

Энгельсский технологический институт (филиал)  
федерального государственного бюджетного образовательного  
учреждения высшего образования  
«Саратовский государственный технический университет имени Гагарина  
Ю.А.»

Кафедра «Естественные и математические науки»

**Оценочные материалы по дисциплине**

**Б.1.1.9 Программирование**

направления подготовки

09.03.01 «Информатика и вычислительная техника»

профиль: «Программное обеспечение средств вычислительной техники  
и автоматизированных систем»

Энгельс 2024

## Перечень компетенций и уровни их сформированности по дисциплинам (модулям), практикам в процессе освоения ОПОП ВО

В процессе освоения образовательной программы у обучающегося в ходе изучения дисциплины «Программирование» должна сформироваться компетенция ОПК-8

Критерии определения сформированности компетенции на различных уровнях ее формирования

Индекс компетенции	Содержание компетенции
ОПК-8	Способен разрабатывать алгоритмы и программы, пригодные для практического применения

Код и наименование индикатора достижения компетенции	Виды занятий для формирования компетенции	Оценочные средства для оценки уровня сформированности компетенции
<b>ИД- 1</b> <small>опк-8</small> Применяет основы информатики и программирования для разработки алгоритмов и программ, пригодных для практического использования	лекции, практические работы, самостоятельная работа	устный опрос, решение задач, вопросы для проведения экзамена, отчет по практической работе

### Уровни освоения компетенции

Уровень освоения компетенции	Критерии оценивания
Продвинутый (отлично)	<p><b>Знает:</b> основные принципы структурного написания программ, конструкции языка высокого уровня и технологию создания программ, базовые средства языка и средства стандартных библиотек, технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах.</p> <p><b>Умеет:</b> определять требования к создаваемой программе, выбирать среду программирования, выбирать или разрабатывать алгоритм решения задачи, реализовывать программный код, выполнять отладку и проводить тестирование программы, ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы, работать с современными системами программирования, включая объектно-ориентированные.</p> <p><b>Владеет:</b> методикой применения средств языка высокого уровня для создания программного обеспечения, языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки</p>

	<p>программ не менее чем на одном из алгоритмических языков программирования высокого уровня.</p>
<p>Повышенный (хорошо)</p>	<p><b>Знает:</b> основные принципы структурного написания программ, конструкции языка высокого уровня и технологию создания программ, базовые средства языка и средства стандартных библиотек, технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ не менее чем в одном режиме.</p> <p><b>Умеет:</b> определять основные требования к создаваемой программе, выбирать среду программирования, выбирать или разрабатывать алгоритм решения задачи, реализовывать программный код, выполнять отладку и проводить основное тестирование программы, ставить задачу среднего уровня сложности и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы, работать с современными системами программирования, включая объектно-ориентированные.</p> <p><b>Владеет:</b> методикой применения средств языка высокого уровня для создания программного обеспечения, языками процедурного и объектно-ориентированного программирования, навыками разработки и отладки программ не менее чем на одном из алгоритмических языков программирования высокого уровня.</p>
<p>Пороговый (базовый) (удовлетворительно)</p>	<p><b>Знает:</b> конструкции языка высокого уровня и технологию создания программ, базовые средства языка и средства стандартных библиотек, технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах достаточные для решения типовой задачи.</p> <p><b>Умеет:</b> определять минимальные требования к создаваемой программе, выбирать алгоритм решения задачи, реализовывать программный код, частично выполнять отладку, использовать прикладные системы программирования, разрабатывать основные программные документы, работать с одной из современных систем программирования, включая объектно-ориентированную.</p> <p><b>Владеет:</b> не менее чем одним из алгоритмических языков программирования высокого уровня, навыками разработки и отладки программ достаточными для решения типовой задачи.</p>

## 2. Методические, оценочные материалы и средства, определяющие процедуры оценивания сформированности компетенций (элементов компетенций) в процессе освоения ОПОП ВО

### 2.1 Оценочные средства для текущего контроля

## **Вопросы для устного опроса**

### **Раздел I. Введение.**

#### **Тема 1. Жизненный цикл программного обеспечения.**

1. Постановка задачи и спецификация программы, способы записи алгоритма.
2. Программа на языке высокого уровня.
3. Понятия тестирования и отладки.
4. Критерии качества программы.
5. Диалоговые программы, дружелюбность интерфейса.
6. Стил программирования.
7. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма.
8. Типовые алгоритмы вычислительных процессов

#### **Тема 2. Общая характеристика системы MSDN.**

1. Понятия системы программирования и среды программирования.
2. Выполнение программы в .NET. Структура платформы .NET.
3. Интегрированная среда разработки MS Visual Studio.NET.
4. Обобщенная система типов.

### **Раздел II. Процедурное программирование в C++**

#### **Тема 1. Состав языка программирования высокого уровня.**

1. Структура программы на языке высокого уровня.
2. Алфавит языка. Лексемы.
3. Понятие типа данных. Классификация типов. Значимые типы и ссылочные типы данных.
4. Классификация по способу хранения. Встроенные типы данных.
5. Переменные. Выражения. Основные операции. Использование стандартных функций.

