i>условие</i> принимает значение true, выполняет <i>оператор2</i> , а затем вычисляет элемент <i>выражение</i> .
<code>switch (выражение) оператор</code>	Элемент <i>оператор</i> почти всегда представляет собой блок, который включает операторы с метками следующего вида <i>case значение</i> : При выполнении switch-оператора вычисляется <i>выражение</i> и управление передается той <i>case</i> -метки, значение которой совпадает с результатом вычисления элемента <i>выражение</i> .
<code>break;</code>	Передаёт управление <i>оператору</i> , непосредственно следующему за окончанием ближайшего оператора <code>while</code> , <code>for</code> , <code>do</code> или <code>switch</code> , который включает инструкцию <code>break</code> ;
<code>continue;</code>	Передаёт управление назад к началу следующей итерации в операторах <code>for</code> , <code>while</code> или <code>do</code> , которые включают оператор <code>continue</code> ;
<code>goto метка;</code>	Передаёт управление <i>оператору</i> , помеченному <i>меткой</i> . <i>Метка</i> должна находиться внутри текущей функции;
<code>try{ оператор } catch{ оператор }</code>	Выполняет код представленный элементом <i>оператор</i> , который может

catch (<i>параметр</i>) { <i>оператор</i> }	сгенерировать исключение (посредством оператора throw). Это исключение должно быть отработано оператором catch. Оператор catch обрабатывает исключения (генерируемые значения которых должны иметь подобный тип, как и тип элемента <i>параметр</i>) посредством выполнения <i>оператора</i> ;
throw <i>выражение</i> ;	Прекращает выполнение программы или передает управление catch-ветви текущего <i>оператора</i> try. Передает <i>выражение</i> , тип которого определяет, в какой именно catch-ветви будет обработано это исключение.

2) Спецификатор `volatile` отмечает, что в процессе выполнения программы значения объекта может изменяться в промежутке между явными обращениями к нему. Например, для многопоточного приложения, на объект может повлиять один из потоков. Поэтому компилятор не должен помещать его в регистровую память и не должен делать никаких предположений о постоянстве объекта в те моменты, когда в программе нет явных операций, изменяющих значения объекта.

3) Спецификатор `mutable` используется только для членов данных класса и позволяет этим членам данных быть модифицированными даже в случае, если они являются членами `const`-объектов. Спецификатор `mutable` необходим в тех случаях, когда строгое придерживание константности неудобно. Объект может оставаться логически константным ("logically const"), но при этом его физическая константность ("physically const") может быть нарушена.

КОНТРОЛЬНАЯ РАБОТА №2. ОСНОВЫ ПРОГРАММИРОВАНИЯ НА ЯЗЫКЕ ВЫСОКОГО УРОВНЯ C++

1) **Вычисление простого выражения.** Разработать программу для вычисления арифметического выражения и вывода полученного результата. Исходные данные ввести с клавиатуры.

Номер варианта	Выражение	Исходные данные
1	$a = \ln(y^{-\sqrt{ x }}) \cdot (\sin(x) + e^{(x+y)})$	x, y
2	$b = \sqrt{c(\sqrt{y} + x^2)} \cdot (\cos(x) - c - y)$	c, x, y
3	$c = \arctg(x) - \frac{3}{5}e^{xy} + 0.5 \frac{ x+y }{(x+y)^b}$	b, x, y
4	$d = \frac{e^{ x-y } \cdot \operatorname{tg}(z)}{\arctg(y) + \sqrt{x}} + \ln(x)$	x, y, z
5	$e = \frac{(\cos(x) - \sin(y))^3}{\sqrt{\operatorname{tg}(z)}} + \ln^2(x \cdot y \cdot z)$	x, y, z
6	$f = y^x + \sqrt{ x + e^y} - \frac{z^3 \cdot \sin^2(y)}{y + z^2 / (y - x)}$	x, y, z
7	$g = \frac{1 + \cos(x+y)}{ e^x - 2y / (1 + x^2 \cdot y^2) } \cdot x^3 + \arcsin(y)$	x, y
8	$h = 2 + \frac{x^2}{\sqrt{2}} + \frac{ y^3 }{\sqrt{2}} + \frac{z^4 \cdot (\ln(x) + 1) \cdot \sqrt{2}}{\sqrt{3}}$	x, y, z
9	$j = \left((1+y) \cdot \sqrt{\sin(3z)} - \frac{ y-x }{5} \right)^3$	x, y, z
10	$k = \ln \left (y - \sqrt{ x }) \cdot \left(x - \frac{y}{z + x^2 / 4} \right) \right $	x, y, z
11	$l = 0.5x^5 + 3 \cdot \cos(x+y) + e^{-0.1yz} - \sqrt{ x \cdot y }$	x, y, z
12	$m = \sqrt{ -3 \cdot \operatorname{tg}(x) \cdot \lg(x^4 + y) / e^{-x} + 1 }$	x, y
13	$n = \sqrt{e^x + \operatorname{tg}(x) + 1 \cdot (\lg(y) + \cos(x \cdot y) + \sqrt[3]{x})}$	x, y
14	$p = \frac{\lg(x) - e^{x+y}}{\sqrt{2+y^2} + x^3 - \ln(y) }$	x, y
15	$q = \sqrt{12x^4 - 3x^3 + 4x^2 - 5x + 6} - \lg^2(z)$	x, y, z
16	$r = \lg 1 - 2x + 3x^2 - 4x^3 + \sqrt{ x } / z$	x, y, z

Номер варианта	Выражение	Исходные данные
17	$s = \frac{2 \cdot \cos(x - 1/6)}{1/2 + \sin^2(y)} - \frac{1}{ x^2 / (y + x^3) }$	x, y
18	$t = \frac{x \cdot y \cdot z - y \cdot x + \sqrt{z} }{10^7 + \sqrt[4]{\lg(4)}}$	x, y, z
19	$u = \frac{(x + y - z)^3 - (x - y + x)^2 + \sqrt{ x + y + z }}{\log_2(\operatorname{tg}(2))}$	x, y, z
20	$w = \frac{(x/y) \cdot (z + x) \cdot e^{ x-y } + \ln(1+e)}{\sin^2(y) - (\sin(x) \cdot \sin(y))^2}$	x, y, z
21	$a = \frac{\sqrt{ x-1 } - \sqrt[3]{ y }}{1 + \frac{x^2}{2} + \frac{y^2}{4}} + \operatorname{tg}^2(xy)$	x, y
22	$b = \frac{3 + e^{y-1}}{1 + x^2 \cdot y - \operatorname{tg}(z) }$	x, y, z
23	$c = (1+y) \cdot \frac{x+y/(x^2+4)}{e^{-x-2} + 1/ x^2+4 }$	x, y
24	$d = \frac{2 \cos(x - \pi/6)}{1/2 + \sin^2(y)} + \frac{ y-x }{3}$	x, y
25	$e = \frac{1 + \sin^2(x+y)}{2 + x - 2x/(1+x^2y^2) } + x$	x, y
26	$f = \ln \left (y - \sqrt{ x }) \cdot \left(x - \frac{y}{z + x^2/4} \right) \right $	x, y, z
27	$h = x - \frac{y \cdot (\operatorname{arctg}(z) + e^{-(x+3)})}{x^2 + 2y^3}$	x, y, z
28	$k = 1 + y-x + \frac{(y-x)^2}{\sqrt{\operatorname{arctg}(xy)}}$	x, y

Рекомендуемые теоретические разделы для ознакомления: операция и выражение присваивания; арифметические операции.

2) **Вычисление условного выражения.** Разработать программу для вычисления выражения и вывода полученного результата. Соответствующие исходные данные ввести с клавиатуры.

Номер варианта	Выражение	Исходные данные
1	$a = \begin{cases} (x+y)^2 - \sqrt{x \cdot y}, & x \cdot y > 0 \\ (x+y)^2 + \sqrt{ x \cdot y }, & x \cdot y < 0 \\ (x+y)^2 + 1, & x \cdot y = 0 \end{cases}$	x, y
2	$b = \begin{cases} \ln(x/y) + (x^2 + y)^3, & x/y > 0 \\ \ln x/y + (x^2 + y)^3, & x/y < 0 \\ (x^2 + y)^3, & y \neq 0, x = 0 \\ 0, & y = 0 \end{cases}$	x, y
3	$c = \begin{cases} x^2 + y^2 + \sin(x), & x - y = 0 \\ (x - y)^2 + \cos(x), & x - y > 0 \\ (y - x)^2 + \operatorname{tg}(x), & x - y < 0 \end{cases}$	x, y
4	$d = \begin{cases} (x - y)^3 + \operatorname{arctg}(x), & x > y \\ (y - x)^3 + \operatorname{arctg}(x), & y > x \\ (y + x)^3 + 0.5, & y = x \end{cases}$	x, y
5	$e = \begin{cases} i \cdot \sqrt{a}, & i - \text{нечетное}, a > 0 \\ i / 2 \cdot \sqrt{ a }, & i - \text{четное}, a < 0 \\ \sqrt{ i \cdot a }, & \text{иначе} \end{cases}$	i, a
6	$g = \begin{cases} e^{ a - b }, & 0.5 < a \cdot b < 10 \\ \sqrt{ a+b }, & 0.1 < a \cdot b < 0.5 \\ 2 \cdot x^2, & \text{иначе} \end{cases}$	a, b, x
7	$h = \begin{cases} \operatorname{arctg}(x + y), & x < y \\ \operatorname{arctg}(x + y), & x > y \\ (x + y)^2, & x = y \end{cases}$	x, y
8	$j = \begin{cases} \sin(5 \cdot k + 3 \cdot m \cdot k), & k < m \\ \cos(5 \cdot k + 3 \cdot m \cdot k), & k > m \\ k^3, & k = m \end{cases}$	k, m

Номер варианта	Выражение	Исходные данные
9	$l = \begin{cases} 3 \cdot k^3 + 3 \cdot p^2, k > p \\ k - p , 3 < k < p \\ (k - p)^2, k = p \\ 0, \text{ иначе} \end{cases}$	k, p
10	$k = \begin{cases} \ln(f + q), f \cdot q > 10 \\ e^{f+q}, f \cdot q < 10 \\ f + a, f \cdot q = 10 \end{cases}$	f, q
11	$m = \begin{cases} z + \sqrt{x^2 + y^2} \cdot \sin(xy), x < -3 \\ 2x \cdot \sqrt{x^2 + 2 \cdot yz }, -3 \leq x \leq 1 \\ x^2 + 2y^2 + z , x > 1 \end{cases}$	x, y, z
12	$p = \begin{cases} \arcsin\left(\frac{x+y}{z}\right), \left \frac{x+y}{z}\right \leq 1 \\ 2x^2 + y + \operatorname{tg}(z), \left \frac{x+y}{z}\right > 1 \end{cases}$	x, y, z
13	$q = \begin{cases} xyz - y - z , xy < -1 \\ x^2 - y^2 + z, -1 \leq xy \leq 1 \\ \sqrt{x^2 + y^2 + 2 \sin(2z)}, xy > 1 \end{cases}$	x, y, z
14	$r = \begin{cases} \sqrt{x^2 + z^2} - \sqrt{x^2 + y^2}, x < -2 \\ 3xz - 4y , -2 \leq x \leq 2 \\ \sin\left(\frac{2x^2 + y^2}{z^2}\right), x > 2 \end{cases}$	x, y, z
15	$s = \begin{cases} 3x + 2y, x < 0 \\ x^2 + 4 \cdot \sqrt{y^2 + 1}, 0 \leq x \leq 1 \\ 3 \cdot \ln\left \arcsin\left(\frac{x}{y}\right)\right , x > 1 \end{cases}$	x, y

Номер варианта	Выражение	Исходные данные
16	$t = \begin{cases} \frac{3x^2 + 2y}{x^2 + y^2}, x < -2 \\ x - 3y , -2 \leq x \leq 2 \\ \ln \left \frac{x - y}{2 + 3y} \right , x > 2 \end{cases}$	x, y
17	$z = \begin{cases} \frac{a \cdot b}{\sqrt{a^2 + 2 \cdot b \cdot \sin(a + b) }}, a + b < -2 \\ 3 \cdot a \cdot b + 2 \cdot b^2 + c, -2 \leq a + b \leq 1 \\ \frac{1}{a \cdot b} + a \cdot c - \sqrt{ a - b }, a + b > 1 \end{cases}$	a, b, c
18	$a = \begin{cases} 2 \cdot x^2 - y \cdot x \cdot \sqrt{x + y} + y, x \cdot y > 0 \\ 3 \cdot y^2 + y \cdot x \cdot \sqrt{ x - y }, x \cdot y < 0 \\ 2 \cdot x \cdot y + 3, x \cdot y = 0 \end{cases}$	x, y
19	$b = \begin{cases} \sin(x + y) + 2 \cdot (x + y)^2, x - y > 0 \\ \sin(x - y) + (x - y)^3, x - y < 0 \\ x^2 + \sqrt{y} , y \neq 0, x = 0 \\ 0, y = 0 \end{cases}$	x, y
20	$c = \begin{cases} x^2 - y^2 + \sqrt{x/y}, -2 < x \cdot y < 0 \\ (x - y)^2 + \sqrt{x \cdot y + 2 \cdot x/y}, 0 < x \cdot y < 2 \\ 1/(x + y), \text{ иначе} \end{cases}$	x, y
21	$d = \begin{cases} (2 \cdot x - y)^2 + \sin(x), x > y \\ (y - 2 \cdot x)^2 + \cos(x), y > x \\ x^2 - 2 \cdot y, y = x \end{cases}$	x, y
22	$e = \begin{cases} x^{\sqrt{ a+b }}, 0.5 < a \cdot b < 10 \\ \sqrt{ a - b } + 2 \cdot x, 0.1 < a \cdot b < 0.5 \\ a \cdot b / x, \text{ иначе} \end{cases}$	a, b, x
23	$g = \begin{cases} \sqrt{(k^2 + m)} + 3 \cdot k , k < m \\ \sqrt{(k + m^3)} - 0.5 \cdot m - k , k > m \\ 2 \cdot m^2 - 0.5, k = m \end{cases}$	k, m

Номер варианта	Выражение	Исходные данные
24	$h = \begin{cases} \sqrt{k+p} + 2 \cdot p^2, k > p \\ 0.5 \cdot p + (p/k), k < p \\ 3 \cdot k - 2, k = p \end{cases}$	k, p
25	$k = \begin{cases} \sqrt{(x + y)} + 2 \cdot x - 3 \cdot y, x \cdot y > 5 \\ 2^{x \cdot y} - (x+y), x \cdot y < 5 \\ x - y, x \cdot y = 5 \end{cases}$	x, y
26	$m = \begin{cases} \sin(x) + 0.5 \cdot \sqrt{ x+yz }, y < -3 \\ 2y \cdot \sqrt{x^2+z}, -3 \leq y \leq 1 \\ 3 \cdot x^3 - 2 \cdot y^2 + z, y > 1 \end{cases}$	x, y, z
27	$n = \begin{cases} \sin(a \cdot b + c), \left \frac{a-b}{c} \right \leq 0.5 \\ 2a \cdot \sqrt{ b+c }, \left \frac{a-b}{c} \right > 0.5 \end{cases}$	a, b, c
28	$p = \begin{cases} \sqrt{ a \cdot b } + 2 \cdot c, a \cdot b < -2 \\ a^3 + b^2 - c^2, -2 \leq a \cdot b \leq 2 \\ a^c - b, a \cdot b > 2 \end{cases}$	a, b, c

Рекомендуемые теоретические разделы для ознакомления: операция и выражение присваивания; арифметические операции; условный оператор.

3) Табулирование функции. Вычислить и вывести на экран таблицу функции $y=f(x)$ в интервале $[a, b]$ с шагом h . Результаты представить в виде таблицы.

x	y
—	—
—	—
—	—
—	—
—	—
...	...

