Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Первый проректор –
проректор по учебной работе
МГТУ им. Н.Э. Баумана
_____ Б.В. Падалкин
«26» апреля 2024 г.

Дополнительное профессиональное образование

ДОПОЛНИТЕЛЬНАЯ ПРОФЕССИОНАЛЬНАЯ ПРОГРАММА
ПРОГРАММА ПОВЫШЕНИЯ КВАЛИФИКАЦИИ
«Программирование на С (базовый уровень)»

Регистрац. №05.21.2303.30

Москва, 2024

АВТОРЫ ПРОГРАММЫ:

Преподаватель _____ В.В. Терехов

СОГЛАСОВАНО:

Начальник УСП _____ Т.А. Гузева

Директор
Центра дополнительного образования _____ М.В. Стоянова

Оглавление

1. ОБЩАЯ ХАРАКТЕРИСТИКА ДПП

Программа подготовлена на основе:

- Федерального закона от 29 декабря 2012 года № 273-ФЗ «Об образовании в Российской Федерации»;
- требований Приказа Минобрнауки России от 01.07.2013 года № 499 «Об утверждении Порядка организации и осуществления образовательной деятельности по дополнительным профессиональным программам»;
- методических рекомендаций-разъяснений Минобрнауки России по разработке дополнительных профессиональных программ на основе профессиональных стандартов от 22 апреля 2015 года № ВК-1030/06.

Реализация программы ДПП направлена на получение новой компетенции, необходимой для профессиональной деятельности.

1.1. Цель ДПП

Сформировать у обучающихся знания, навыки и умения в области разработки, отладки, проверки работоспособности, модификации компьютерного программного обеспечения.

1.2. Планируемые результаты обучения

Планируемые результаты обучения по ДПП:

- освоение профессиональных компетенций в процессе изучения перечисленных тем в учебном плане;
- успешное освоение программы повышения квалификации;
- успешное прохождение итоговой аттестации (зачет).

Обучающимся, успешно прошедшим обучение, выполнившим текущие контрольные задания и выдержавшим предусмотренное учебным планом зачет, выдается удостозерение о повышении квалификации по ДПП «Программирование на С (базовый уровень)».

1.3. Дополнительные характеристики ДПП

Характеристики новой квалификации определены в приказе Минтруда России от 20.07.2022 №424н «Об утверждении профессионального стандарта «Программист»

Вид профессиональной деятельности:

- Разработка компьютерного программного обеспечения (Код 06.001).

Трудовые функции:

- Проектирование компьютерного программного обеспечения (D/03.6).

1.4. Перечень профессиональных компетенций в рамках имеющейся квалификации, качественное изменение которых осуществляется в результате обучения

Получаемые компетенции базируются на основании Приказа Минобрнауки России от 19 сентября 2017 г. № 929 «Об утверждении федерального государственного образовательного стандарта высшего образования - бакалавриат по направлению подготовки 09.03.01 Информатика и вычислительная техника».

Перечень компетенций:

ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения.

1.5. Соответствие видов деятельности профессиональным компетенциям и их составляющих

Профессиональные компетенции	Практический опыт	Умения	Знания
Проектирование компьютерного программного обеспечения (D/03.6)			
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Проектирование структур данных; Проектирование баз данных; Проектирование программных интерфейсов	Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов	Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения

2. УЧЕБНЫЙ ПЛАН ДПП

2.1. Категория слушателей ДПП

Имеющаяся квалификация (требования к слушателям) – к освоению ДПП допускаются лица, имеющие среднее профессиональное и/или высшее образование.

2.2. Общая трудоёмкость программы, аудиторная и самостоятельная работа

Общая трудоемкость программы 70 академических часов, из них 46 академических часов аудиторной работы, 22 академических часа самостоятельной работы и 2 академических часов итоговой аттестации.

2.3. Форма обучения

Форма обучения по ДПП – очная с применением дистанционных образовательных технологий.

2.4. Учебный план

ДПП «Программирование на С (базовый уровень)» реализуется одним модулем.

№ п/п	Наименование темы, модуля	Форма контроля	Всего, час	В том числе			
				Лекции	Практ. занятия	Самост. работа	Итоговая аттестация
1.	Вступление	Практ. задание	10	4	2	4	-
2.	Основные понятия	Практ. задание	10	4	4	2	-
3.	Функции и управление потоком	Практ. задание	10	4	4	2	-
4.	Указатели	Практ. задание	10	4	4	2	-
5.	Структуры	Практ. задание	12	4	4	4	-
6.	Особенности С для МК и единицы трансляции	Практ. задание	16	4	4	8	-
7.	Итоговая аттестация	Зачет	2	-	-	-	2
	ИТОГО	-	70	24	22	22	2

3. КАЛЕНДАРНЫЙ УЧЕБНЫЙ ГРАФИК

№ п/п	Наименование темы, модуля	1 день	2 день	3 день	4 день	5 день	6 день	7 день	8 день
1	Вступление								
2	Основные понятия								
3	Функции и управление потоком								
4	Указатели								
5	Структуры								
6	Особенности С для МК и единицы трансляции								
7	Итоговая аттестация								Зачет

Минимальный срок освоения ДПП – 8 дней.

4. РАБОЧАЯ ПРОГРАММА ДПП

4.1. Рабочая программа модуля «Программирование на С (базовый уровень)»

4.1.1. Цель изучения модуля: сформировать у обучающихся знания, навыки и умения в области разработки, отладки, проверки работоспособности, модификации компьютерного программного обеспечения.