## **6. Приведение типов.**

### **Тема 2. Основные операторы.**

1. Операторы и основные управляющие структуры: итерация, ветвление, повторение.
2. Суперпозиция структур.
3. Рекуррентные вычисления.

### **Тема 3. Процедурно-ориентированная технология разработки программ.**

1. Понятие функции. Спецификатор `static`. Фактические и формальные параметры функции.
2. Способы передачи параметров в функцию (по значению, по ссылке, выходные параметры, параметры-массивы).

## **Раздел III. Структуры данных C++**

### **Тема 1. Массивы, строки.**

1. Массивы в C++. Многомерные массивы. Рваные массивы.
2. Цикл `foreach` для перебора массивов.
3. Статические элементы класса `Array` для работы с массивами.
4. Типы задач по обработке массивов.
5. Простые методы сортировки.
6. Поисковые задачи.
7. Двумерные массивы.

### **Тема 2. Динамические структуры данных (списки, бинарные деревья)**

1. Понятие динамической структуры данных.
2. Структура список.

3. Структура дерева.

### **Тема 3. Файлы.**

1. Понятие потока. Классификация потоков.
2. Основные классы для работы с файловыми потоками.
3. Потоки байтов FileStream.
4. Потоки символов (StreamWriter и StreamReader).
5. Двоичные файлы (BinaryWriter, BinaryReader).

### **Тема 4. Визуальное проектирование Windows приложений.**

1. Основные особенности Windows. Типы программ в ОС Windows.
2. Сообщения. Структура приложения с обработкой сообщений.
3. Шаблон Windows – приложения. Основные типы Windows.Forms.
4. Элементы управления. Диалоговые окна.

## **Раздел IV. Объектно-ориентированное программирование C++**

### **Тема 1. Классы, объекты, методы. Перегрузка методов.**

1. Определение класса. Поля. Методы. Объекты.
2. Конструкторы. Деструктор. Сбор мусора.
3. Ключевое слово this. Ключевое слово static. Статические конструкторы.
4. Статические классы.
5. Перегрузка операций (унарные и бинарные операции, приведение типов).

### **Тема 2. Основные свойства ООП: Инкапсуляция.**

1. Инкапсуляция. Спецификаторы доступа. Свойства.
2. Автоматические свойства. Индексаторы.

### **Тема 3. Основные свойства ООП: наследование, полиморфизм.**

1. Иерархии классов. Наследование.
2. Виртуальные методы. Механизм позднего связывания.
3. Абстрактные классы. Класс object.
4. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET.

### **Тема 4. Абстрактные типы данных. Коллекции.**

1. Основные абстрактные структуры данных.
2. Физическое представление данных. Массив, список, двоичное дерево, хеш-таблица, стек, очередь
3. Пространство имен System.Collections. Интерфейсы. Коллекции общего назначения.
4. Основные элементы класса ArrayList. Основные элементы класса Hashtable. Основные элементы класса SortedList.
5. Основные элементы класса Stack. Основные элементы класса Queue.
6. Доступ к коллекциям с помощью нумератора.

### **Тема 5. Обобщенное программирование.**

1. Классы-прототипы.
2. Основные преимущества использования обобщений.
3. Ограничения.
4. Параметризованные коллекции библиотеки .NET
5. Создание собственных классов-прототипов.
6. Обобщенные методы.

## **Раздел V. Объектно-ориентированное программирование C#**

### **Тема 1. Введение**

1. Платформа Microsoft .Net Framework 2.0. Среда разработки Visual Studio .Net.

2. Компиляция и выполнение программы в среде CLR.

## **Тема 2. Основные понятия языка C#**

1. Технология объектно-ориентированного программирования.

2. Состав языка. Типы данных.

3. Переменные, именованные константы.

4. Ввод-вывод данных.

5. Форматирование данных.

## **Тема 3. Основные операции C#. Выражения. Преобразование типов.**

1. Основные операции C#.

2. Выражения.

3. Преобразование типов.

## **Тема 4. Операторы языка C#**

1. Блоки и пустые операторы, операторы ветвления (if, switch).

2. Операторы цикла (while, do, for).

3. Операторы передачи управления (goto, break, continue)

## **Тема 5 Методы: основные понятия**

Методы: основные понятия. Перегрузка методов.

## **Тема 6. Рекурсивные методы**

1. Рекурсия: прямая и косвенная, простая и «сложная».

2. Рекурсивные методы возвращающие и не возвращающие значение.



### **Тема. 7.Обработка исключений**

1. Обработка исключений: операторыtry,checkedиunchecked.
2. Генерация собственных исключений

### **Тема. 8.Массивы**

1. Массивы: одномерные, многомерные и ступенчатые.

### **Тема. 9.Символы и строки**

2. Символьный тип char, неизменяемые строки string.
3. Изменяемые строки StringBuider.