Таблицу выровнять с помощью функций форматирования `cout.width()`, `cout.precision()`.

Номер варианта	Функция $y=f(x)$
1	$y = \begin{cases} \frac{1}{(x+1)\sqrt{x^2+1}}, & x > -1 \\ \frac{-(\ln x)^3 + 3(\ln x)^2/2 + 3(\ln x)/2 + \frac{3}{4}}{2x^2}, & x \leq -1 \end{cases}$ $x \in [-3, 0], h = 0.1$
2	$y = \begin{cases} \frac{e^x(1+\sin x)}{1+\cos x}, & x < 0 \\ e^x \cdot \operatorname{tg} \frac{x}{2}, & x \geq 0 \end{cases}$ $x \in \left[-\frac{\pi}{4}, \frac{\pi}{4}\right], h = \frac{\pi}{20}$
3	$y = \begin{cases} \sin x \cdot \ln(\operatorname{tg} x), & x < 0 \\ 0, & x = 0 \\ \ln(\operatorname{tg} \frac{x}{2}) - \cos x \cdot \ln(\operatorname{tg} x), & x > 0 \end{cases}$ $x \in \left[-\frac{\pi}{8}, \frac{\pi}{8}\right], h = \frac{\pi}{40}$
4	$y = \begin{cases} (x \cdot \ln x)^2, & x > 0 \\ 0, & x = 0 \\ \frac{x^3}{27} (9 \ln^2 x - 6 \ln x + 2), & x < 0 \end{cases}$ $x \in [-0.6, 0.6], h = 0.12$
5	$y = \begin{cases} \arccos \sqrt{\frac{x}{1+x}}, & x \geq 0 \\ 1 - x^2 - 2 \sin^2 x, & x < 0 \end{cases}$ $x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], h = \frac{\pi}{20}$
6	$y = \begin{cases} \frac{2}{3}, & x = 0 \\ \frac{2}{3} \cdot \frac{\sin x}{x} + \frac{1}{x^2} \sin \frac{1}{x}, & \text{иначе} \end{cases}$ $x \in [0, 1], h = 0.1$
7	$y = \begin{cases} \sqrt{e^x - 1}, & x \geq 0 \\ -x^2, & x < 0 \end{cases}$ $x \in [-0.5, 0.5], h = 0.1$

Номер варианта	Функция $y=f(x)$
8	$y = \begin{cases} \operatorname{tg}^2 x, & x > 0 \\ x^2 \cdot \sin x, & x \leq 0 \end{cases}$ $x \in [-1, 1], h = 0.2$
9	$y = \begin{cases} \frac{\ln^2 x}{x}, & x > 0 \\ 0, & x = 0 \\ \frac{3}{5} \cdot \frac{\sin 2x}{x}, & x < 0 \end{cases}$ $x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], h = \frac{\pi}{20}$
10	$y = \begin{cases} x \operatorname{sh} 2x, & x > 0 \\ 0, & x = 0 \\ -x^3 \cdot e^x, & x < 0 \end{cases}$ $x \in [-0.5, 0.5], h = 0.1$
11	$y = \begin{cases} \frac{3}{7}, & x < 4 \\ \sqrt{9+x^2}, & \\ \frac{3}{35}(x-4)^3, & \text{иначе} \end{cases}$ $x \in [3, 5], h = 0.1$
12	$y = \begin{cases} 2e^{\sqrt{x}}, & x \geq 0 \\ 2x^3 \cdot \cos(x^2+1), & x < 0 \end{cases}$ $x \in [-1, 1], h = 0.2$
13	$y = \begin{cases} \frac{x}{x^4+3x^2+2}, & x \leq 0 \\ \frac{1}{2} \ln \frac{x^2+1}{x^2+2} - \frac{1}{2} \ln \frac{2x}{3}, & \text{иначе} \end{cases}$ $x \in [-2, 2], h = 0.2$
14	$y = \begin{cases} \frac{x^3}{3+x}, & x \leq -3 \\ 0, & x = 0 \\ 9x - 27 \ln(3+x), & \text{иначе} \end{cases}$ $x \in [-4, -2], h = 0.1$

Номер варианта	Функция $y=f(x)$
15	$y = \begin{cases} \left(\frac{\ln x}{3}\right)^3, & x > 0 \\ \frac{1}{\sqrt{2}} \ln \frac{1 + \sqrt{2}(x-1) + \sqrt{2x^2+1}}{2}, & \text{иначе} \end{cases}$ $x \in [-0.5, 0.5], h = 0.1$
16	$y = \begin{cases} \frac{1}{(3 \sin x + 2 \cos x)^2}, & x < -1 \\ \frac{3}{26} - \frac{3 \cos x - 2 \sin x}{13(2 \cos x + 3 \sin x)}, & x \geq -1 \end{cases}$ $x \in [-1.5, -0.5], h = 0.1$
17	$y = \begin{cases} 2 \sin^2 2x + 3 \cos^2 3x, & x < 0 \\ 3e^{\sin x + 2 \cos x}, & x \geq 0 \end{cases}$ $x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right], h = \frac{\pi}{10}$
18	$y = \begin{cases} x e^x (\sin x - \cos x) / 2, & x < -1 \\ (e^x \cos x - 1) / 2, & -1 \leq x \leq 0 \\ x^2 \sin 2x, & x > 0 \end{cases}$ $x \in [-2, 2], h = 0.2$
19	$y = \begin{cases} \frac{\ln^2 x}{x}, & x > 0 \\ 0, & x = 0 \\ \frac{1}{x^2 \sin 2x}, & x < 0 \end{cases}$ $x \in [-2, 2], h = 0.2$
20	$y = \begin{cases} x \operatorname{sh} x, & x > 0 \\ \frac{1 - \cos x}{1 + 2 \sin 3x}, & x \leq 0 \end{cases}$ $x \in [-1, 3], h = 0.2$
21	$y = \begin{cases} 0, & x = 0 \\ x \cdot \sin\left(\frac{1}{x}\right), & x > 0 \\ \frac{1}{x} \cdot \cos\left(\frac{2}{x}\right), & x < 0 \end{cases}$ $x \in \left[-\frac{2}{\pi}, \frac{2}{\pi}\right], h = \frac{1}{10\pi}$

Номер варианта	Функция $y=f(x)$
22	$y = \begin{cases} \frac{\sin(x-1)}{x-1}, & x > 1 \\ 1, & x = 1 \\ e^{-x} \cdot \cos(\pi x), & x < 1 \end{cases}$ $x \in [0, 2], h = 0.1$
23	$y = \begin{cases} \frac{\sin(2x)}{x}, & x > 0 \\ 2, & x = 0 \\ 2 \cdot e^{-x} \cdot \cos(x), & x < 0 \end{cases}$ $x \in [-1, 1], h = 0.1$
24	$y = \begin{cases} \sqrt{x+0.5}, & x = 0 \\ \ln(x+2), & x > 0 \\ \frac{1}{x} + 0.2 \cdot x, & x < 0 \end{cases}$ $x \in [-0.5, 1], h = 0.1$
25	$y = \begin{cases} x/2 - x, & x = 0.5 \\ \sin(x) - 2x + 0.5x, & x > 0.5 \\ \frac{\sqrt{x^2+2}}{x-1}, & x < 0.5 \end{cases}$ $x \in [0, 1], h = 0.1$
26	$y = \begin{cases} \frac{\pi x}{2}, & x > 0 \\ \pi/x + \sin(\pi/x), & x < 0 \\ 0, & x = 0 \end{cases}$ $x \in [-1, 2], h = 0.2$
27	$y = \begin{cases} e^{-2x}, & x > 0 \\ e^x + 2 \cdot x, & x < 0 \\ -2, & x = 0 \end{cases}$ $x \in [-1, 1], h = 0.1$

Номер варианта	Функция $y=f(x)$
28	$y = \begin{cases} \sin(x / 2), & x > 0.5 \\ 2x, & x = 0.5 \\ \cos\left(\frac{ 2x }{0.5\pi}\right), & x < 0.5 \end{cases}$ $x \in [0, 1.5], h = 0.1$

Рекомендуемые теоретические разделы для ознакомления: оператор цикла while; цикл с постусловием do-while; оператор for.

4) Работа с битами

Вариант №1

Ввести 8 символов. В символе с наибольшим кодом заменить 3-й бит нулем, а в символе с наименьшим кодом 4-й бит – единицей. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №2

Ввести последовательность из 8 символов. В каждом из символов в их двоичном представлении заменить: для нечетных (по порядку) символов 3-й бит единицей; для четных символов 4-й бит нулем. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №3

Ввести последовательность из 8 символов. В их двоичном представлении заменить: если младший бит 1, заменить его на 0; если младший бит 0, заменить его и 2-й бит единицами. Вывести исходную последовательность и ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №4

Ввести последовательность из 8 символов. Если символ – буква, то заменить в нем 3-й бит нулем, иначе – заменить 2-й бит единицей. Вывести исходную последовательность и ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №5

Ввести последовательность из 8 символов. Если символ – цифра, то заменить в нем 4-й бит единицей, иначе – 2-й бит нулем. Вывести исходную последовательность и ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №6

Ввести последовательность из 8 целых чисел. Если число четное, то заменить его младший байт нулями, если нечетное, то заменить в его младшем байте 3-й и 4-й бит единицами. Вывести исходную последовательность в десятичной и восьмеричной форме; преобразованную последовательность в десятичной и восьмеричной форме.

Вариант №7

Ввести последовательность из 8 целых чисел. В каждом втором числе заменить $(i-1)$ -й бит единицей, где i -номер члена последовательности. Вывести исходную последовательность в десятичной и восьмеричной формах; преобразованную последовательность в десятичной и восьмеричной формах.

Вариант №8

Ввести последовательность из 8 символов. Если символ – русская гласная буква, то заменить в нем младший бит единицей, иначе – заменить 2-й и 3-й биты нулями. Вывести исходную и преобразованную последовательности в символьной форме и в восьмеричных кодах.

Вариант №9

Ввести последовательность из 8 символов. Если символ – восьмеричная цифра, то заменить в нем бит, номер которого совпадает с этой цифрой, нулем; иначе – заменить младший бит единицей. Вывести исходную и преобразованную последовательности в символьной и восьмеричной формах.

Вариант №10

Ввести последовательность из 8 символов. Если код символа четный, то заменить в нем младший бит единицей, иначе – заменить два младших бита нулями. Вывести исходную и преобразованную последовательности в символьной и восьмеричной формах.

Вариант №11

Ввести последовательность из 8 символов. Если символ есть + – * / % , то заменить в нем четыре младших бита единицами, иначе – заменить 5-й бит нулем. Вывести исходную и преобразованную последовательности в символьной и восьмеричной формах.

Вариант №12

Ввести последовательность из 8 символов. Если символ – большая латинская буква, то заменить в нем 3-й бит нулем, иначе – заменить младший бит единицей. Вывести исходную и преобразованную последовательности в символьной и восьмеричной формах.

Вариант №13

Ввести последовательность из 8 целых чисел. В каждом нечетном числе заменить $(i-1)$ -й бит нулем (i -номер члена последовательности). Вывести исходную последовательность в десятичной и восьмеричной формах; преобразованную последовательность в десятичной и восьмеричной формах.

Вариант №14

Ввести последовательность из 8 символов. Если символ – латинская согласная буква, то заменить в нем младший бит единицей, иначе – заменить 3-й и 5-й биты нулями. Вывести исходную и преобразованную последовательности в символьной форме и в восьмеричных кодах.

Вариант №15

Реализовать алгоритм инвертирования n -разрядов целого числа без знака, начинающихся с p -ой позиции. Оставшиеся разряды остаются без изменения. Значения переменной, подлежащей преобразованию, а также значения n и p вводятся с клавиатуры. Результат вывести на экран в восьмеричном виде.

Вариант №16

Реализовать алгоритм, выполняющий зеркальное отображение битов значения целого числа без знака. Значение переменной, подлежащей преобразованию, вводится с клавиатуры. Результат вывести на экран в восьмеричном виде.

Вариант №17

Реализовать алгоритм поиска первой пары несовпадающих разрядов в двух переменных типа `unsigned`. Значения сравниваемых переменных вводятся с клавиатуры, результат выводится на экран.

Вариант №18

Ввести последовательность из 8 символов. Сравнить 5-й и 6-й биты каждого символа. Если они не равны, то сделать их равными младшему биту, иначе – старшему. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №19

Ввести 8 символов. В символе с наибольшим кодом заменить 5-й бит единицей, а в символе с наименьшим кодом 6-й бит – нулем. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №20

Ввести последовательность из 8 символов. В их двоичном представлении заменить: если старший бит 1, заменить его на 0; если старший бит 0, заменить его и младший бит единицами. Вывести исходную последовательность и ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №21

Реализовать алгоритм зеркального отображения тетрад битов значения целого числа без знака. Значение переменной, подлежащей преобразованию, вводится с клавиатуры. Результат вывести на экран в восьмеричном виде.

Вариант №22

Ввести последовательность из 8 целых чисел. Если код символа нечетный, то заменить в нем старший бит нулем, иначе – заменить два младших бита единицами. Вывести исходную последовательность в десятичной и восьмеричной формах; преобразованную последовательность в десятичной и восьмеричной формах.

Вариант №23

Ввести последовательность из 8 символов. В каждом из символов в их двоичном представлении заменить все четные биты единицами. Вывести

исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №24

Ввести последовательность из 8 символов. Если сумма единиц в представлении символа нечетная, то заменить 2 старших бита нулями, иначе – единицами. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №25

Ввести последовательность из 8 символов. Если сумма трех старших бит в символе равна единице, то заменить их единицами, иначе – нулями. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №26

Ввести последовательность из 8 символов. Сравнить их младший и старший биты. Если они равны, то заменить старший нулем, младший – единицей, иначе заменить старший бит единицей, младший – нулем. Вывести исходную последовательность, ее восьмеричные коды; преобразованную последовательность и ее восьмеричные коды.

Вариант №27

Ввести последовательность из 8 символов. Если символ – цифра, то заменить в нем 3 младших бита единицами, иначе – первый и последний нулями. Вывести исходную и преобразованную последовательности в символьной форме и в восьмеричных кодах.

Вариант №28

Ввести последовательность из 8 символов. Если символ – латинская гласная буква, то заменить в нем 2 младших бита нулем, иначе – 2-й и 4-й единицами. Вывести исходную и преобразованную последовательности в символьной форме и в восьмеричных кодах.

Рекомендуемый теоретический раздел для ознакомления: побитовые операции.

5) Рекуррентные последовательности

Вариант №1

Найти произведение

$$P = \prod_{i=1}^{30} Z_i$$

$$Z_1 = 0.45;$$

$$Z_2 = 0.17;$$

$$Z_k = 0.5 \sin 2Z_{k-1} - 0.9 \cos 3Z_{k-2}$$

Массивом не пользоваться.

Вариант №2

Вычислить сумму

$$S = \sum_{i=1}^{150} a_i$$

$$a_1 = 3.14;$$

$$a_2 = 1.57;$$

$$a_i = 2 \sin(ka_{i-1}) + 3 \cos(ka_{i-2})$$

$$k = \sqrt{a_1^2 + a_2^2}$$

Массивом не пользоваться.

Вариант №3

Задана последовательность $\{r_k\}$

$$r_1 = 2.2;$$

$$r_2 = 3.3;$$

$$r_3 = r_1 \cdot r_2;$$

$$r_k = (\sqrt{r_{k-1}^2 + 2r_{k-2}^2}) \cdot \sin(r_{k-1} \cdot r_{k-2})$$

Найти и напечатать наименьший элемент этой последовательности.

Массивом не пользоваться.