### **Тема. 10.Регулярные выражения**

### **Тема. 11.Организация C#-системы ввода-вывода**

1. Организация C#-системы ввода-вывода: стандартные, байтовые, символьные и двоичные потоки данных

### **Тема. 12.Работа с файловой системой**

1. Работа с файловой системой: классы DirectoryInfoиFile и классы DirectoryInfoиFileInfo.

### **Тема. 13.Классы**

1. Классы: основные понятия, данные, методы, конструкторы, свойства.
2. Классы: деструкторы, индексаторы, операции класса, операции преобразования типов.

### **Тема. 14.Иерархия классов**

1. Механизм наследования. Использование защищенного доступа.
2. Наследование конструкторов. Многоуровневая иерархия.

3. Полиморфизм. Виртуальные методы.
4. Абстрактные методы и классы. Запрет наследования.

### **Тема. 15.Интерфейсы и структуры**

1. Пользовательские и стандартные интерфейсы.
2. Структуры.

### **Тема. 16.Коллекции пространства именSystem.Collection**

1. Классификация коллекций.
2. Коллекции общего назначения: стек, очередь, динамический массив, хеш-таблица.

### **Задания для выполнения практических работ**

#### **Арифметические операции в C++**

1. Распространённая задача: Дано четырехзначное число (к примеру 5678), вывести на экран в обратном порядке цифры из которых это число состоит. То есть мы должны увидеть на экране 8765. Подсказка: чтобы взять из числа отдельные цифры, надо применять деление по модулю на 10.

2. На сайте практически любого коммерческого банка, можно встретить так называемый Депозитный калькулятор, который позволяет людям, не желающим углубляться в формулы расчета процентных ставок, узнать какую прибыль они получат. Для этого им достаточно заполнить определённые поля, нажать на кнопку и увидеть результат. Это простая программа, которую уже сможет написать каждый из вас. Итак, задача: Пользователь вводит сумму депозита и количество месяцев хранения денег в банке. Необходимо провести расчет и показать на экран прибыль с депозита в месяц, за весь срок депозита, и общую сумму к выплате в конце срока. Валюта пусть будет – доллар США. Процентная ставка – 5% годовых.

Формула расчета процентов в месяц –  $\frac{\text{СуммаДепозита} * (\text{ПроцентнаяСтавка} / 100)}{\text{ДнейВГоду} * \text{ДнейВМесяце}}$ .

### **Операторы выбора if и else в C++**

1. Первая задача интересная и достаточно простая. Но код получится длинным. Эту задачу придумали наши преподаватели из компьютерной академии. Идея была такая – как бы заставить начинающих студентов помучиться с написанием кода, чтобы писанины было побольше ))) Задание: Пользователь вводит число от 1 до 9999 (сумму выдачи в банкомате). Необходимо вывести на экран словами введенную сумму и в конце написать название валюты с правильным окончанием. Например: 7431 – семь тысяч четыреста тридцать один доллар, 2149 – две тысячи сто сорок девять долларов, 15 – пятнадцать долларов, 3 – три доллара. Для решения этой задачи вам необходимо будет применять оператор % (остаток от деления).

2. Пользователь вводит порядковый номер пальца руки. Необходимо показать его название на экран.

3. Еще одна задача для самостоятельного решения. Необходимо написать программу, которая проверяет пользователя на знание таблицы умножения. Пользователь сам вводит два целых однозначных числа. Программа задаёт вопрос: результат умножения первого числа на второе. Пользователь должен ввести ответ и увидеть на экране правильно он ответил или нет. Если нет – показать еще и правильный результат.

### **Оператор switch**

1. Составить расписание на неделю. Пользователь вводит порядковый номер дня недели и у него на экране отображается, то, что запланировано на этот день.

2. Допустим, у пользователя есть нумерованный список станций метро Барселоны. Необходимо написать код, в котором будет реализован диалог с пользователем, а именно предложено ввести номер станции метро. После ввода номера станции, надо показать на экран её название и время в пути.

Если же станции с таким номером нет, сообщить об этом и предложить ввести номер снова.

3. Пользователю для расчета времени в пути надо вводить не номера станций, а буквы (A,B,C,D,E). Необходимо предусмотреть ввод пользователем и маленьких, и больших букв (избавиться от регистрозависимости). То есть если введено или D, или d – должен сработать один и тот же case.

### **Цикл for в C++**

1. Написать программу, которая будет показывать на экран квадрат числа, введенного пользователем. Пользователь должен сам решать – выйти из программы или продолжить ввод. (Подсказка – необходимо запустить бесконечный цикл, в котором предусмотреть его прерывание, при наступлении определённого условия).

2. В спортзал ежедневно приходит какое-то количество посетителей. Необходимо предложить пользователю ввести такие данные: сколько человек посетило спортзал за день, ввести возраст каждого посетителя и в итоге показать возраст самого старшего и самого молодого из них, а так же посчитать средний возраст посетителей.

3. На складе имеется определённое количество ящиков с яблоками (в нашем примере 15). Когда подъезжает машина для погрузки, попросить пользователя ввести, сколько ящиков загрузить в первую машину, во вторую и так далее, пока не закончатся ящики с яблоками. Предусмотреть тот случай, когда пользователь введёт количество ящиков больше, чем есть на складе.

### **Циклы while, do while, вложенные циклы**

1. Организовать непрерывный ввод чисел с клавиатуры, пока пользователь не введёт 0. После ввода нуля, показать на экран количество чисел, которые были введены, их общую сумму и среднее арифметическое. Подсказка: необходимо объявить переменную-счетчик, которая будет

считать количество введенных чисел, и переменную, которая будет накапливать общую сумму чисел.

2. Необходимо суммировать все нечётные целые числа в диапазоне, который введёт пользователь с клавиатуры.

3. Задача посложней. Нарисовать равнобедренный треугольник из символов ^. Высоту выбирает пользователь. Например: высота = 5, на экране



### **Одномерные массивы C++**

1. Создать массив типа `int` на 10 элементов и заполнить его случайными числами от 7 до 14. После заполнения перезаписать все числа, которые больше десяти: от хранимого значение отнять 10. Например в ячейке хранится число 12:  $12 - 10 = 2$ . Записать в эту ячейку 2 Записывая новое значение, используйте составные (комбинированные) операторы.

2. Заполнить массив из 50-ти элементов нечётными числами от 1 до 99. (используйте операцию остаток от деления, чтобы проверить число на чётность)

3. Объявить три массива. Первые два заполнить случайными значениями от 10 до 30. В элементы третьего массива записать сумму соответствующих элементов первых двух массивов. (в нулевую ячейку третьего – сумму нулевых ячеек первого и второго массивов и так далее). Затем найти среднее арифметическое элементов третьего массива, максимальное значение и минимальное значение, которые он хранит.

### **Двумерные массивы C++**

1. Объявить двумерный массив, заполнить целыми числами и показать на экран.

2) Объявить двумерный массив и заполнить его построчно с клавиатуры. После заполнения – показать заполненную матрицу на экран и посчитать сумму элементов отдельно в каждом столбце и каждой строке.

3) Заполнить двумерный массив случайными числами от 10 до 100. Посчитать сумму элементов отдельно в каждой строке и определить номер строки, в которой эта сумма максимальна.

### Функции C++

1. Объявить два целочисленных массива с разными размерами и написать функцию, которая заполняет их элементы значениями и показывает на экран. Функция должна принимать два параметра – массив и его размер.

2. Необходимо создать двумерный массив 5 x 5. Далее написать функцию, которая заполнит его случайными числами от 30 до 60. Создать еще две функции, которые находят максимальный и минимальный элементы этого двумерного массива.

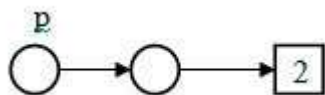
3. Написать игру в которой имитируется бросание кубиков компьютером и пользователем. В игре 2 кубика и на каждом из них может выпасть от 1 до 6 очков. Реализовать определение программой первого ходящего. Каждый делает по четыре броска. После бросков показать, нарисованные символами кубики и количество очков, выпавших на них. После пары бросков (бросок компьютера + бросок пользователя) выводить на экран промежуточный результат – количество набранных очков игроком и компьютером. В конце сообщить о том, кто выиграл по итогам всех бросков.

### Указатели C++

1. Дан указатель: `double **p = 0;`

Выполните следующие задания (решения можно оформлять внутри функции `main`):

- \* создайте конструкцию, изображенную на рисунке;
- \* выведите число, указанное в квадратике, на экран;
- \* после этого удалите все динамические объекты.



Опять же – ищем самое короткое решение.

2. Объявите указатель на массив типа `double` и предложите пользователю выбрать его размер. Далее напишите четыре функции: первая должна выделить память для массива, вторая – заполнить ячейки данными, третья – показать данные на экран, четвертая – освободить занимаемую память. Программа должна предлагать пользователю продолжать работу (создавать новые динамические массивы ) или выйти из программы.

3. Объявите указатель на массив типа `int` и выделите память для 12-ти элементов. Необходимо написать функцию, которая поменяет значения четных и нечетных ячеек массива. Например есть массив из 4-х элементов:

	ячейка 0	ячейка 1	ячейка 2	ячейка 3
<b>Исходные данные массива</b>	1	2	3	4

	ячейка 0	ячейка 1	ячейка 2	ячейка 3
<b>Данные после работы функции</b>	2	1	4	3

4. Объявить и заполнить двумерный динамический массив случайными числами от 10 до 50. Показать его на экран. Для заполнения и показа на экран написать отдельные функции. (подсказка: функции должны принимать три параметра – указатель на динамический массив, количество строк, количество столбцов). Количество строк и столбцов выбирает пользователь.

#### **Указатель на указатель**

1) Создать двумерный массив 5x5, заполнить его случайными числами от 10 до 99 и вывести на экран. Поменять местами максимальный элемент каждой строки с первым элементом в соответствующей строке. Задачу решить с помощью указателей.