Вариант №4

Вычислить сумму

$$S = \sum_{i=1}^{100} x_i$$

$$x_1 = 0.327;$$

$$x_2 = 0.3;$$

$$x_i = i + 2 \sin(x_{i-1}) - x_{i-2}$$

Массивом не пользоваться.

Вариант №5

Найти минимальный член последовательности $\{U_i\}$ и его номер

$$U_1 = 0.5;$$

$$U_2 = 0.27;$$

$$U_i = 2 \sin(U_{i-1}) - 3 \cos(U_{i-2})$$

Массивом не пользоваться.

Вариант №6

Найти максимальный член последовательности $\{x_i\}$ и его номер

$$x_1 = 0.15;$$

$$x_2 = 0.19;$$

$$x_i = \cos(ix_{i-1}) - 3 \sin(x_{i-2})$$

Массивом не пользоваться.

Вариант №7

Вычислить произведение

$$P = \prod_{i=1}^{100} U_i,$$

где последовательность $\{U_i\}$ задается так:

$$U_1 = 0.4;$$

$$U_2 = 0.5;$$

$$U_3 = 0.93;$$

$$U_i = \sin(U_{i-1}) + \cos(U_{i-3}), i = 4, 5, \dots, 100.$$

Массивом не пользоваться.

Вариант №8

Напечатать значения очередной пары x_i, y_i последовательностей:

$$x_1 = 1;$$

$$y_1 = 0.14;$$

$$x_i = x_{i-1} + y_{i-1};$$

$$y_i = y_{i-1} + \sin(x_{i-1}) - 0.5x_i$$

Найти и напечатать $\max x_i, \min y_i$.

Массивом не пользоваться.

Вариант №9

Напечатать значения очередных пар x_k, y_k последовательностей. Найти $\min\{x_k\}$

$\{x_k\}, \{y_k\}$:

$$x_1 = 0.5; \quad x_2 = 0.6; \quad y_1 = y_2 = 0.67;$$

$$x_k = y_{k-1} + y_{k-2};$$

$$y_k = x_k^2 + x_{k-2} - y_{k-1} - 0.3x_k, \quad k=3, 4, \dots, 10.$$

Массивом не пользоваться.

Вариант №10

Напечатать значения очередных пар последовательностей $\{z_i\}, \{t_i\}$

$$z_1 = 0.14; \quad t_1 = 0.53;$$

$$z_i = \sin z_{i-1} + 2 \cdot t_{i-1};$$

$$t_i = t_{i-1} + z_{i-1} \cdot \sin z_{i-1}$$

$i = 2, 3, \dots, 30.$

Массивом не пользоваться.

Вариант №11

Напечатать значения очередных пар последовательностей и произведение

$$P = \prod_{i=1}^{100} y_i$$

$\{x_i\}, \{y_i\}$:

$$x_1 = 1; \quad x_2 = 1.5; \quad y_1 = -1; \quad y_2 = 0.6;$$

$$x_i = 5 \sin(y_{i-1}) + [y_{i-2}];$$

$$y_i = 2 x_{i-1} + 3 x_{i-2} + 5 y_{i-1}$$

$i = 3, 4, \dots, 100.$ $[]$ – обозначение целой части.

Массивом не пользоваться.

Вариант №12

Вычислить сумму

$$S = \sum_{i=1}^{100} U_i$$

$$U_1 = 2, \quad U_2 = 2.5, \quad U_3 = 0.15;$$

$$U_i = \cos[U_{i-1}] + |0.7 \sin U_{i-3}|$$

$i = 4, 5, \dots, 100.$ $[]$ – обозначение целой части.

Массивом не пользоваться.

Вариант №13

Вычислить произведение

$$P = \prod_{i=1}^{150} x_i$$

$$x_1 = 2.3; \quad x_2 = 1.5; \quad x_3 = -2.1;$$

$$x_i = \log_7 \left| \frac{1 + x_{i-1}}{2} \right| + 2 \sin(x_{i-3}),$$

$$i = 4, 5, \dots, 150.$$

Массивом не пользоваться.

Вариант №14

Вычислить сумму

$$S = \sum_{i=1}^{1000} r_i$$

$$r_1 = 0.27; \quad r_2 = -0.5;$$

$$r_k = (\sin r_{k-1}^2 + \cos r_{k-1}) \sqrt{r_{k-1}^2 + r_{k-2}^2}$$

Массивом не пользоваться.

Вариант №15

Найти произведение

$$P = \prod_{i=1}^{20} A_i$$

$$A_1 = 0.8$$

$$A_2 = 0.75$$

$$A_3 = 0.3$$

$$A_i = 0.25 \cdot \sin A_{i-1} + 0.35 \cdot \cos A_{i-3}, \quad i = 4, 5, \dots, 20.$$

Массивом не пользоваться.

Вариант №16

Вычислить сумму

$$S = \prod_{i=1}^{100} R_i$$

$$R_1 = 1.01;$$

$$R_2 = 0.75;$$

$$R_i = 0.25i + 2 \sin R_{i-1} - 0.35 \cos R_{i-2}$$

$$i = 3, \dots, 100.$$

Массивом не пользоваться.

Вариант №17

Задана последовательность $\{r_i\}$ следующим образом:

$$r_1 = 1.55;$$

$$r_2 = 1.77;$$

$$r_i = |r_{i-1} - r_{i-2}| + \cos r_{i-2}$$

Найти и напечатать наименьший элемент этой последовательности. Массивом не пользоваться.

Вариант №18

Вычислить сумму

$$S = \prod_{i=1}^{200} U_i$$

$$U_1 = 0.5;$$

$$U_2 = 1.2;$$

$$U_k = U_{k-1} - \cos(0.3k + U_{k-2}), \quad k = 3, 4, \dots, 200.$$

Массивом не пользоваться.

Вариант №19

Найти минимальный член последовательности $\{U_k\}$ и его номер:

$$U_1 = 2;$$

$$U_2 = 0.8;$$

$$U_k = \sin(1.5k + U_{k-1}) + 0.25U_{k-2}$$

$$i = 3, 4, \dots, 150.$$

Массивом не пользоваться.

Вариант №20

Найти максимальный член последовательности $\{U_k\}$ и его номер:

$$U_1 = 0.9;$$

$$U_2 = 1;$$

$$U_k = \sin k U_{k-1} - 0.77 \cos U_{k-2}$$

$$k = 3, 4, \dots, 100.$$

Массивом не пользоваться.

Вариант №21

Вычислить произведение

$$P = \prod_{i=1}^{50} W_i$$

$$W_1 = 0.3;$$

$$W_2 = 0.5;$$

$$W_i = i + \sin(W_{i-1}) - \cos(W_{i-2})$$

Массивом не пользоваться.

Вариант №22

Напечатать значения очередной пары x_i, y_i последовательностей:

$$x_1 = 0.7; x_2 = 0.65;$$

$$y_1 = 0.4; y_2 = 0.5;$$

$$x_i = x_{i-1} + x_{i-2} - 0.6y_{i-1};$$

$$y_i = y_{i-1} + 0.2x_{i-2}$$

$i = 3, 4, \dots, 15$. Найти и напечатать $\max x_i, \min y_i$.

Массивом не пользоваться.

Вариант №23

Напечатать значения очередных пар x_k, y_k последовательностей.

Найти

$$\sum_{k=1}^{50} x_k$$

$\{x_k\}, \{y_k\}$:

$$x_1 = 0.35; x_2 = 0.8;$$

$$y_1 = 0.7; y_2 = 0.3;$$

$$x_k = y_{k-1} - 0.7x_{k-2};$$

$$y_k = x_k + x_{k-1} - 0.5y_{k-1}$$

$k = 3, 4, \dots, 100$.

Массивом не пользоваться.

Вариант №24

Напечатать значения очередных пар последовательностей $\{u_i\}, \{v_i\}$

$$u_1 = 0.33;$$

$$v_1 = 0.77;$$

$$u_i = 0.8u_{i-1} + 1.2v_{i-1};$$

$$v_i = \sqrt{i} + u_i - 0.7v_{i-1}$$

$i = 2, 3, \dots, 35$.

Массивом не пользоваться.

Вариант №25

Напечатать значения очередных пар последовательностей и произведение

$$P = \prod_{i=1}^{50} U_i$$

$\{u_i\}, \{v_i\}$:

$$u_1 = -0.8; u_2 = 0.3;$$

$$v_1 = 0.23; v_2 = 0.44;$$

$$u_i = 0.3 v_{i-1} + 0.5 v_{i-2} - u_{i-1};$$

$$v_i = 0.4 \sqrt{(|u_i - v_{i-2}|)} + 0.6 v_{i-1}$$

, $i = 3, 4, \dots, 100$.

Массивом не пользоваться.

Вариант №26

Вычислить сумму

$$S = \sum_{i=1}^{150} A_i$$

$$A_1 = 1.5; A_2 = 2;$$

$$A_i = 2.5 \sin(A_{i-1} - A_{i-2}) + 3 \cos(A_{i-1} + A_{i-2}),$$

$i = 3, 4, \dots, 150$.

Массивом не пользоваться.

Вариант №27

Вычислить произведение

$$P = \prod_{i=1}^{100} Z_i$$

$$z_1 = 1.33; z_2 = -1.44; z_3 = 1.55;$$

$$z_i = \sin(i) + 2.2 \cos([z_{i-3} + z_{i-1}]) - 1.1 z_{i-2}$$

$i = 4, 5, \dots, 100$.

$[]$ – обозначение целой части.

Массивом не пользоваться.

Вариант №28

Вычислить сумму

$$S = \sum_{i=1}^{1000} r_i$$

$$r_1 = 1.23; r_2 = 0.65; r_3 = -0.25;$$

$$r_k = \frac{1.2 \sin(r_{k-1} + r_{k-2}) + 0.8 r_{k-3} - [0.55 r_{k-2}]}{1 - \cos(r_{k-1})}$$

$k = 4, 5, \dots, 1000.$

[] – обозначение целой части.

Массивом не пользоваться.

Рекомендуемые теоретические разделы для ознакомления: оператор цикла while; цикл с постусловием do-while.

б) Одномерные массивы

Вариант №1

Даны действительные числа a_1, \dots, a_{15} .

Получить

$$\bar{a} = \frac{1}{15} \sum_{i=1}^{15} a_i, \quad s = \sqrt{\frac{\sum_{i=1}^{15} (a_i - \bar{a})^2}{14}}$$

Вариант №2

Получены экспериментальные данные по влажности материала a_1, a_2, \dots, a_{50} в различные моменты времени. Вычислить среднее значение влажности и отклонение от среднего для каждого значения.

Вариант №3

Система из 25 материальных точек в пространстве задана с помощью последовательности действительных чисел $x_1, y_1, z_1, p_1; x_2, y_2, z_2, p_2; \dots; x_{25}, y_{25}, z_{25}, p_{25}$, где x_i, y_i, z_i – координаты i -й точки, а p_i – ее вес ($i=1, 2, \dots, 25$). Получить координаты центра тяжести системы, а также расстояние от центра тяжести до всех точек системы.

Вариант №4

Даны действительные числа a_1, \dots, a_{20} . Получить числа b_1, \dots, b_{20} , где b_i – среднее арифметическое всех членов последовательности a_1, \dots, a_{20} , кроме a_i ($i = 1, 2, \dots, 20$).

Вариант №5

Построить последовательность целых чисел a_1, \dots, a_{30} , где $a_1=1, a_2=1, a_i = a_{i-1} + a_{i-2}$ ($i = 3, \dots, 30$).

Вариант №6

Даны действительные числа a_1, \dots, a_{30} . Получить $a_{30}, a_{29}, \dots, a_1$.

Вариант №7

Даны натуральные числа n_1, \dots, n_{20} , действительные числа x_1, \dots, x_{20} .
Вычислить

$$\frac{n_1 x_1 + \dots + n_{20} x_{20}}{n_1 + \dots + n_{20}}$$

Вариант №8

Даны действительные числа $a_1, \dots, a_{20}, b_1, \dots, b_{20}$. Вычислить $(a_1+b_{20}) \cdot (a_2+b_{19}) \cdot \dots \cdot (a_{20}+b_1)$.

Вариант №9

Даны действительные числа $a_1, \dots, a_{28}, b_1, \dots, b_{28}$. Члены последовательности c_1, \dots, c_{29} связаны с членами данных последовательностей соотношениями

$$c_{29} = 0,$$

$$c_{29-i} = \frac{a_{29-i}}{b_{29-i} - c_{29-i+1}}.$$

$$i = 1, \dots, 28.$$

Получить c_1, \dots, c_{29} .

Вариант №10

Даны действительные числа a_1, \dots, a_{30} . Если в результате замены отрицательных членов последовательности a_1, \dots, a_{30} их квадратами члены будут образовывать неубывающую последовательность, то получить сумму членов исходной последовательности; в противном случае получить их произведение.

Вариант №11

Даны целые числа a_1, \dots, a_{30} . Все члены последовательности с четными номерами, предшествующие первому по порядку члену со значением $\max(a_1, \dots, a_{30})$, домножить на $\max(a_1, \dots, a_{30})$.

Вариант №12

Даны натуральное число m , действительные числа a_1, \dots, a_{30} (числа a_1, \dots, a_{30} попарно различны, $m \leq 30$). В последовательности a_1, \dots, a_{30} поменять местами наибольший член и член с номером m .

Вариант №13

Даны действительные числа $x_1, \dots, x_{101}, y_1, \dots, y_{101}$. Получить действительные

$$x'_1, \dots, x'_{101},$$

$$y'_1, \dots, y'_{101},$$

преобразовав для этого члены x_i, y_i по правилу: если они оба отрицательны, то каждый из них увеличить на 0.5; если отрицательно только одно число, то отрицательное число заменить его квадратом; если оба числа неотрицательны, то каждое из них заменить на среднее арифметическое исходных значений.

Вариант №14

Даны действительные числа a_1, \dots, a_{30} . Получить:

а) $\max(a_1 + a_{30}, a_2 + a_{29}, \dots, a_{15} + a_{16})$;

б) $\min(a_1 \cdot a_{16}, a_2 \cdot a_{17}, \dots, a_{15} \cdot a_{30})$.

Вариант №15

Даны действительные числа a_1, \dots, a_{20} . Преобразовать эту последовательность по правилу: большее из a_i и a_{10+i} ($i = 1, \dots, 10$) принять в качестве нового значения a_i , а меньшее – в качестве нового значения a_{10+i} .

Вариант №16

Даны целые числа a_1, \dots, a_{30} . Если в данной последовательности ни одно четное число не расположено после нечетного, то распечатать все отрицательные члены последовательности, иначе – все положительные. Порядок следования чисел в обоих случаях заменяется на обратный.

Вариант №17

Даны действительные числа r_1, \dots, r_{17} , среди которых заведомо есть как отрицательные, так и неотрицательные. Получить $x_1 y_1 + \dots + x_s y_s$, где x_1, \dots, x_p – отрицательные члены последовательности r_1, \dots, r_{17} , взятые в порядке их следования, y_1, \dots, y_q – неотрицательные члены, взятые в обратном порядке, $s = \min(p, q)$.

Вариант №18

Даны целые числа a_1, \dots, a_{20} . Наименьший член этой последовательности заменить целой частью среднего арифметического всех членов, остальные члены оставить без изменения. Если в последовательности несколько членов со значением $\min(a_1, \dots, a_{20})$, то заменить последний по порядку.

Вариант №19

Даны действительные числа a_1, \dots, a_{20} (все числа попарно различны). Поменять в этой последовательности местами:

- а) наибольший и наименьший члены;
- б) наибольший и последний члены.

Вариант №20

Даны целые числа a_1, \dots, a_{100} . Получить новую последовательность из 100 целых чисел, заменяя a_i нулями, если $|a_i|$ не равно $\max(a_1, \dots, a_{100})$, и заменяя a_i единицей в противном случае ($i = 1, \dots, 100$).