2) Задача на умножение матриц. Пользователь произвольно задает размерность двух матриц и заполняет значениями вручную. Не забывайте,

что необходимо выделить соответствующие участки памяти для матриц, которые будет заполнять пользователь и для третьей итоговой матрицы. Далее отобразите на экране обе заполненные матрицы и итоговую матрицу (их произведение).

### **Классы C++**

1. Создать класс Tiles (кафель), который будет содержать поля с открытым доступом: brand, size\_h, size\_w, price и метод класса getData(). В главной функции объявить пару объектов класса и внести данные в поля. Затем отобразить их, вызвав метод getData().

2. Определить класс Children, который содержит такие поля (члены класса): закрытые – имя ребенка, фамилию и возраст, публичные – методы ввода данных и отображения их на экран. Объявить два объекта класса, внести данные и показать их.

3. Преобразовать строки двумерного массива в столбцы. Например было:

<b>11</b>	<b>22</b>
<b>33</b>	<b>44</b>
<b>55</b>	<b>66</b>
<b>77</b>	<b>88</b>

стало:

<b>11</b>	<b>33</b>	<b>55</b>	<b>77</b>
<b>22</b>	<b>44</b>	<b>66</b>	<b>88</b>

Класс будет содержать два метода: один заполнит массив значениями, второй произведет замену значений строк на значения столбцов.

## **2.2 Оценочные средства для промежуточного контроля<sup>4</sup>**

### **Вопросы к экзамену**

1. Постановка задачи и спецификация программы, способы записи алгоритма.



2. Программа на языке высокого уровня.
3. Понятия тестирования и отладки.
4. Критерии качества программы.
5. Диалоговые программы, дружелюбность интерфейса.
6. Стиль программирования.
7. Понятие алгоритма. Свойства алгоритма. Способы записи алгоритма.
8. Типовые алгоритмы вычислительных процессов
9. Понятия системы программирования и среды программирования.
10. Выполнение программы в .NET. Структура платформы .NET.
11. Интегрированная среда разработки MS Visual Studio.NET.
- 12.** Обобщенная система типов.
13. Структура программы на языке высокого уровня.
14. Алфавит языка. Лексемы.
15. Понятие типа данных. Классификация типов. Значимые типы и ссылочные типы данных.
16. Классификация по способу хранения. Встроенные типы данных.
17. Переменные. Выражения. Основные операции. Использование стандартных функций.
- 18.** Приведение типов.
19. Операторы и основные управляющие структуры: итерация, ветвление, повторение.
20. Суперпозиция структур.
21. Рекуррентные вычисления.
22. Понятие функции. Спецификатор `static`. Фактические и формальные параметры функции.
23. Способы передачи параметров в функцию (по значению, по ссылке, выходные параметры, параметры-массивы).
24. Массивы в C++. Многомерные массивы. Рваные массивы.
25. Цикл `foreach` для перебора массивов.

26. Статические элементы класса Array для работы с массивами.
27. Типы задач по обработке массивов.
28. Простые методы сортировки.
29. Поисковые задачи.
30. Двумерные массивы.
31. Понятие динамической структуры данных.
32. Структура список.
33. Структура дерево.
34. Понятие потока. Классификация потоков.
35. Основные классы для работы с файловыми потоками.
36. Потоки байтов FileStream.
37. Потоки символов (StreamWriter и StreamReader).
- 38.** Двоичные файлы (BinaryWriter, BinaryReader).
39. Основные особенности Windows. Типы программ в ОС Windows.
40. Сообщения. Структура приложения с обработкой сообщений.
41. Шаблон Windows – приложения. Основные типы Windows.Forms.
42. Элементы управления. Диалоговые окна.
43. Определение класса. Поля. Методы. Объекты.
44. Конструкторы. Деструктор. Сбор мусора.
45. Ключевое слово this. Ключевое слово static. Статические конструкторы.
46. Статические классы.
- 47.** Перегрузка операций (унарные и бинарные операции, приведение типов).
48. Инкапсуляция. Спецификаторы доступа. Свойства.
- 49.** Автоматические свойства. Индексаторы.
50. Иерархии классов. Наследование.
51. Виртуальные методы. Механизм позднего связывания.
52. Абстрактные классы. Класс object.

53. Интерфейсы. Работа с объектами через стандартные интерфейсы .NET.
54. Основные абстрактные структуры данных.
55. Физическое представление данных. Массив, список, двоичное дерево, хеш-таблица, стек, очередь
56. Пространство имен System.Collections. Интерфейсы. Коллекции общего назначения.
57. Основные элементы класса ArrayList. Основные элементы класса Hashtable. Основные элементы класса SortedList.
58. Основные элементы класса Stack. Основные элементы класса Queue.
59. Доступ к коллекциям с помощью нумератора.
60. Классы-прототипы.
61. Основные преимущества использования обобщений.
62. Ограничения.
63. Параметризованные коллекции библиотеки .NET
64. Создание собственных классов-прототипов.
65. Обобщенные методы.

## **Экзаменационные билеты.**

### **Билет 1**

1. Язык Си++. Понятие функции. Описание функций. Передача параметров. Возвращаемое значение.
2. Список, как линейно упорядоченная структура данных последовательного доступа. Одно- и двусвязные списки. Алгоритмы поиска, вставки и удаления элемента.
3. Простой способ шифровки текста, состоящего из строчных латинских букв и знаков препинания, состоит в замене каждой буквы на букву с заданным циклическим сдвигом  $n$  (если сдвиг 1, то "a" заменяется на "b", "b" на "c", "z" на "a"; если сдвиг 2, то "a" заменяется на "c", "y" на "a", "z" на "b"). Написать

функцию зашифровки и расшифровки текста. Исходный текст берется из файла, результаты помещаются в другой файл.

#### **Билет 2**

1. Понятие поиска в массиве. Линейный и бинарный поиск. Оценка числа операций.
2. Язык Си. Выражения. Порядок вычисления выражения. Понятие приоритета и ассоциативности операции.
3. Односвязный список задан указателем на первый элемент. Написать функцию, переставляющую элементы списка в обратном порядке.

#### **Билет 3**

1. Язык Си. Массивы. Многомерные массивы. Индексация многомерных массивов. Распределение памяти в многомерных массивах. Связь понятия указателя и массива. Инициализаторы массивов.
2. Язык Си. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
3. Многочлен от двух переменных задается в виде матрицы вещественных коэффициентов, в позиции  $(i,j)$  стоит коэффициент при  $x_i u_j$ . Написать функцию, умножающую многочлены, заданные таким образом.

#### **Билет 4**

1. Язык Си. Препроцессор. Включаемые файлы. Условная компиляция.
2. Язык Си. Представление целых чисел в памяти. Целочисленная машинная арифметика. Дополнительный код.
3. Дана непустая последовательность слов, составленных из строчных латинских букв; между соседними словами пробел, в конце - точка. Написать функцию, которая печатает в алфавитном порядке все согласные буквы, которые входят только в одно единственное слово.

#### **Билет 5**

1. Язык Си. Препроцессор. Макроопределения.
2. Язык Си. Динамически распределяемая память. Функции работы с динамически распределяемой памятью.
3. Дан массив целых чисел. Поменять местами пары элементов в массиве: 1 и 2, 3 и 4, 5 и 6 и т. д.

#### **Билет 6**

1. Язык Си. Понятие времени жизни и области видимости переменных. Глобальные и локальные переменные. Модификаторы области видимости и времени жизни.
2. Язык Си. Понятие указателя. Операции над указателями.
3. Определить, является ли строка палиндромом

#### **Билет 7**

1. Язык Си. Выражения. Порядок вычисления выражения. Понятие приоритета и ассоциативности операции.
2. Язык Си. Массивы. Многомерные массивы. Индексация многомерных массивов. Распределение памяти в многомерных массивах. Связь понятия указателя и массива. Инициализаторы массивов.
3. Найти количество цифр в строке символов, используя функции.

#### **Билет 8**

1. Язык Си. Структуры. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
2. Язык Си. Препроцессор. Включаемые файлы. Условная компиляция.
3. Написать функцию для перевода числа из одной позиционной системы счисления в другую. Цифры, большие 9, обозначить латинскими буквами A, B, C,... Основания систем счисления не больше 30.

#### **Билет 9**

1. Язык Си. Понятие функции. Описание функций в языке Си. Передача параметров. Возвращаемое значение.
2. Язык Си. Выражения. Порядок вычисления выражения. Понятие приоритета и ассоциативности операции.
3. Время суток представлено в виде структуры, содержащей информацию о часе, минутах и секундах. Написать функцию, которая увеличивает значение времени на n секунд (после 23:59:59 идет 00:00:00)

#### **Билет 10**

1. Язык Си. Функции с переменным числом параметров. Передача и получение необязательных параметров.
2. Язык Си. Операции над целыми числами. Преобразования между целочисленными типами языка Си.

3. Простое число называется числом Мерсенна, если оно может быть представлено в виде  $2^p - 1$ , где  $p$  — тоже простое число. Написать функцию, печатающую все числа Мерсена, меньшие данного  $n$ .

#### **Билет 11**

1. Язык Си. Стандартные функции форматированного ввода-вывода. Форматная строка (целые знаковые и беззнаковые в десятичном и шестнадцатеричном виде, числа с плавающей точкой, символы, строки). Возвращаемое значение.
2. Язык Си. Массивы. Многомерные массивы. Индексация многомерных массивов. Распределение памяти в многомерных массивах. Связь понятия указателя и массива. Инициализаторы массивов.
3. Дана последовательность целых чисел  $a_0, a_1, a_2, \dots$  длины не менее 3. Тройка  $a_i, a_{i+1}, a_{i+2}$  может иметь следующие комбинации знаков чисел ее составляющих:

+++ , ++- , +-+ , +-- , -++ , -+- , --+ , ---.