Вариант №21

Даны целые числа $a_1, \dots, a_{25}, b_1, \dots, b_{25}$. Преобразовать последовательность b_1, \dots, b_{25} по правилу: если $a_i \leq 0$, то b_i увеличить в 10 раз, иначе b_i заменить нулем ($i=1, \dots, 25$).

Вариант №22

Даны действительные числа a_1, \dots, a_{26} . Требуется домножить все члены этой последовательности на квадрат ее наименьшего члена, если $a_1 \geq 0$, и на квадрат ее наибольшего члена, если $a_1 < 0$.

Вариант №23

Даны натуральное число n , действительные числа a_1, \dots, a_{30} . Получить b_1, \dots, b_{10} , где b_i равно сумме тех членов последовательности a_1, \dots, a_{30} , которые принадлежат полуинтервалу $(i - 1, i]$ ($i = 1, \dots, 10$). Если полуинтервал не содержит членов последовательности, то соответствующее b_i положить равным нулю.

Вариант №24

В массиве $A[30]$ найти наибольшее число подряд идущих одинаковых элементов (например $\{1,5,3,6,6,6,6,6,3,4,4,5,5,5\} = 5$).

Вариант №25

В массиве $B[30]$ найти и вывести значение наиболее часто встречающегося элемента.

Вариант №26

Интервал между минимальным и максимальным значениями элементов массива $C[25]$ разбить пополам и относительно этого значения разбить массив на две части, каждую из которых записать в новый массив (части не сортировать).

Вариант №27

Найти в массиве $D[30]$ элемент, наиболее близкий к среднему арифметическому его элементов.

Вариант №28

Задана последовательность вещественных чисел b_1, \dots, b_{30} . Сформировать одномерный массив A такой, что:

$$a_i = 1 / (b_i - b_{i-1}), \quad i = 2, 3, \dots, 30;$$

$$a_1 = 1 / (b_1 - b_{30}).$$

Последовательность b_1, \dots, b_{30} ввести с клавиатуры.

Рекомендуемый теоретический раздел для ознакомления: указатели и массивы.

7) Двумерные массивы

Вариант №1

Задана матрица $Z(5,4)$. Найти в каждой строке, если там есть отрицательный элемент, среднее арифметическое всех элементов, исключая нулевые и записать эти значения в массив B . Вывести исходную матрицу Z и массив B .

Вариант №2

В матрице $X(4,5)$ в каждой строке найти максимальный элемент и заменить им первый элемент строки. Предварительно первый элемент строки вывести в массив, если он не равен нулю. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №3

В матрице $Z(4,5)$ сдвинуть все элементы влево (циклически) в тех строках, которые начинаются с положительного элемента. Сдвинутые элементы вывести в массив. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №4

В матрице $Z(5,5)$ найти сумму элементов в тех строках, в которых элемент на главной диагонали равен нулю. Этой суммой заменить элемент на главной диагонали. Вывести исходную и преобразованную матрицы.

Вариант №5

Каждую строку матрицы $Z(5,4)$ преобразовать по правилу: если максимальный элемент не первый, то поменять его местом с первым. Вывести количество таких строк, исходную и преобразованную матрицы.

Вариант №6

В каждой строке матрицы $Z(5,6)$ сдвинуть все элементы вправо на один разряд (циклически). Если при этом в последнем столбце оказался нуль, то заменить его числом P , введенным с клавиатуры. Элементы последнего столбца вывести в массив. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №7

Задана матрица $Z(4,5)$. В каждой строке найти произведение элементов, расположенных до первого нулевого и их количество. Этим количеством заменить первый нулевой, а произведение записать в массив B . Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №8

В матрице $Z(4,5)$ переписать в обратном порядке элементы в тех строках, которые начинаются с нуля. Все отрицательные элементы вывести в массив B . Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №9

Дана действительная квадратная матрица порядка 7. Если в i -ой строке матрицы элемент, принадлежащий главной диагонали, отрицателен, то заменить этот элемент суммой элементов i -той строки, предшествующих первому отрицательному элементу; в противном случае – суммой последних

элементов i -той строки, начиная с первого отрицательного элемента. Элементы главной диагонали (неизмененные) вывести в массив. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №10

В матрице $Z(5,6)$ первый отрицательный элемент каждого столбца заменить суммой оставшихся. Отрицательные элементы до замены вывести в массив B . Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №11

Задана матрица $Z(5,6)$. Выбрать строку с наибольшей суммой элементов и вывести элементы этой строки в массив, затем каждый отрицательный элемент умножить на номер столбца. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №12

Дана действительная квадратная матрица порядка 7. Вычислить сумму тех ее элементов, расположенных на главной диагонали и выше нее, которые превосходят по величине все элементы, расположенные ниже главной диагонали. Заменить этой суммой элемент на главной диагонали соответствующего столбца. Если на главной диагонали и выше нее нет элементов с указанным свойством, то элемент на главной диагонали оставить без изменения. Элементы главной диагонали (неизмененные) вывести в массив. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №13

В матрице $Z(5,6)$ поделить элементы нечетных столбцов на свой номер, если в остатке не нуль, то заменить этот элемент полученным значением. Вывести количество таких элементов, исходную и преобразованную матрицы.

Вариант №14

В матрице $Z(5,5)$ найти номер строки, в которой содержится наибольшее количество отрицательных элементов. Количеством элементов (отрицательных) в каждой строке заменить соответствующий элемент главной диагонали. Вывести исходную и преобразованную матрицы.

Вариант №15

Даны две действительные квадратные матрицы порядка 6. Получить новую матрицу умножением элементов каждой строки первой матрицы на наибольшее из значений элементов соответствующей строки второй матрицы. Вывести исходные и полученную матрицы.

Вариант №16

В матрице $Z(5,4)$ в каждой нечетной строке выполнить следующее преобразование: сложить все отрицательные элементы и заменить этой суммой элемент первого столбца, а все элементы первого столбца вывести в вектор. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №17

В матрице $Z(4,6)$ поменять местами первый элемент в каждой строке с последним, второй – с предпоследним и т.д., если ни один из этих элементов не равен нулю. Вывести исходную и преобразованную матрицы.

Вариант №18

Даны две действительные квадратные матрицы порядка 6. Получить новую матрицу прибавлением к элементам каждого столбца первой матрицы произведения элементов соответствующих строк второй матрицы. Вывести исходные и полученную матрицы.

Вариант №19

В матрице $Z(5,5)$ найти сумму элементов в тех строках, в которых элемент на главной диагонали равен нулю. Найти и вывести количество таких строк. Элемент на главной диагонали, равный нулю, заменить найденной суммой элементов. Вывести исходную и преобразованную матрицы.

Вариант №20

Дана матрица $Z(5,5)$. В каждой четной строке просуммировать отрицательные элементы и заменить этой суммой первый элемент строки, в каждой нечетной строке найти количество отрицательных элементов. Вывести исходную и преобразованную матрицы.

Вариант №21

Дана целочисленная квадратная матрица порядка 6. Найти матрицу, получающуюся из данной перестановкой строк – первой с последней, второй – с предпоследней и т.д. Перестановка осуществляется при условии, что

элемент главной диагонали обеих строк не равен нулю. Вывести исходную и преобразованную матрицы.

Вариант №22

Задана матрица $Z(5,4)$. В каждой строке найти первый отрицательный элемент и заменить его произведением этого элемента на его номер. Первоначальное значение запомнить в массиве. Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №23

Дана матрица $Z(5,4)$. Найти номера строк, которые содержат не более двух отрицательных элементов. Эти элементы возвести в квадрат. Вывести номера строк, исходную и полученную матрицы.

Вариант №24

В матрице $Z(4,6)$ в каждой строке элементы, стоящие на нечетных местах, заменить суммой, на четных – произведением соответствующей пары. Элементы главной диагонали вывести в массив. Распечатать результат в виде исходной и преобразованной матрицы, полученного массива.

Вариант №25

В массиве $B[30]$ найти и вывести значение наиболее часто встречающегося элемента.

Вариант №26

Задана матрица $Z(6,6)$. Элемент главной диагонали на каждой строке заменить суммой элементов, расположенных за ним (если элемент на главной диагонали не равен нулю). Элементы главной диагонали вывести в массив B . Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №27

Дана матрица $Z(4,5)$. Составить новую, заменив нулями элементы строки и столбца, где находится максимальный элемент. Элементы из строки и столбца переписать в вектор C . Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №28

Дана матрица $Z(4,6)$. Определить и вывести в массив B все элементы, которые в своей строке больше предыдущего и меньше последующего. Вывести исходную матрицу и полученный массив.

Рекомендуемый теоретический раздел для ознакомления: многомерные массивы.

8) Динамическое распределение памяти

Вариант №1

Разработать программу перемножения двух матриц A , B размерности $n \times l$. Все матрицы размещаются в оперативной памяти динамически с помощью операции `new`, а значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

Вариант №2

Разработать программу нормирования матрицы размерности $m \times n$, которое заключается в том, что каждый элемент в этой матрице вычисляется на основании исходной матрицы, как отношение суммы всех других элементов в его строке к сумме всех других элементов в его столбце. Матрица должна размещаться в оперативной памяти динамически с помощью операции `new`, значения m и n вводятся с клавиатуры по запросу. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Вариант №3

Разработать программу, которая в матрице размерности $n \times n$ меняет местами строку, содержащую элемент с минимальным значением со столбцом, содержащим элемент с максимальным значением. Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Вариант №4

Дана действительная квадратная матрица порядка n , все элементы которой различны. Найти наибольший элемент среди стоящих на главной и побочной диагоналях и поменять его местами с элементом, стоящим на пересечении этих диагоналей. Матрица размещается в памяти динамически с

помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Вариант №5

Построить квадратную матрицу порядка $2n$:

$$\begin{matrix} n \\ \left[\begin{array}{cc} 11 \dots 1 & 22 \dots 2 \\ 11 \dots 1 & 22 \dots 2 \\ \dots & \dots \\ 11 \dots 1 & 22 \dots 2 \\ 33 \dots 3 & 44 \dots 4 \\ 33 \dots 3 & \dots 4 \\ \dots & \dots \\ 33 \dots 3 & 44 \dots \end{array} \right] \\ n \end{matrix}$$

$\underbrace{\hspace{10em}}_n \quad \underbrace{\hspace{10em}}_n$

Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №6

Дано действительное число x . Получить квадратную матрицу порядка $n < 10$:

$$\begin{bmatrix} 1 & x & \dots & x^{n-2} & x^{n-1} \\ x & 0 & \dots & 0 & x^{n-2} \\ \vdots & & & & \\ x^{n-2} & 0 & \dots & 0 & x \\ x^{n-1} & x^{n-2} & \dots & x & 1 \end{bmatrix}$$

Матрица размещается в памяти динамически с помощью операции new, значения x и n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №7

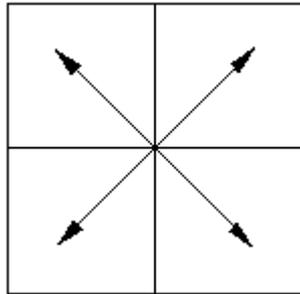
Даны действительные числа a_1, \dots, a_n . Получить квадратную матрицу порядка n :

$$\begin{bmatrix} a_1 & a_2 & a_3 & \dots & a_{n-2} & a_{n-1} & a_n \\ a_2 & a_3 & a_4 & \dots & a_{n-1} & a_n & a_1 \\ a_3 & a_4 & a_5 & \dots & a_n & a_1 & a_2 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ a_n & a_1 & a_2 & \dots & a_{n-3} & a_{n-2} & a_{n-1} \end{bmatrix}$$

Матрица размещается в памяти динамически с помощью операции new, значение n и числа a_1, \dots, a_n вводятся по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №8

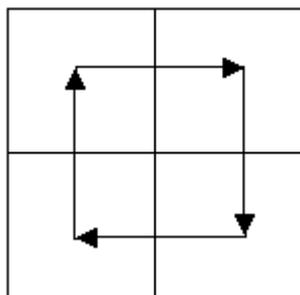
Дана действительная квадратная матрица порядка $2n$. Получить новую матрицу, переставляя ее блоки размера $n \times n$:



Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Вариант №9

Дана действительная квадратная матрица порядка $2n$. Получить новую матрицу, переставляя ее блоки размера $n \times n$:



Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы

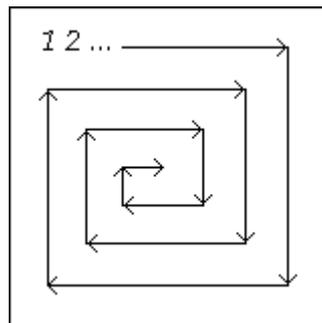
матрицы. Матрицы размещаются в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

Вариант №14

Даны две действительные квадратные матрицы порядка n . Получить новую матрицу прибавлением к элементам каждого столбца первой матрицы произведения элементов соответствующих строк второй матрицы. Матрицы размещаются в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходные и результирующую матрицы.

Вариант №15

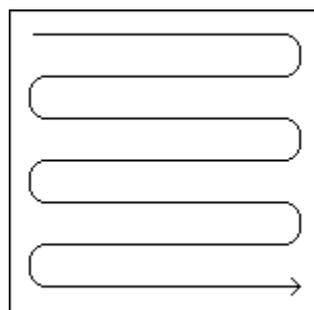
Получить целочисленную квадратную матрицу порядка $n < 8$, элементами которой являются числа $1, 2, \dots, n^2$, расположенные в ней по спирали:



Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №16

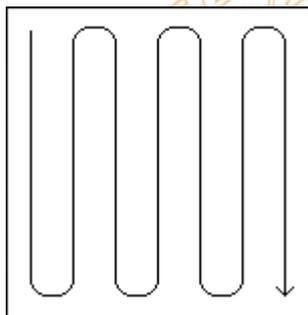
Даны действительные числа a_1, a_2, \dots, a_{n^2} . Получить действительную квадратную матрицу порядка $n < 8$, элементами которой являются числа a_1, a_2, \dots, a_{n^2} , расположенные в ней по схеме:



Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №17

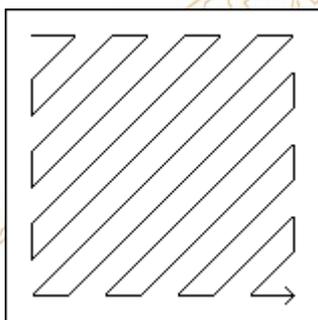
Даны действительные числа a_1, a_2, \dots, a_{n^2} . Получить действительную квадратную матрицу порядка $n < 8$, элементами которой являются числа a_1, a_2, \dots, a_{n^2} , расположенные в ней по схеме:



Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №18

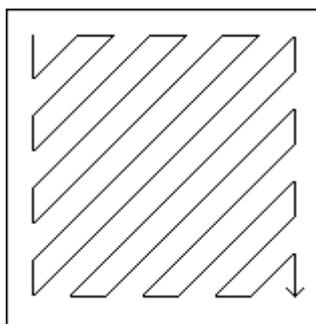
Даны действительные числа a_1, a_2, \dots, a_{n^2} . Получить действительную квадратную матрицу порядка $n < 8$, элементами которой являются числа a_1, a_2, \dots, a_{n^2} , расположенные в ней по схеме:



Матрица размещается в памяти динамически с помощью операции new, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №19

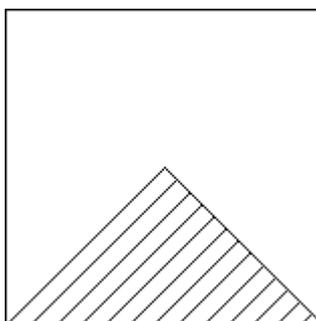
Даны действительные числа a_1, a_2, \dots, a_{n^2} . Получить действительную квадратную матрицу порядка $n < 8$, элементами которой являются числа a_1, a_2, \dots, a_{n^2} , расположенные в ней по схеме:



Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести полученную матрицу.