Написать функцию, печатающую число троек  $\langle a_0, a_1, a_2 \rangle, \langle a_1, a_2, a_3 \rangle$  и т.д., имеющих каждую из перечисленных комбинаций знаков

#### **Билет 12**

1. Язык Си. Строки. Инициализаторы строк. Стандартные функции работы со строками: определение длины строки, копирование строк, слияние строк.
2. Язык Си. Понятие функции. Описание функций. Передача параметров. Возвращаемое значение.
3. Написать функцию для печати распределения слов во входном тексте по их длине (сколько слов из одной буквы, сколько из двух и т.д.). Слова состоят из строчных латинских букв и отделяются разделителями или пробелами

#### **Билет 13**

1. Язык Си. Понятие указателя. Операции над указателями.
2. Язык Си. Структуры. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
3. Удалить из текстового файла все четные строки

#### **Билет 14**

1. Язык Си. Стандартные функции для работы файлами `fopen`, `fclose`, `feof`, `fprintf`, `fscanf`, `fread`, `fwrite`. Файловые переменные `stdin`, `stdout`, `stderr`.

2. Понятие сортировки. Алгоритмы «простых включений», «простого выбора», «пузырька». Оценка числа операций.
3. Добавить порядковый номер в каждую строку текстового файла.

#### **Билет 15**

1. Поиск подстроки в строке.
2. Язык Си. Массивы. Многомерные массивы. Индексация многомерных массивов. Распределение памяти в многомерных массивах. Связь понятия указателя и массива. Инициализаторы массивов.
3. Написать функцию, подсчитывающая в файле целых чисел количество элементов равных минимальному (файл просмотреть один раз).

#### **Билет 16**

1. Язык Си. Понятие времени жизни и области видимости переменных. Глобальные и локальные переменные. Модификаторы области видимости и времени жизни.
2. Язык Си. Строки. Инициализаторы строк. Стандартные функции работы со строками: определение длины строки, копирование строк, слияние строк.
3. Написать функцию, которая объединяет два упорядоченных списка целых чисел в один упорядоченный список.

#### **Билет 17**

1. Язык Си. Представление целых чисел в памяти. Целочисленная машинная арифметика. Дополнительный код.
2. Язык Си. Стандартные функции форматированного ввода-вывода. Форматная строка (целые знаковые и беззнаковые в десятичном и шестнадцатеричном виде, числа с плавающей точкой, символы, строки). Возвращаемое значение.
3. Имеется файл, содержащий информацию о деталях на складе. Для каждой детали задано название (не более 10 символов), код (целое число), и количество (целое число). Файл упорядочен по возрастанию кода. Имеется файл-заявка с аналогичной структурой. Написать функцию, печатающую названия и количество деталей, которых нет или не хватает для удовлетворения заявки.

#### **Билет 18**

1. Понятие сортировки. Пирамидальная сортировка. Оценка числа операций.

2. Язык Си. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
3. Дана матрица  $A$  целых чисел. Написать функцию, которая отметит единицами в массиве  $B$  все симметричные строки  $A$  и нулями все несимметричные строки  $A$ .

#### **Билет 19**

1. Язык Си. Массивы. Многомерные массивы. Индексация многомерных массивов. Распределение памяти в многомерных массивах. Связь понятия указателя и массива. Инициализаторы массивов.
2. Язык Си. Понятие указателя. Операции над указателями.
3. Анкета студента содержит: фамилию, номер группы и набор слушаемых курсов лекций (не более 10). Написать функцию, считывающую анкеты из одного файла и записывающую в другой файл списки фамилий студентов, слушающих каждый курс лекций, разделяя их пустыми строками

#### **Билет 20**

1. Стек и очередь. Сравнение реализации основных операций с помощью массива и с помощью структур в динамически распределяемой памяти.
2. Язык Си. Препроцессор. Макроопределения.
3. Дана последовательность вещественных чисел. Написать функцию, которая строит по этой последовательности два односвязных списка, первый из которых содержит все неповторяющиеся положительные числа последовательности, а второй -- все неповторяющиеся отрицательные

#### **Билет 21**

1. Язык Си. Строки. Инициализаторы строк. Функции работы со строками: определение длины строки, копирование строк, слияние строк.
2. Список, как линейно упорядоченная структура данных последовательного доступа. Одно- и двусвязные списки. Алгоритмы поиска, вставки и удаления элемента.
3. Анкета студента содержит: фамилию, номер группы и набор слушаемых курсов лекций (не более 10). Написать функцию, считывающую анкеты из одного файла и записывающую в другой файл все пары студентов, которые не слушают одинаковые лекции

#### **Билет 22**



1. Язык Си. Нормализованное представление вещественного числа. Мантисса и порядок. Разрядность мантиссы и порядка для стандартных типов float и double.
2. Язык Си. Понятие указателя. Операции над указателями.
3. Написать функцию, проверяющую, является ли строка арифметическим выражением, составленным из однобуквенных идентификаторов, скобок и двуместных операций + и -.