Вариант №20

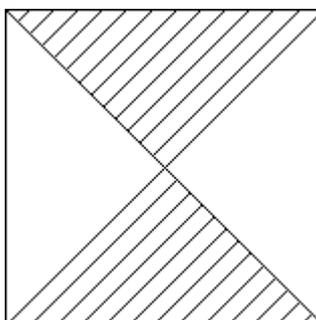
Дана действительная квадратная матрица порядка n . Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

Вариант №21

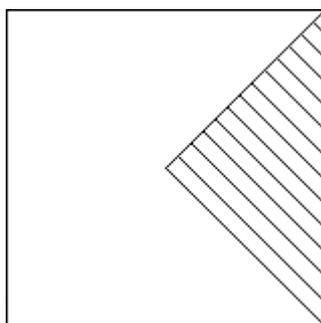
Дана действительная квадратная матрица порядка n . Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

Вариант №22

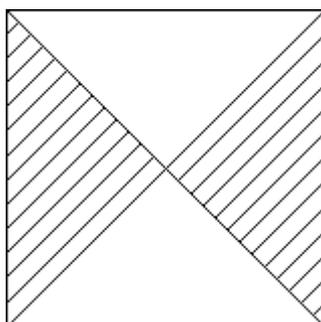
Дана действительная квадратная матрица порядка n . Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

Вариант №23

Дана действительная квадратная матрица порядка n . Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

Вариант №24

Дана целочисленная матрица размера $m \times n$. Найти матрицу, получающуюся из данной перестановкой столбцов – первого с последним, второго с предпоследним и т.д. Матрица размещается в памяти динамически с помощью операции `new`, значения m и n вводятся по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Вариант №25

Дана целочисленная матрица размера $m \times n$. Найти матрицу, получающуюся из данной перестановкой строк – первой с последней, второй

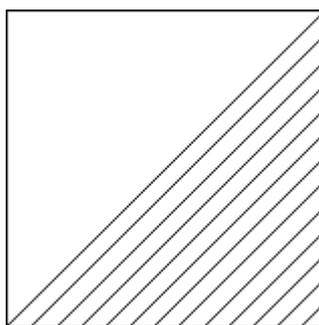
с предпоследней и т.д. Матрица размещается в памяти динамически с помощью операции `new`, значения m и n вводятся по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Вариант №26

Задана матрица $Z(6,6)$. Элемент главной диагонали на каждой строке заменить суммой элементов, расположенных за ним (если элемент на главной диагонали не равен нулю). Элементы главной диагонали вывести в массив B . Вывести исходную и преобразованную матрицы, полученный массив.

Вариант №27

Дана действительная квадратная матрица порядка n . Найти и вывести наибольшее из значений элементов, расположенных в заштрихованной части матрицы:



Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную матрицу.

Вариант №28

Дана действительная квадратная матрица порядка n , все элементы которой различны. В этой матрице в каждой строке элементы, стоящие на нечетных местах, заменить суммой, на четных – произведением соответствующей пары. Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

Рекомендуемые теоретические разделы для ознакомления: операция выделения памяти `new`; операция освобождения памяти `delete`.

9) Работа с символьными данными

Вариант №1

Прочитать из файла строку символов. Преобразовать данную строку, удалив из нее каждую пару символов >> и повторив (вставив еще раз) каждую пару символов <<. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №2

Прочитать из файла строку символов. Расположить символы в этой строке так, чтобы сначала шли символы, больше введенного с клавиатуры, а затем меньше. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №3

Прочитать из файла строку символов. Данная строка состоит из нулей, единиц и пробелов. Группы нулей и единиц, разделенные пробелами (одним или несколькими) и не содержащие пробелов внутри себя, будем называть словами. Требуется подсчитать и вывести количество слов в данной строке. Рассматривая слова как числа, определить и вывести количество слов, делящихся на 5 без остатка. Новую строку не создавать. Вывести исходную строку.

Вариант №4

Прочитать из файла строку символов. Написать функцию, которая изменяет данную строку. Если она находит в строке число, то она заменяет его на соответствующую по счету букву из этой строки (например, "aaabblbbcc5cc" – "aaabbabbccbcc"). Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №5

Прочитать из файла строку символов. Преобразовать данную строку, содержащую выражение на Си с операциями (=, ==, !=, a+=", a-=), в строку, содержащую эти же операции с синтаксисом языка Паскаль (:=, =, <>, a=a+, a=a-). Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №13

Прочитать из файла строку символов. Выяснить, имеется ли в этой строке пара одинаковых соседних букв и вывести их количество. Вывести исходную строку.

Вариант №14

Прочитать из файла строку символов. Выяснить, имеется ли в этой строке пара соседствующих букв *он* или *но* и вывести их количество. Вывести исходную строку.

Вариант №15

Прочитать из файла строку символов. Выяснить, верно ли, что среди символов имеются все буквы, входящие в слово *диаметр*. Вывести исходную строку и результат проверки.

Вариант №16

Прочитать из файла строку символов. Изменить данную строку так, чтобы каждый символ не цифра был заменен цифрой. Цифры при замене брать последовательно, начиная с 9 и заканчивая 0. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №17

Прочитать из файла строку символов. Если в ней встречается символ-цифра, то заменить его символом *!*, причем количество замен должно соответствовать значению цифры. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №18

Прочитать из файла строку символов. Изменить ее, заменив две подряд встречающиеся буквы *CH* символом *S*. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №19

Прочитать из файла строку символов. В этой строке подсчитать, сколько символов стоят перед символом *?*, все последующие заменить символами *!*. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №20

Прочитать из файла строку символов. В данной строке каждый символ ? удвоить. Каждый символ, стоящий перед !, заменить символом, введенным с клавиатуры. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №21

Прочитать из файла строку символов. Зашифровать текст из этой строки так, чтобы каждый символ-цифра был заменен его порядковым номером. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №22

Прочитать из файла строку символов. В данной строке каждый символ-цифру заменить символом, введенным с клавиатуры, если символы равны, то заменить его следующим по алфавиту. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №23

Прочитать из файлов две строки символов. В первой оставить только те символы, которых нет во второй строке. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №24

Прочитать из файла строку символов. Исключить из данной строки все символы, идущие подряд, равные символу, введенному с клавиатуры. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №25

Прочитать из файла строку символов. Поместить в ее начале все латинские заглавные буквы, которые в ней присутствуют. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №26

Прочитать из файла строку символов. Изменить данную строку так, чтобы каждый символ не цифра был заменен цифрой. Цифры при замене брать последовательно, начиная с нуля. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №27

Прочитать из файла строку символов. Заменить в ней символ, равный символу, введенному с клавиатуры, символом ?, а все следующие – порядковыми номерами предыдущего символа. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №28

Прочитать из файла строку символов. Заменить в ней символ, равный символу, введенному с клавиатуры, его порядковым номером. Новую строку не создавать. Вывести исходную и преобразованную строки.

Рекомендуемый теоретический раздел для ознакомления: символьные массивы и строки.

Пример выполнения контрольной работы № 2

Для написания программ использовалась IDE Visual Studio 2005, программы представляют консольные приложения.

Вариант № 28

1) Расчет арифметического выражение:

$$k = 1 + |y - x| + \frac{(y - x)^2}{\sqrt{\arctg(xy)}}$$

```
#include <iostream>
#include <math.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    double x, y;
    cout << "Введите положительные значения параметров x, y" << endl;
    cout << "x = ";
    cin >> x;
    cout << "y = ";
    cin >> y;
    if (x < 0 || y < 0)
    {
        cerr << "Параметры x, y должны быть положительными!";
        return 1;
    }
    double k = 1 + fabs(y - x) + pow(y - x, 2) / (sqrt(atan(x * y)));
    cout << "Результат вычисления k = " << k;
    return 0;
}
```

x	y	Результат
1	1	Результат вычисления k = 1
5	7	Результат вычисления k = 6,20779
-1	3	Параметры x, y должны быть положительными!

2) Вычисление условного выражения:

$$p = \begin{cases} \sqrt{|a \cdot b|} + 2 \cdot c, a \cdot b < -2 \\ a^3 + b^2 - c^2, -2 \leq a \cdot b \leq 2 \\ a^c - b, a \cdot b > 2 \end{cases}$$

```
#include <iostream>
#include <math.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    double a, b, c, p;
    cout << "Введите значения параметров a, b, c" << endl;
    cout << "a = ";
    cin >> a;
    cout << "b = ";
    cin >> b;
    cout << "c = ";
    cin >> c;
    if (a * b > 2.0)
    {
        p = pow(a, c) - b;
    }
    else if (a * b < -2.0)
    {
        p = sqrt(fabs(a * b)) + 2 * c;
    }
    else
    {
        p = a * a * a + b * b - c * c;
    }
    cout << "Результат условного выражения p = " << p;
    return 0;
}
```

a	b	c	p
2	1	4	-7
-1	5	1	4.23607
3	2	-1	-1.6667

3) Вычислить функцию на интервале и вывести таблицу:

$$y = \begin{cases} \sin(x/2), x > 0.5 \\ 2x, x = 0.5 \\ \cos\left(\frac{|2x|}{0.5\pi}\right), x < 0.5 \end{cases}$$

$$x \in [0, 1.5], h = 0.1$$

```
#include <iostream>
#define _USE_MATH_DEFINES
#include <math.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    double y;
```

```

cout << "    x    |    y    " << endl;
cout << "-----|-----" << left << endl;
for(double x = 0.0; x < 1.6; x += 0.1)
{
    if (x < 0.5)
    {
        y = cos(fabs(2 * x)/(0.5 * M_PI));
    }
    else if (x > 0.5)
    {
        y = sin(x / 2.0);
    }
    else
    {
        y = 2.0 * x;
    }
    cout.width(10);
    cout.precision(10);
    cout << x << "|" << y << endl;
}
return 0;
}

```

x	y
0	1
0.1	0.9919052498
0.2	0.9677520491
0.3	0.9279314261
0.4	0.8730880569
0.5	1
0.6	0.2955202067
0.7	0.3428978075
0.8	0.3894183423
0.9	0.4349655341
1	0.4794255386
1.1	0.5226872289
1.2	0.5646424734
1.3	0.6051864057
1.4	0.6442176872
1.5	0.68163876

4) Ввести последовательность из 8 символов. Если символ – латинская гласная буква, то заменить в нем 2 младших бита нулем, иначе – 2-й и 4-й единицами. Вывести исходную и преобразованную последовательности в символьной форме и в восьмеричных кодах.

```

#include <iostream>
#include <conio.h>
#include <ctype.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int c;
    char symbol[8], ch;

```

```

const char *vowel = "eyuioaEYUIOA";
cout << "Введите 8 символов латинского алфавита: ";
const int size_mass = 8;
for (int count = 0; count < size_mass;)
{
    c = getch();
    if(isalpha(c))
    {
        ch = static_cast<char>(c);
        symbol[count++] = ch;
        cout << ch << ' ';
    }
}
cout << endl << oct;
for(int i = 0; i < size_mass; i++)
{
    if (strchr(vowel, symbol[i]))
        ch = static_cast<char>(symbol[i] & 0xFC);
    else
        ch = static_cast<char>(symbol[i] | 0x0A);
    cout << symbol[i] << '\t' << static_cast<int>(symbol[i])
        << " | " << ch << '\t' << static_cast<int>(ch) << endl;
}
return 0;
}
q 161 | { 173
a 141 | ` 140
z 172 | z 172
w 167 | Δ 177
s 163 | { 173
x 170 | z 172
e 145 | d 144
d 144 | n 156

```

5) Вычислить сумму

$$S = \sum_{i=1}^{1000} r_i$$

$$r_1 = 1.23; \quad r_2 = 0.65; \quad r_3 = -0.25;$$

$$r_k = \frac{1.2 \sin(r_{k-1} + r_{k-2}) + 0.8 r_{k-3}}{1 - \cos(r_{k-1})} - [0.55 r_{k-2}]$$

$$k = 4, 5, \dots, 1000.$$

[] – обозначение целой части.

Массивом не пользоваться.

```

#include <iostream>
#include <math.h>
using namespace std;

double calculate(double r1, double r2, double r3)
{
    return (1.2 * sin(r3 + r2) + 0.8 * r1) /

```

```

        (1 - cos(r3)) - floor(0.55 * r2);
    }

int _tmain(int argc, _TCHAR* argv[])
{
    double r1 = 1.23, r2 = 0.65, r3 = -0.25, result;
    for(int i = 3; i < 1000; i++)
    {
        result = calculate(r1, r2, r3);
        r1 = r2; r2 = r3; r3 = result;
    }
    cout << "Результат вычисления рекуррентного выражения: " << result;
    return 0;
}

```

Результат вычисления рекуррентного выражения: 5.60909*e+205.

б) Задана последовательность вещественных чисел b_1, \dots, b_{30} .
Сформировать одномерный массив A такой, что:

$$a_i = 1 / (b_i - b_{i-1}), i = 2, 3, \dots, 30;$$

$$a_1 = 1 / (b_1 - b_{30}).$$

Последовательность b_1, \dots, b_{30} ввести с клавиатуры.

```

#include <iostream>
#include <math.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int i;
    const int size_mass = 30;
    double a[size_mass], b[size_mass];
    cout << "Введите последовательность вещественных чисел:" << endl;
    for(i = 0; i < size_mass; i++)
    {
        cout << "b" << i + 1 << " = ";
        cin >> b[i];
    }
    a[0] = 1.0 / (b[0] - b[size_mass - 1]);
    for(i = 1; i < size_mass; i++)
        a[i] = 1.0 / (b[i] - b[i - 1]);
    for(i = 0; i < size_mass; i++)
        cout << "b" << i + 1 << " = " << b[i] << " | " << "a" << i + 1
    << " = " << a[i] << endl;
    return 0;
}

```

7) Дана матрица $Z(4,6)$. Определить и вывести в массив B все элементы, которые в своей строке больше предыдущего и меньше последующего. Вывести исходную матрицу и полученный массив.

```

#include <iostream>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int i, j, elements = 0;
    const int row_size = 4, col_size = 6;

```

```

double matrix[row_size][col_size] =
    {{1.1, 4.7, 7.2, 5.0, 2.7, 8.3},
     {5.6, 6.0, 77.1, 99.0, 1.2, 3.4},
     {7.2, 5.4, 7.8, 7.9, 8.9, 10.0},
     {5.5, 2.3, 4.6, 6.0, 7.8, 11.2}};
double vec[row_size * col_size];
for(i = 0; i < row_size; i++)
    for(j = 0; j < col_size; j++)
    {
        if (j > 0 && j < col_size - 1 &&
            (matrix[i][j - 1] < matrix[i][j]
             && matrix[i][j] < matrix[i][j + 1]))
        {
            vec[elements++] = matrix[i][j];
        }
    }
cout << "Исходная матрица: " << endl;
for(i = 0; i < row_size; i++)
{
    for(j = 0; j < col_size; j++)
    {
        cout << matrix[i][j] << "    ";
    }
    cout << endl;
}
cout << endl << "Вектор результатов: " << endl;
for (i = 0; i < elements; i++)
    cout << vec[i] << "    ";
return 0;
}

```

Результирующий вектор:

4.7	6	77.1	7.8	7.9	8.9	4.6	6	7.8
-----	---	------	-----	-----	-----	-----	---	-----

8) Дана действительная квадратная матрица порядка n , все элементы которой различны. В этой матрице в каждой строке элементы, стоящие на нечетных местах, заменить суммой, на четных – произведением соответствующей пары. Матрица размещается в памяти динамически с помощью операции `new`, значение n вводится по запросу с клавиатуры. В конце работы программы освободить выделенную память. Вывести исходную и результирующую матрицы.

```

include <iostream>
#include <stdlib.h>
#include <time.h>
using namespace std;
int _tmain(int argc, _TCHAR* argv[])
{
    int n, i, j, temp;
    cout << "Введите размер квадратной матрицы n = ";
    cin >> n;
    int** matrix = new int*[n];
    for(i = 0; i < n; i++)
        matrix[i] = new int[n];
    srand(static_cast<unsigned>(time(NULL)));
    cout << "Исходная матрица:" << endl;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)

```

```

        {
            temp = rand();
            matrix[i][j] = temp;
            cout << temp << " ";
        }
        cout << endl;
    }
    cout << "Результирующая матрица:" << endl;
    for(i = 0; i < n; i++)
    {
        for(j = 0; j < n; j++)
        {
            matrix[i][j] = (j + 1) % 2 ? i + j + 2: (i + 1) * (j + 1);
            cout << matrix[i][j] << " ";
        }
        cout << endl;
    }
    for(i = 0; i < n; i++)
        delete[]matrix[i];
    delete[]matrix;
    return 0;
}

```

Исходная матрица при n = 3:			Результирующая матрица при n = 3:		
14568	15923	20173	2	2	4
4418	11273	14447	3	4	5
13180	32328	21502	4	6	6

9) Прочитать из файла строку символов. Заменить в ней символ, равный символу, введенному с клавиатуры, его порядковым номером. Новую строку не создавать. Вывести исходную и преобразованную строки.

```

#include <iostream>
#include <fstream>
using namespace std;
char* getStringFromFile(const char* filename) throw (char*)
{
    int len;
    ifstream in(filename, ios::binary | ios::in);
    if(!in) throw "Не возможно открыть файл";
    if(!(len = in.seekg(0, ios::end).tellg())) throw "Пустой файл";
    in.seekg(0, ios::beg);
    char *buffer = new char[len + 1];
    in.read(buffer, len);
    buffer[len] = 0;
    in.close();
    return buffer;
}
int _tmain(int argc, _TCHAR* argv[])
{
    char *str, ch;
    const char *filename = "test.txt";
    try
    {
        str = getStringFromFile(filename);
        cout << "Введите искомый символ: ";
        cin >> ch;
        cout << "Исходная строка: " << str << endl;
        cout << "Результирующая строка: ";
    }
}

```

```
        for(int i = 0; i < strlen(str); i++)
        {
            if(str[i] == ch) cout << i + 1;
            else cout << str[i];
        }
        delete[]str;
    }
    catch (char* s)
    {
        cerr << s << endl;
    }
    return 0;
}
```

Введите искомый символ: l

Исходная строка: Hello world!

Результирующая строка: He34o wor10d!

КОНТРОЛЬНАЯ РАБОТА №3. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ

1) Классы. Протокол класса. Конструкторы и деструкторы

Вариант №1

Создать абстрактный тип данных – класс вектор, который имеет указатель на `int`, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного `int`. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №2

Создать класс матрица. Данный класс содержит указатель на `int`, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №3

Создать класс типа – дата с полями: день (1-31), месяц (1-12), год (целое число). Класс имеет конструктор. Функции-члены установки дня, месяца и года, функции-члены получения дня, месяца и года, а также две

функции-члены печати: печать по шаблону: «5 января 1997 года» и «05.01.1997». Функции-члены установки полей класса должны проверять корректность задаваемых параметров.

Вариант №4

Создать абстрактный тип данных – класс вектор, который имеет указатель на float, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного float. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №5

Создать класс матрица. Данный класс содержит указатель на float, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №6

Создать класс типа – время с полями: час (0-23), минуты (0-59), секунды (0-59). Класс имеет конструктор. Функции-члены установки времени, функции-члены получения часа, минуты и секунды, а также две функции-члены печати: печать по шаблону: «16 часов 18 минут 3 секунды» и

«4 p.m. 18 минут 3 секунды». Функции-члены установки полей класса должны проверять корректность задаваемых параметров.

Вариант №7

Создать абстрактный тип данных – класс вектор, который имеет указатель на `double`, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, - размер вектора, - выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного `double`. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №8

Создать класс матрица. Данный класс содержит указатель на `double`, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №9

Создать класс типа – прямоугольник. Поля – высота и ширина. Функции-члены вычисляют площадь, периметр, устанавливают поля и возвращают значения. Функции-члены установки полей класса должны проверять корректность задаваемых параметров. Функция печати.

Вариант №10

Создать абстрактный тип данных – класс вектор, который имеет указатель на long, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного long. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №11

Создать класс матрица. Данный класс содержит указатель на long, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №12

Создать класс типа – циклическая очередь. Функции-члены получают элемент и вставляют элемент.

Вариант №13

Создать абстрактный тип данных – класс вектор, который имеет указатель на int, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, –

размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного `int`. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №14

Создать класс матрица. Данный класс содержит указатель на `int`, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №15

Создать класс типа – двухсвязный список. Функции-члены добавляют элемент к списку, удаляют элемент из списка. Отображают элементы списка от начала и от конца. Найти элемент в списке.

Вариант №16

Создать абстрактный тип данных – класс вектор, который имеет указатель на `float`, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый

элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного float. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №17

Создать класс матрица. Данный класс содержит указатель на float, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №18

Создать абстрактный тип данных – класс вектор, который имеет указатель на double, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного double. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №19

Создать класс матрица. Данный класс содержит указатель на double, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №20

Создать класс типа – односвязный список. Функции-члены добавляют элемент к списку, удаляют элемент из списка. Отображают элементы списка от начала. Найти элемент в списке.

Вариант №21

Создать абстрактный тип данных – класс вектор, который имеет указатель на long, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного long. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №22

Создать класс матрица. Данный класс содержит указатель на long, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя параметрами, деструктор. Определить методы доступа: возвращать значение

элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №23

Создать класс типа – окружность. Поля – радиус. Функции-члены вычисляют площадь, длину окружности, устанавливают поля и возвращают значения. Функции-члены установки полей класса должны проверять корректность задаваемых параметров. Функция печати.

Вариант №24

Создать абстрактный тип данных – класс вектор, который имеет указатель на `int`, число элементов и переменную состояния. Определить конструктор без параметров, конструктор с параметром, конструктор с двумя параметрами. Конструктор без параметров выделяет место для одного элемента и инициализирует его в ноль. Конструктор с одним параметром, – размер вектора, – выделяет место и инициализирует номером в массиве, конструктор с двумя параметрами выделяет место (первый аргумент) и инициализирует вторым аргументом. Деструктор освобождает память. Определить функцию, которая присваивает элементу массива некоторое значение (параметр по умолчанию), функцию, которая получает некоторый элемент массива. В переменную состояния устанавливать код ошибки, когда не хватает памяти, выходит за пределы массива. Определить функцию печати. Определить функции сложения, умножения, вычитания, которые производят эти арифметические операции с данными этого класса и встроенного `int`. Определить методы сравнения: больше, меньше или равно. Предусмотреть возможность подсчета числа объектов данного типа. Проверить работу этого класса.

Вариант №25

Прочитать из файла строку символов. Поместить в ее начале все латинские заглавные буквы, которые в ней присутствуют. Новую строку не создавать. Вывести исходную и преобразованную строки.

Вариант №26

Создать класс матрица. Данный класс содержит указатель на `int`, размер строк и столбцов и состояние ошибки. Определить конструктор без параметров, конструктор с одним параметром и конструктор с двумя

параметрами, деструктор. Определить методы доступа: возвращать значение элемента (i, j) и адрес этого элемента. Определить функцию печати. Определить функции сложения и вычитания (матрицы с матрицей), умножение матрицы на матрицу. Определить умножение матрицы на число. Проверить работу этого класса. В случае нехватки памяти, несоответствия размерностей, выхода за пределы устанавливать код ошибки.

Вариант №27

Создать класс типа – дата с полями: день (1-31), месяц (1-12), год (целое число). Класс имеет конструктор. Функции-члены установки дня, месяца и года, функции-члены получения дня, месяца и года, а также две функции-члены печати: печать по шаблону: «5 января 2008 года» и «05.01.2008». Функции-члены установки полей класса должны проверять корректность задаваемых параметров. Функция-член дает приращение на 1 день.

Вариант №28

Создайте класс, который реализует безопасный двумерный массив целых чисел размерностью (2x3). Класс должен содержать методы для произвольного защищенного доступа к элементам массива, метод для записи элементов массива через возвращаемое значение ссылки.

2) Преобразование типов. Дружественные функции. Конструктор копирования

Вариант №1

Создать класс комплексных чисел, члены класса – реальная и мнимая части. Класс имеет конструктор по умолчанию, конструктор, преобразующий float в объект класса. Определить оператор преобразования объекта типа комплексных чисел в число типа float. Создать класс вещественных чисел. Определить взаимное преобразование с классом комплексных чисел.

Вариант №2

Создать класс комплексных чисел. Определить перегруженную функцию, возвращающую максимальный из двух аргументов. Функция не является членом класса комплексных чисел. Перегруженные функции имеют аргументы типа int, double, complex. Тело перегруженных функций должны быть одинаковыми.

Вариант №3

Создать два класса вектор (`int *`) и матрица (`int **`). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор как дружественную.

Вариант №4

Создать класс вещественных чисел. Класс имеет конструктор по умолчанию, конструктор – преобразующий `float` в объект класса. Определить оператор преобразования объекта типа вещественных чисел в число типа `float`. Создать класс целых чисел. Определить взаимное преобразование с классом вещественных чисел.

Вариант №5

Создать класс комплексных чисел. Определить перегруженную функцию, возвращающую минимальный из двух аргументов. Функция не является членом класса комплексных чисел. Перегруженные функции имеют аргументы типа `int`, `double`, `complex`. Тело перегруженных функций должны быть одинаковыми.

Вариант №6

Создать два класса: целые (`Integer`) и матрица (`int *`). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на целое как дружественную.

Вариант №7

Создать класс целых чисел. Класс имеет конструктор по умолчанию, конструктор, преобразующий `int` в объект класса. Определить оператор преобразования объекта типа целых чисел в число типа `int`. Создать класс вещественных чисел. Определить взаимное преобразование с классом целых чисел.

Вариант №8

Создать класс целых чисел `Integer`. Определить перегруженную функцию, возвращающую максимальное из двух аргументов. Функция не является членом класса целых чисел. Перегруженные функции имеют аргументы типа `int`, `double`, `Integer`. Тело перегруженных функций должны быть одинаковыми.

Вариант №9

Создать два класса: вещественные (Float) и матрица (float **). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вещественное (Float) как дружественную.

Вариант №10

Создать класс целых чисел Integer. Определить перегруженную функцию, возвращающую минимальное из двух аргументов. Функция не является членом класса целых чисел. Перегруженные функции имеют аргументы типа int, double, Integer. Тело перегруженных функций должны быть одинаковыми.

Вариант №11

Создать класс вещественных чисел Double. Определить перегруженную функцию, возвращающую максимальное из двух аргументов. Функция не является членом класса Double. Перегруженные функции имеют аргументы типа int, double, Double. Тело перегруженных функций должны быть одинаковыми.

Вариант №12

Создать два класса вектор (double *) и матрица (double **). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор как дружественную.

Вариант №13

Создать два класса вектор (float *) и матрица (float **). Определить конструкторы – по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор как дружественную.

Вариант №14

Создать класс вещественных чисел Double. Определить перегруженную функцию, возвращающую максимальное из двух аргументов. Функция не является членом класса Double. Перегруженные функции имеют аргументы типа int, double, Double. Тело перегруженных функций должны быть одинаковыми.

Вариант №15

Создать два класса вектор (`long *`) и матрица (`long **`). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор как дружественную.

Вариант №16

Создать класс вещественных чисел `Float`. Определить перегруженную функцию, возвращающую минимальное из двух аргументов. Функция не является членом класса `Float`. Перегруженные функции имеют аргументы типа `int`, `float`, `Float`. Тело перегруженных функций должны быть одинаковыми.

Вариант №17

Создать класс комплексных чисел. Определить перегруженную функцию, возвращающую максимальный из двух аргументов. Функция не является членом класса комплексных чисел. Перегруженные функции имеют аргументы типа `int`, `double`, `complex`. Тело перегруженных функций должны быть одинаковыми.

Вариант №18

Создать два класса вектор (`int *`) и матрица (`int **`). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор как дружественную.

Вариант №19

Создать класс вещественных с двойной точностью чисел. Класс имеет конструктор по умолчанию, конструктор – преобразующий `double` в объект класса. Определить оператор преобразования объекта типа вещественных чисел с двойной точностью в число типа `double`. Создать класс целых чисел. Определить взаимное преобразование с классом вещественных чисел.

Вариант №20

Создать класс комплексных чисел. Определить перегруженную функцию, возвращающую максимальный из двух аргументов. Функция не является членом класса комплексных чисел. Перегруженные функции имеют аргументы типа `int`, `double`, `complex`. Тело перегруженных функций должны быть одинаковыми.

Вариант №21

Создать два класса вектор (`float *`) и матрица (`float **`). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор.

Вариант №22

Определить два класс, строку с преобразованием из `char *` в строку и обратно и Целое `Int` с преобразованием из `int` и обратно, а также взаимное преобразование `String` и `Int`.

Вариант №23

Создать класс целых чисел `Integer`. Определить перегруженную функцию, возвращающую максимальное из двух аргументов. Функция не является членом класса целых чисел. Перегруженные функции имеют аргументы типа `int`, `double`, `Integer`. Тело перегруженных функций должны быть одинаковыми.

Вариант №24

Создать два класса вектор (`long *`) и матрица (`long **`). Определить конструкторы по умолчанию, с параметром, для класса матрица с двумя параметрами, копирования, деструкторы. Определить функцию умножения матрицу на вектор

Вариант №25

Создать класс коротких целых чисел. Класс имеет конструктор по умолчанию, конструктор, преобразующий `short int` в объект класса. Определить оператор преобразования объекта типа короткое целое в число типа `short int`. Создать класс вещественных чисел. Определить взаимное преобразование с классом коротких целых чисел.

Вариант №26

Создать класс вещественных чисел `Double`. Определить перегруженную функцию, возвращающую максимальное из двух аргументов. Функция не является членом класса `Double`. Перегруженные функции имеют аргументы типа `int`, `double`, `Double`. Тело перегруженных функций должны быть одинаковыми.

Вариант №27

Создать два класса: целые (Integer) и вектор (int *). Определить конструкторы по умолчанию, с параметром, для класса вектор с двумя параметрами, копирования, деструкторы. Определить функцию умножения вектора на целое как дружественную.

Вариант №28

Создать класс длинных целых чисел. Класс имеет конструктор по умолчанию, конструктор, преобразующий long в объект класса. Определить оператор преобразования объекта типа длинных целых чисел в число типа long. Создать класс целых чисел. Определить взаимное преобразование с классом длинных целых чисел.

3) Перегрузка операторов

Вариант №1

Создать класс целых чисел. Определить оператор ++, как функцию-член и -- как дружественную функцию.

Вариант №2

Создать класс целых чисел. Определить оператор +, как функцию-член и – как дружественную функцию.

Вариант №3

Создать класс целых чисел. Определить оператор --, как функцию-член и ++ как дружественную функцию.

Вариант №4

Создать класс координат. Определить оператор +, как функцию-член и – как дружественную функцию. Сложить и вычесть координаты с друг другом и с числом. Присвоить координаты, сравнить координаты (==, !=).

Вариант №5

Создать класс вещественных чисел. Определить оператор ++, как функцию-член и -- как дружественную функцию.

Вариант №6

Создать класс целых чисел. Определить оператор –, как функцию-член и + как дружественную функцию.

Вариант №7

Создать объект типа очередь. Перегрузить оператор ++ как функцию член и -- как дружественную функцию. (Как постфиксными так префиксными). ++ добавляет элемент в очередь (пустой элемент, например `int i=0`), -- вытаскивает элемент из очереди. Оператор ! проверяет очередь на пустоту.