#### **Билет 23**

1. Язык Си. Структуры. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
2. Метод динамического программирования.
3. Дано произвольное целое число K и массив. Написать функцию, перемещающую каждый элемент на K позиций вперед, считая, что за последним элементом массива снова идет 0-й. Каждый элемент разрешается перемещать только один раз. Использовать дополнительный массив нельзя

#### **Билет 24**

1. Язык Си. Понятие указателя. Операции над указателями.
2. Понятие сортировки. Алгоритмы «простых включений», «простого выбора», «пузырька». Оценка числа операций.
3. Написать функцию, которая сосчитает число пар рядом стоящих единиц в двоичной записи целого числа без знака.

#### **Билет 25**

1. Язык Си. Понятие указателя. Операции над указателями.
2. Понятие сортировки. Быстрая сортировка Хоара. Оценка числа операций.
3. Написать функцию, вычисляющую квадратный корень из данного числа, используя только операции +, /, \*

#### **Билет 26**

1. Язык Си. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
2. Язык Си. Динамически распределяемая память. Стандартные функции работы с динамически распределяемой памятью.
3. Дан массив целых чисел без знака. Написать функцию, которая найдет в массиве число, в двоичной записи которого больше всего единиц

### **Билет 27**

1. Язык Си. Массивы. Многомерные массивы. Индексация многомерных массивов. Распределение памяти в многомерных массивах. Связь понятия указателя и массива. Инициализаторы массивов.
2. Язык Си. Представление целых чисел в памяти. Целочисленная машинная арифметика. Дополнительный код.
3. Написать функцию, которая изменит порядок слов в строке на обратный. Слова разделяются пробелами

### **Билет 28**

1. Язык Си. Понятие указателя. Операции над указателями.
2. Язык Си. Нормализованное представление вещественного числа. Мантисса и порядок. Разрядность мантиссы и порядка для стандартных типов float и double.
3. Даны три числа a, b, c. Положительные и четные числа разделить на минимальный среди a и b; Отрицательные и нечетные разделить на минимальный среди c и b; остальные –разделить на среднее арифметическое положительных. Написать подпрограмму для нахождения минимума из 2 - х чисел.

### **Билет 29**

1. Язык Си. Синтаксис описания структур. Обращение к полям структур для объектов и к полям по указателю на объект типа структура. Инициализатор структур.
2. Язык Си. Понятие указателя. Операции над указателями.
3. Дано число X ( $1 \leq X \leq 100$ ). Требуется перевести это число в римскую систему счисления.

### **Билет 30**

1. Язык Си. Стандартные функции форматированного ввода-вывода. Форматная строка (целые знаковые и беззнаковые в десятичном и шестнадцатеричном виде, числа с плавающей точкой, символы, строки). Возвращаемое значение.
2. Понятие сортировки. Пирамидальная сортировка, оценка числа операций.
3. Перевести целое десятичное число в римскую систему счисления. Число получить с клавиатуры

Оценивание результатов обучения в форме уровня сформированности элементов компетенций проводится путем контроля во время промежуточной аттестации в форме экзамена:

а) оценка «отлично» – компетенция(и) или ее часть(и) сформированы полностью на продвинутом уровне;

б) оценка «хорошо» – компетенция(и) или ее часть(и) сформированы на повышенном уровне;

в) оценка «удовлетворительно» - компетенция(и) или ее часть(и) сформированы на пороговом уровне;

г) оценка «неудовлетворительно» - компетенция(и) или ее часть(и) не сформированы.

Критерии, на основе которых выставляются оценки при проведении текущего контроля и промежуточной аттестации приведены в табл. 1.

Оценка «неудовлетворительно» ставятся также в случаях, если обучающийся не приступал к выполнению задания, а также при обнаружении следующих нарушений:

- списывание;
- плагиат;
- фальсификация данных и результатов работы.

Таблица 1 – Критерии выставления оценок при проведении текущего контроля и промежуточной аттестации

Шкала оценки	Оценка	Критерий выставления оценки
Четырехбалльная шкала	Отлично	Обучающийся ответил на все теоретические вопросы. Показал знания в рамках учебного материала, в том числе и по заданиям СРС. Выполнил практические и лабораторные задания. Показал высокий уровень умения и владения навыками применения полученных знаний и умений при решении задач в расширенных рамках учебного материала.
	хорошо	Обучающийся ответил на большую часть теоретических вопросов. Показал знания в узких рамках учебного материала. Выполнил практические и лабораторные задания с допустимой погрешностью. Показал хороший уровень умения и владения навыками применения полученных знаний и умений при решении задач в рамках учебного материала.
	удовлетворительно	Обучающиеся при ответе на теоретические вопросы и при выполнении практических заданий и лабораторных работ, продемонстрировал низкий уровень знаний и умений при решении задач в рамках

		учебного материала. При ответах на дополнительные вопросы были допущены неправильные ответы
	неудовлетворительно	Обучающиеся при ответе на теоретические вопросы и при выполнении практических заданий и лабораторных работ, продемонстрировал крайне низкий уровень знаний и умений при решении задач в рамках учебного материала. При ответах на дополнительные вопросы было допущено множество неправильных ответов