Вариант №8

Создать объект типа стек. Перегрузить оператор + как функцию член и * как дружественную функцию. + складывает элемент в новый стек, * умножает верхушку стека на параметр. Стеки можно присваивать, проверять на равенство == или !=, вводить и выводить в поток, добавлять += элемент в стек.

Вариант №9

Создать объект типа стек. Перегрузить оператор ++ как функцию член и -- как дружественную функцию. (Как постфиксными так префиксными). ++ добавляет элемент новый в стек, -- удаляет верхушку стека. Оператор ! проверяет стек на пустоту.

Вариант №10

Создать объект типа очередь. Перегрузить оператор + как функцию член и * как дружественную функцию. + добавляет элемент в очередь, * умножает элемент в очереди. Вытаскивает элемент из очереди --. Очереди можно присваивать, проверять на равенство == или !=, вводить и выводить в поток, добавлять += элемент в очередь.

Вариант №11

Создать объект – связный двунаправленный список, с перегруженными унарными операциями ++, --, как движение по списку. (Как постфиксными так префиксными).

Вариант №12

Создать объект динамический стек. Перегрузить операции +, +=, -= (с извлечением элемента).

Вариант №13

Создать объект стек, перегрузив ++ и --. (Как постфиксными так префиксными). ++ Добавляет элемент в стек. -- извлекает элемент из стека.

Вариант №14

Создать объект очередь с перегруженными $+$, $+=$, добавление элемента в очередь и сложение очередей, $--$ для извлечения из очереди, $-$ для вычитания очередей.

Вариант №15

Создать класс – координаты с унарным $++$ и $--$, $-$. $++$ и $--$ постфиксная и префиксная. $-$ меняет знак у обеих координат. $++$ как функция-член, $--$ как дружественная функция.

Вариант №16

Создать класс целых чисел (`long`). Определить оператор $-$, как функцию-член и $+$ как дружественную функцию. Оператор присвоения, и сравнений.

Вариант №17

Создать класс вещественных чисел (`double`). Определить оператор $++$, как функцию-член и $--$ как дружественную функцию.

Вариант №18

Создать класс целых чисел (`long`). Определить оператор $+$, как функцию-член и $-$ как дружественную функцию.

Вариант №19

Создать класс вещественных чисел (`double`). Определить оператор $--$, как функцию-член и $++$ как дружественную функцию.

Вариант №20

Создать класс вещественных чисел (`double`). Определить оператор $-$, как функцию-член и $+$ как дружественную функцию.

Вариант №21

Создать класс целых чисел (`long`). Определить оператор $++$, как функцию-член и $--$ как дружественную функцию.

Вариант №22

Создать класс вещественных чисел (`double`). Определить оператор $+$, как функцию-член и $-$ как дружественную функцию.

Вариант №23

Создать класс целых чисел (long). Определить оператор --, как функцию-член и ++ как дружественную функцию.

Вариант №24

Создать класс вещественных чисел. Определить оператор -, как функцию-член и + как дружественную функцию.

Вариант №25

Создать объект – очередь с перегруженными операциями ++ как функциями-членами, -- как дружественными функциями. (Как постфиксными так префиксными).

Вариант №26

Создать объект – однонаправленный список, в котором определены операции, + - добавляет в конец списка, += добавляет в этот же список в конец списка. - удаляет указанный элемент из списка (номер элемента через параметр), = - присвоение списков, сравнение списков ==, !=, >, <, >=, <=, [] получение элемента списка, ++ - устанавливает указатель на следующий элемент. () выдать подсписок от первого до второго элемента.

Вариант №27

Создать объект - однонаправленный список, в котором определены операции, ++ - добавляет в конец списка, -- удаляет элемент из списка. (Как постфиксными так префиксными).

Вариант №28

Создать класс тип строка с перегруженным оператором присваивания = и индексирования массива [] для работы со строкой как с массивом.

4) Наследование. Иерархия и контейнерные классы

Вариант №1

Создать класс студент, имеющий имя (указатель на строку), курс и идентификационный номер. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – студент-дипломник, имеющий тему диплома. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения названия диплома и идентификационного номера.

Вариант №2

Создать класс цех, имеющего площадь. Определить конструктор и метод доступа. Создать класс цехов по производству минеральных удобрений, содержащий площадь, этаж. Определить конструкторы, методы доступа. Определить public-производный класс цехов по производству минеральных удобрений разных предприятий (дополнительный параметр – название предприятия). Определить конструкторы, деструктор и функцию печати.

Вариант №3

Создать класс сорбентов, имеющий классификацию (указатель на строку), диаметр гранул, механическую прочность, удельную поверхность и др. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – сорбенты для газоочистки. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения названия марки сорбента и значения диаметра гранул.

Вариант №4

Создать класс брак полимерной пленки (царапина, имеющая длину). Определить конструкторы и метод доступа. Создать класс полимерных пленок, содержащий класс брак. Дополнительно есть цвет (указатель на строку), толщина. Определить конструкторы и деструктор. Определить public- производный класс полимерная пленка, имеющая дополнительно марку (указатель на строку). Определить конструкторы, деструкторы и функцию печати.

Вариант №5

Создать класс машина, имеющий марку (указатель на строку), число цилиндров, мощность. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – грузовики, имеющий грузоподъемность кузова. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения марки и грузоподъемности.

Вариант №6

Создать класс двигатель, имеющий мощность. Определить конструкторы и метод доступа. Создать класс машин, содержащий класс двигатель. Дополнительно есть марка (указатель на строку), цена. Определить конструкторы и деструктор. Определить public- производный

класс грузовик, имеющий дополнительно грузоподъемность. Определить конструкторы, деструкторы и функцию печати.

Вариант №7

Создать иерархию классов химических реакторов. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №8

Создать иерархию классов точка и цветная точка. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №9

Создать класс строк и производный от него – класс строк из цифр. Определить конструкторы и деструкторы, переопределить вывод и ввод в поток. Перегрузить оператор присваивания и конструктор копирования в базовом и производном классе.

Вариант №10

Создайте класс точка, которая имеет координаты. Класс эллипсов, и класс окружностей. Определить иерархию типов. Определить функции печати, конструкторы, деструкторы, вычисление площади.

Вариант №11

Используя иерархию и композицию классов, создать бинарное дерево. У бинарного дерева есть корневой узел. Мы можем вставлять узел. Мы можем обходить в ширину и обратный обход. Узел может быть помещен в дерево двоичного поиска только в качестве конечного узла. Если дерево является пустым, то создается новый экземпляр класса узел дерева и узел помещается в дерево. Если дерево не является пустым, то программа сравнивает вставляемое в дерево значение со значением в корневом узле и если меньше, то помещает в левые поддеревья, а если больше, то в правые. Если значения равны, то выводится сообщение, что повтор и не вставляется.

Вариант №12

Создать класс четырехугольников, квадратов и прямоугольников. Создать из них иерархию. Определить функции печати, конструкторы и деструкторы, вычисление площади и периметра.

Вариант №13

Создать иерархию классов вектор и безопасный вектор с проверкой выхода за пределы. Безопасный вектор определяет переменные нижний и верхний предел. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №14

Создать класс оборудование для гранулирования, одношнековый экструдер, планетарный гранулятор, тарельчатый гранулятор, шнековый пресс. Создать из них иерархию. Определить функции печати, конструкторы и деструкторы.

Вариант №15

Создать классы сырьё, полупродукт и продукция. Составить из них иерархию или композицию.

Вариант №16

Создать иерархию классов студент и преподаватель. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №17

Создать класс жидкость, имеющий название (указатель на строку), плотность. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – жидкость, имеющая вредные вещества. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения плотности и вредных веществ.

Вариант №18

Создать иерархию классов студент и студент дипломник. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №19

Создать класс человек, имеющий имя (указатель на строку), возраст, вес. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – студент, имеющий класс (год обучения). Определить конструкторы по умолчанию и с разным числом параметров,

деструкторы, функцию печати. Определить функции переназначения возраста и класса.

Вариант №20

Создать класс жесткий диск, имеющий объем (Мбайт). Определить конструкторы и метод доступа. Создать класс компьютер, содержащий класс жесткий диск. Дополнительно есть марка (указатель на строку), цена. Определить конструкторы и деструктор. Определить private-, public-производный класс компьютеров с монитором, имеющий дополнительно размер монитора. Определит конструкторы, деструкторы и функцию печати.

Вариант №21

Создать иерархию классов четырехугольник и квадрат. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №22

Создать класс окно, имеющий координаты верхнего левого и нижнего правого угла, цвет фона (указатель на строку). Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – окно с меню, имеющий строку меню. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения цвета фона и строки меню.

Вариант №23

Создать класс процессор, имеющий мощность (МГц). Определить конструкторы и метод доступа. Создать класс компьютер, содержащий класс процессор. Дополнительно есть марка (указатель на строку), цена. Определить конструкторы и деструктор. Определить private-, public-производный класс компьютеров с монитором, имеющий дополнительно размер монитора. Определить конструкторы, деструкторы и функцию печати.

Вариант №24

Создать класс точка, имеющая координаты. Определить конструкторы, деструктор и функцию печати. Создать public-производный класс – цветная точка, имеющий цвет точки. Определить конструкторы по умолчанию и с разным числом параметров, деструкторы, функцию печати. Определить функции переназначения цвета и координат точки, вывода точки на экран.

Вариант №25

Создать класс колесо, имеющий радиус. Определить конструкторы и метод доступа. Создать класс машин, содержащий класс колесо. Дополнительно есть марка (указатель на строку), цена. Определить конструкторы и деструктор. Определить public- производный класс грузовик, имеющий дополнительно грузоподъемность. Определить конструкторы, деструкторы и функцию печати.

Вариант №26

Создать иерархию классов окно и окно с заголовком. Переопределить вывод в поток и ввод из потока, конструктор копирования, оператор присваивания через соответствующие функции базового класса.

Вариант №27

Создать класс точка и производные от него – окружность и эллипс. Определить конструкторы, деструктор и функцию печати. Определить функции переустановки центра окружности и эллипса.

Вариант №28

Создать иерархию классов транспортное средство с дочерними классами: тип двигателя и пассажироемкость. Создать класс автомобиль, который является наследником классов тип двигателя и пассажироемкость. В каждом классе создать метод отображения особенностей автомобиля. Продемонстрировать виртуальное наследование.

5) Виртуальные функции

Вариант №1

Создать абстрактный базовый класс с виртуальной функцией – площадь. Создать производные классы: прямоугольник, круг, прямоугольный треугольник, трапеция со своими функциями площади. Для проверки определить массив ссылок на абстрактный класс, которым присваиваются адреса различных объектов. Площадь трапеции: $S=(a+b) \cdot h/2$.

Вариант №2

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №3

Создать абстрактный класс с виртуальной функцией: норма. Создать производные классы: комплексные числа, вектор из 10 элементов, матрица (2x2). Определить функцию нормы – для комплексных чисел – модуль в квадрате, для вектора – корень квадратный из суммы элементов по модулю, для матрицы – максимальное значение по модулю.

Вариант №4

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №5

Создать абстрактный класс (кривые) вычисления координаты y для некоторой x . Создать производные классы: прямая, эллипс, гипербола со своими функциями вычисления y в зависимости от входного параметра x .

Вариант №6

Уравнение прямой: $y=ax+b$, эллипса: $x^2/a^2+y^2/b^2=1$, гиперболы: $x^2/a^2-y^2/b^2=1$

Вариант №7

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №8

Создать абстрактный базовый класс с виртуальной функцией – сумма прогрессии. Создать производные классы: арифметическая прогрессия и геометрическая прогрессия. Каждый класс имеет два поля типа double. Первое – первый член прогрессии, второе (double) – постоянная разность (для арифметической) и постоянное отношение (для геометрической). Определить функцию вычисления суммы, где параметром является количество элементов прогрессии.

Арифметическая прогрессия $a_j=a_0+jd, j=0,1,2,\dots$

Сумма арифметической прогрессии: $s_n=(n+1)(a_0+a_n)/2$

Геометрическая прогрессия: $a_j=a_0r^j, j=0,1,2,\dots$

Сумма геометрической прогрессии: $s_n=(a_0-a_nr)/(1-r)$

Вариант №9

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №10

Создать базовый класс список. Реализовать на базе списка стек и очередь с виртуальными функциями вставки и вытаскивания.

Вариант №11

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №12

Создать базовый класс – фигура, и производные классы – круг, прямоугольник, трапеция. Определить виртуальные функции площадь, периметр и вывод на печать.

Вариант №13

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №14

Создать базовый класс – работник и производные классы – служащий с почасовой оплатой, служащий в штате и служащий с процентной ставкой. Определить функцию начисления зарплаты.

Вариант №15

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №16

Создать абстрактный базовый класс с виртуальной функцией – площадь поверхности. Создать производные классы: параллелепипед, тетраэдр, шар со своими функциями площади поверхности. Для проверки определить массив ссылок на абстрактный класс, которым присваиваются адреса различных объектов.

Площадь поверхности параллелепипеда: $S=6xy$. Площадь поверхности шара: $S=4\pi r^2$. Площадь поверхности тетраэдра: $S=\sqrt{3}a^2$.

Вариант №17

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №18

Создать класс вещество, производные от которого органические вещества и неорганические. Определить виртуальную функцию на результат взаимодействия с водой.

Вариант №19

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №20

Создать абстрактный базовый класс с виртуальной функцией – объем. Создать производные классы: параллелепипед, пирамида, тетраэдр, шар со своими функциями объема. Для проверки определить массив ссылок на абстрактный класс, которым присваиваются адреса различных объектов.

Объем параллелепипеда $V=xyz$, пирамиды $V=xyh$, тетраэдра $V = \sqrt{2} / 12 \cdot a^3$, шара $V=4/3 \cdot \pi \cdot r^3$.

Вариант №21

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №22

Создать абстрактный класс – агрегатное состояние вещества. Определить производные классы – твердое тело, жидкость, газообразное состояние, плазма. У твердого тела определить производные классы аморфное и кристаллическое, у жидкости определить производные классы атомарная, ассоциированная и т.д. Определить виртуальные функции описания твердого тела, жидкости, газообразного состояния, плазмы.

Вариант №23

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №24

Создать базовый класс – химическая реакция определить виртуальную функцию печати. Создать производный класс классификация (по фазовому составу реагирующей системы; по изменению степеней окисления реагентов; по тепловому эффекту реакции; по типу превращений реагирующих частиц). Создать производный класс от последнего класса:

по фазовому составу реагирующей системы (гомогенные гомофазные реакции, гетерогенные гетерофазные реакции, гетерогенные гомофазные реакции, гомогенные гетерофазные реакции);

по изменению степеней окисления реагентов (окислительно-восстановительные реакции, не окислительно-восстановительные реакции);

по тепловому эффекту реакции (эндотермические, экзотермические);

по типу превращений реагирующих частиц (соединения, разложения, замещения, обмена). Написать функцию печати названия класса.

Вариант №25

Создать класс – данные – абстрактный базовый класс. Создать производные классы – данные типа сигнал, данные типа результат обработки и вспомогательные данные. Все данные имеют функции отображения, сохранения и обработки.

Вариант №26

Создать абстрактный базовый класс с виртуальной функцией – корни уравнения. Создать производные классы: класс линейных уравнений и класс квадратных уравнений. Определить функцию вычисления корней уравнений.

Вариант №27

Создать абстрактный базовый класс с виртуальной функцией – процессы химической технологии. Создать производные классы: гидромеханические; тепловые; массообменные (или диффузионные) процессы; химические процессы; механические процессы. Написать функцию печати.

Вариант №28

Создать класс – связанный список – абстрактный базовый класс. Интерфейс списка определяется с помощью чистых виртуальных функций store() и retrieve(). Для хранения значения в списке вызывается функция store(). Для выборки значения из списка вызывается функция retrieve(). В базовом классе list для выполнения этих действий никакого встроенного метода не задается. Необходимо реализовать списки двух типов: очередь и стек.

б) Применение основ программирования на языке высокого уровня в области химической технологии

Создать класс «Скорость химической реакции».

Создать процедуру ввода параметров химической реакции:

- стехиометрические коэффициенты α_i ;
- число исходных реагентов m ;
- константа скорости химической реакции k ;
- концентрации исходных реагентов во времени $C_i(t)$.

Создать функцию расчёта скорости химической реакции.

Для вывода уравнения скорости химической реакции используется закон действия масс (основной постулат химической кинетики).

Если в реагирующей системе протекает реакция в соответствии со следующим стехиометрическим уравнением:



то в соответствии с законом действия масс скорость элементарной химической реакции пропорциональна **концентрациям реагирующих веществ** в степенях, равных их стехиометрическим коэффициентам:

$$w = k \prod_{i=1}^m C_i^{|\alpha_i|},$$

где k – константа скорости реакции (размерность зависит от порядка реакции); m – число реагентов в реакции; C_i – концентрация i -го компонента, моль/м³; α_i – стехиометрический коэффициент i -го компонента.

Создать функцию вывода результатов расчёта скорости реакции в каждый момент времени.

Номер варианта	Химическая реакция
1	$A + B = C + D$
2	$A + 2B = C + D + E$
3	$2A + 4B = C$
4	$A + 3B = 2C$
5	$2A + B = 2C$
6	$A = B + 2C$
7	$2A + 3B = 2C$
8	$2A = 2B + C$
9	$A + B + C = D + E$
10	$3A + 2B = C + 5D$
11	$4A = 4B + 2C + D$
12	$A + 4B = C + 2D + 2E$
13	$3A + 8B = 3C + 2D + 4E$
14	$3A + B = C + D + 2E$
15	$6A = 2B + C$
16	$2A + 5B = 2C + 4D + E$
17	$4A + 6B = 2C + 2D$
18	$3A + 8B + 5C + 18D = 3E + 8F$
19	$A + 10B = C$
20	$3A + 4B = C$
21	$2A + 9B = C + 7D$
22	$5A + 3B = 3C + 2D$
23	$3A + B = C + D$
24	$A + 2B = 2C + D + E$
25	$7A + 2B = 4C + D$
26	$2A + 9B = C + 7D$
27	$A + B + 3C = 3D + E$
28	$2A + 4B + C = D + 2E$

Пример выполнения контрольной работы № 3

Для написания программ использовалась IDE Visual Studio 2005, программы представляют консольные приложения.

Вариант №28

1) Создайте класс, который реализует безопасный двухмерный массив целых чисел размерностью (2x3). Класс должен содержать методы для произвольного защищенного доступа к элементам массива, метод для записи элементов массива через возвращаемое значение ссылки.

```
#include <iostream>
#include <cstdlib>
using namespace std;
class array
{
    int isize, jsize;
    int *p;
public:
    array(int i, int j);
    int&put(int i, int j);
    int get(int i, int j);
};
array::array(int i, int j)
{
    p = new int[i * j];
    isize = i;
    jsize = j;
}
int & array::put(int i, int j)
{
    if (i < 0 || i >= isize || j < 0 || j >= jsize)
    {
        cerr << "Ошибка, нарушены границы массива!!!!" << endl;
        exit(1);
    }
    return p[i* jsize + j];
}
int array::get(int i, int j)
{
    if (i < 0 || i >= isize || j < 0 || j >= jsize)
    {
        cerr << "Ошибка, нарушены границы массива!!!!" << endl;
        exit(1);
    }
    return p[i* jsize + j];
}
int _tmain(int argc, _TCHAR* argv[])
{
    array a(2, 3);
    int i, j;
    for(i = 0; i < 2; i++)
        for(j = 0; j < 3; j++)
            a.put(i, j) = i + j;
    for(i = 0; i < 2; i++)
        for(j = 0; j < 3; j++)
            cout << a.get(i, j) << ' ';
```

```

        a.put(10, 10);
        return 0;
    }

```

Результат работы: 0 1 2 1 2 3

Ошибка, нарушены границы массива!!!!

2) Создать класс длинных целых чисел. Класс имеет конструктор по умолчанию, конструктор, преобразующий long в объект класса. Определить оператор преобразования объекта типа длинных целых чисел в число типа long. Создать класс целых чисел. Определить взаимное преобразование с классом длинных целых чисел.

```

class Long
{
    long value;
public:
    Long(): value(0){}
    Long(long l):value(l) {}
    operator long() {return value;}
    operator int() {return value;}
};
class Int
{
    int value;
public:
    Int(): value(0){}
    Int(int i):value(i) {}
    Int(Long l)
    {
        value = static_cast<int>(l);
    }
    operator int() {return value;}
    operator Long() {return Long(value);}
};
int _tmain(int argc, _TCHAR* argv[])
{
    Long l1(100L), l2(50L);
    Int i1(10);
    long l = 1L + static_cast<long>(l2);
    Long ll = i1;
    return 0;
}

```

3) Создать класс тип строка с перегруженным оператором присваивания = и индексирования массива [] для работы со строкой как с массивом.

```

#include <iostream>
#include <cstring>
#include <cstdlib>
using namespace std;
class strtype
{
    char *p;
    int len;
public:
    strtype(char*s);

```

```

~strtype(){delete p;}
char *get() {return p;}
strtype &operator=(strtype &ob);
char operator[](int i);
};
strtype::strtype(char*s)
{
    int l = strlen(s) + 1;
    p = new char[l];
    len = l;
    strcpy(p, s);
}
strtype &strtype::operator=(strtype &ob)
{
    if (len < ob.len)
    {
        delete[]p;
        p = new char[ob.len];
    }
    len = ob.len;
    strcpy(p, ob.p);
    return *this;
}
char strtype::operator[](int i)
{
    if (i > len)
    {
        cerr << "Ошибка, индекс не может быть больше количества символов в строке";
        exit(1);
    }
    return p[i];
}
int _tmain(int argc, _TCHAR* argv[])
{
    strtype a("Hello!"), b("Hi!");
    cout << a.get() << endl;
    cout << b.get() << endl;
    for(int i = 0; i < strlen(a.get()); i++)
        cout << a[i] << " ";
    cout << endl;
    a = b;
    cout << a.get() << endl;
    cout << b.get() << endl;
    return 0;
}

```

Результат работы:

Hello!

Hi!

H e l l o !

Hi!

Hi!

4) Создать иерархию классов транспортное средство с дочерними классами: тип двигателя и пассажироемкость. Создать класс автомобиль, который является наследником классов тип двигателя и пассажироемкость. В каждом классе создать метод отображения особенностей автомобиля. Продемонстрировать виртуальное наследование.

```

#include <iostream>
using namespace std;
class vehicle
{
    int num_whels;
    int range;
public:
    vehicle(int w, int r): num_whels(w), range(r){}
    void showv()
    {
        cout << "число колес " << num_whels << endl;
        cout << "Грузоподъемность " << range << endl;
    }
};
enum motor {gas, electric, diesel};
class motorized : public virtual vehicle
{
    enum motor mtr;
public:
    motorized(enum motor m, int w, int r): vehicle(w, r), mtr(m){}
    void showm()
    {
        cout << "Мотор:";
        switch (mtr)
        {
            case gas: cout << "На газе" << endl;
                break;
            case electric: cout << "На электроэнергии" << endl;
                break;
            case diesel: cout << "Дизельный" << endl;
                break;
        }
    }
};
class road_use: public virtual vehicle
{
    int passengers;
public:
    road_use(int p, int w, int r): vehicle(w, r), passengers(p) {}
    void showr()
    {
        cout << "Пассажироемкость " << passengers << endl;
    }
};
enum steering {power, rack_pinion, manual};
class car: public motorized, public road_use
{
    enum steering strng;
public:
    car(enum steering s, enum motor m, int w, int r, int p):
        motorized(m, w, r), road_use(p, w, r), vehicle(w, r), strng(s) {}
    void show ()
    {
        showv(); showr(); showm();
        cout << "Управление: ";
        switch (strng)
        {
            case power: cout << "Силовой привод" << endl;
                break;
            case rack_pinion: cout << "Механический привод" << endl;
                break;
            case manual: cout << "Ручной привод" << endl;
                break;
        }
    }
};

```

```

    }
};
int _tmain(int argc, _TCHAR* argv[])
{
    car c(power, gas, 4, 500, 5);
    c.show();
    return 0;
}

```

Результат работы:

Число колес: 4

Грузоподъемность: 500

Пассажироемкость: 5

Мотор: на газе

Управление: силовой привод

5) Создать класс – связанный список – абстрактный базовый класс. Интерфейс списка определяется с помощью чистых виртуальных функций store() и retrieve(). Для хранения значения в списке вызывается функция store(). Для выборки значения из списка вызывается функция retrieve(). В базовом классе list для выполнения этих действий никакого встроенного метода не задается. Необходимо реализовать списки двух типов: очередь и стек.

```

#include <iostream>
#include <cstdlib>
#include <cctype>
using namespace std;
class list
{
public:
    list *head;
    list *tail;
    list *next;
    int num;
    list () { head = tail = next = NULL; }
    virtual void store (int i) = 0;
    virtual int retrieve ( ) = 0;
};
class queue: public list
{
public:
    void store(int i);
    int retrieve();
};
void queue::store (int i)
{
    list *item = new queue;
    item->num = i;
    if (tail) tail->next = item;
    tail = item;
    item->next = NULL;
    if (!head) head = tail;
}
int queue::retrieve()
{
    int i;

```

```

list *p;
if(!head)
{
    cout << "Список пуст" << endl;
    return 0;
}
i = head->num;
p = head;
head = head->next;
delete p;
return i;
}
class stack: public list {
public:
    void store(int i);
    int retrieve();
};
void stack::store(int i)
{
    list *item = new stack;
    item->num = i;
    if(head) item->next = head;
    head = item;
    if(!tail) tail = head;
}
int stack::retrieve()
{
    int i;
    list *p;
    if(!head)
    {
        cout << "Список пуст" << endl;
        return 0;
    }
    i = head->num;
    p = head;
    head = head->next;
    delete p;
    return i;
}
int _tmain(int argc, _TCHAR* argv[])
{
    list *p;
    queue q_ob;
    p = &q_ob;
    p->store(1);
    p->store(2);
    p->store(3);
    cout << "Очередь: ";
    cout << p->retrieve();
    cout << p->retrieve();
    cout << p->retrieve();
    cout << endl;
    stack s_ob;
    p = &s_ob;
    p->store(1);
    p->store(2);
    p->store(3);
    cout << "Стек: ";
    cout << p->retrieve();
    cout << p->retrieve();
    cout << p->retrieve();
    cout << endl;
    return 0;
}

```

}

Результат работы:

Очередь: 123

Стек: 321

б) Расчет скоростей химической реакции $2A + 4B + C = D + 2E$ на заданных временных интервалах.

```
#include <iostream>
#include <math.h>
using namespace std;
class ChemSpeed
{
    double k;
    double *a, *result;
    double **c;
    double time;
    int m;
    int t;
public:
    ChemSpeed(): a(NULL), result(NULL), c(NULL) {}
    void inputData()
    {
        int i, j;
        cout << "Введите количество реагентов смеси m = ";
        cin >> m;
        cout << "Введите число замеров концентраций t = ";
        cin >> t;
        cout << "Введите шаг по времени time = ";
        cin >> time;
        cout << "Введите константу скорости реакции k = ";
        cin >> k;
        a = new double[m];
        c = new double*[m];
        for(i = 0; i < m; i++)
            c[i] = new double[t];
        cout << "Введите стехиометрические коэффициенты: " << endl;
        for(i = 0; i < m; i++)
        {
            cout << "a" << i + 1 << " = ";
            cin >> a[i];
        }
        for(i = 0; i < m; i++)
        {
            cout << "Введите концентрации компонента №" << i + 1 <<
            endl;
            for(j = 0; j < t; j++)
            {
                cout << "c[" << j << "] = ";
                cin >> c[i][j];
            }
        }
    }
    void calculate()
    {
        double temp;
        result = new double[t];
        for(int j = 0; j < t; j++)
        {
            temp = k;
```

```

        for(int i = 0; i < m; i++)
            temp *= pow(c[i][j], fabs(a[i]));
        result[j] = temp;
    }
}
void printResult()
{
    cout << "    time    |      w      " << endl;
    cout << "-----|-----" << left << endl;
    for(int i = 0; i < t; i++)
    {
        cout.width(10);
        cout.precision(5);
        cout << time * (i + 1) << "|" << result[i] << endl;
    }
}
~ChemSpeed()
{
    if(a) delete[]a;
    if(result) delete[]result;
    if(c)
    {
        for(int i = 0; i < m; i++)
            delete[]c[i];
        delete[]c;
    }
}
};
int _tmain(int argc, _TCHAR* argv[])
{
    ChemSpeed speed;
    speed.inputData();
    speed.calculate();
    speed.printResult();
    return 0;
}

```

Входные данные: $m = 3$, $t = 3$, $time = 5$, $k = 1.5$

a1	a2	a3
2	4	1

Время	Концентрация А	Концентрация В	Концентрация С
5	0.9	0.75	0.55
10	0.8	0.62	0.50
15	0.5	0.20	0.36

Результат расчета скорости химической реакции на временных интервалах:

Время	Скорость реакции
5	0.2114
10	0.0709
15	0.0002

СПИСОК ЛИТЕРАТУРЫ

- 1) Страуструп, Бьерн. Язык программирования С++ / Бьерн Страуструп; Пер. с англ. С. Анисимова, М. Кононова; Под ред. Ф. Андреева, А. Ушакова. – Спец. изд. – М.: Бином; СПб.: Нев. диалект, 2001. – 1098 с.
- 2) Подбельский, В.В. Язык СИ++ / В.В. Подбельский. – 5-е изд. – М.: Финансы и статистика, 2008. – 559 с.
- 3) Шилдт, Герберт. С++. Базовый курс / Герберт Шилдт; [пер. с англ. Н.М. Ручко]. – 3-е изд. – М.: Вильямс, 2008. – 620 с.
- 4) Эккель, Брюс. Философия С++: Практик. программирование / Брюс Эккель, Чак Эллисон; [Пер. с англ. Е. Матвеев]. – М.: Питер, 2004. – 608 с.
- 5) Липпман, Стенли Б. Язык программирования С++: вводный курс / Стенли Б. Липпман, Жози Лажойе, Барбара Му; [пер. с англ. и ред. В. А. Коваленко]. – 4-е изд. – М.: Вильямс, 2007. – 889 с.
- 6) Астахова, И.Ф. Язык С++ / И.Ф. Астахова, С.В. Власов, В.В. Фертиков, А.В. Ларин. – Минск: Новое знание, 2003. – 200 с.

Заочное отделение

Заочное отделение

Кафедра систем автоматизированного проектирования и
управления

ПРОГРАММИРОВАНИЕ НА ЯЗЫКАХ ВЫСОКОГО УРОВНЯ.
КОНТРОЛЬНЫЕ РАБОТЫ

Учебное пособие
для студентов заочной формы обучения

Чистякова Тамара Балабековна
Антипин Роман Васильевич
Новожилова Инна Васильевна

Отпечатано с оригинал макета. Формат 60x90^{1/16}
Печ. л. 6,5. Тираж 100 экз. Заказ №

Государственное образовательное учреждение
высшего профессионального образования
Санкт-Петербургский государственный технологический институт
(технический университет), ИК «Синтез»

190013, г. Санкт-Петербург, Московский пр., д. 26

Заочное отделение

Заочное отделение