4.1.2. Задачи изучения модуля:

1. Знакомство с синтаксисом, семантикой и паттернами программирования на языке С
2. Получение практических навыков разработки консольных приложений на С
3. Получение теоретических и практических знаний по применению инструментов разработчика С
4. Знакомство с особенностями применения языка С для микроконтроллеров

4.1.3. Планируемые результаты обучения

Процесс изучения раздела направлен на формирование следующих компетенций

Код компетенции	Перечень планируемых результатов обучения по модулю	Формы и методы обучения, способствующие формированию и развитию компетенции
ОПК-8	**Знать:** Типовые решения, библиотеки программных модулей, шаблоны, классы объектов, используемые при разработке компьютерного программного обеспечения. **Уметь:** Применять методы и средства проектирования компьютерного программного обеспечения, структур данных, баз данных, программных интерфейсов. **Владеть:** Проектирование структур данных; Проектирование баз данных; Проектирование программных интерфейсов.	Формы обучения: Фронтальная. Методы обучения: Лекция; Практическая работа; Самостоятельная работа.

4.1.4 Содержание курса

Тема 1. Вступление (10 часа)

Лекции (4 часа). Краткая история, предназначение и ниша языка С++. Демонстрация примера «Hello, world». Описание инструментов разработчика. Процесс сборки программ. Правила оформления кода. Где искать информацию.

Практические занятия (2 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (4 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Вступление	Инструменты разработчика	Проработка дополнительной литературы	Огнева, М.В. Программирование на языке C++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.	Практ. задание

Тема 2. Основные понятия (10 часов)

Лекции (4 часа). Переменные. Выражения. Декларация и инициализация. Точка входа. Аргументы функций и возврат значений. Директива препроцессора #include. Хранение данных в ОЗУ. Целочисленные переменные и литералы. Встроенные целочисленные типы. Целочисленные типы фиксированной длины. Инициализация переменных. Квалификатор const. Неявное приведение типов. Математические операторы. Оператор приведения типа. Оператор sizeof. Перечисления. Статические переменные.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Основные понятия	Переменные, выражения	Проработка дополнительной литературы	Огнева, М.В. Программирование на языке C++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.	Практ. задание

Тема 3. Функции и управление потоком (10 часов)

Лекции (4 часа). Функции и процедуры. Операторы if и switch. Циклы for, while и do-while. Области видимости. Логические операторы и операторы сравнения. Тернарный оператор. Оператор goto. Рекурсия.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Функции и управление потоком	Функции	Проработка дополнительной литературы	Огнева, М.В. Программирование на языке C++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.	Практ. задание

Тема 4. Указатели (10 часов)

Лекции (4 часа). Массивы. Строки. Указатели, типизированные и обобщённые. Нулевой указатель. Операции взятия адреса и разыменования. Арифметика указателей. Понятия стека и кучи. Указатели на функции. Передача функций как аргументов. Динамическое выделение памяти.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (2 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Указатели	Указатели	Проработка дополнительной литературы	Огнева, М.В. Программирование на языке C++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.	Практ. задание

Тема 5. Структуры (12 часов)

Лекции (4 часа). Структуры. Оператор «стрелка». Массивы нулевого размера в структурах. Реализация односвязного и двусвязного списков. Двоичное дерево, красно-чёрное дерево. Работа с файлами, чтение и запись.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (4 часа).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Структуры	Структуры	Проработка дополнительной литературы	Огнева, М.В. Программирование на языке C++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.	Практ. задание

Тема 6. Особенности С для МК и единицы трансляции (16 часов)

Лекции (4 часа). Поддержка систем счисления в С. Битовые поля в структурах. Выравнивание и упаковка структур. Объединения. Использование объединений и структур для доступа к регистрам. Логические побитовые операции. Квалификаторы volatile и

restrict. Спецификаторы static, extern, register. Понятие единицы трансляции. Заголовочные файлы. Защита от повторного включения заголовочных файлов. Макросых.

Практические занятия (4 часа). Ответы на вопросы. Разбор задания для самостоятельной работы.

Самостоятельная работа (8 часов).

Наименование темы	Дидактические единицы, вынесенные на самостоятельное изучение	Формы самостоятельной работы	Учебно-методическое обеспечение	Форма контроля
Особенности С для МК и единицы трансляции	Макросы	Проработка дополнительной литературы	Огнева, М.В. Программирование на языке С++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.	Практ. задание

4.1.5. Оценочное средство для текущего контроля (формулировка практических заданий):

Тема 1. Формулировка практического задания:

Настроить рабочее окружение, собрать тестовую программу. Инструментарий: WSL2/Linux, VS Code, gcc.

Критерии оценивания: наличие рабочего окружения, возможность сборки тестовой программы.

Тема 2. Формулировка практического задания:

Разработать программу для вывода таблицы основных типов данных С.

Критерии оценивания: соответствие программы заданию; оформление кода.

Тема 3. Формулировка практического задания:

Разработать программу для поиска среди вводимых пользователем чисел минимальных и максимальных значений. Распечатать среднее арифметическое всех введённых чисел.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

Тема 4. Формулировка практического задания:

Пользователь вводит матрицу чисел. Программа должна транспонировать матрицу (заменить столбцы на строки). Необходимо использование двумерных массивов. Предусмотреть защиту от неправильного ввода данных.

Базовый уровень: поддержка только квадратных матриц.

Продвинутый уровень: поддержка любых матриц.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

Тема 5. Формулировка практического задания:

Разработать программу для парсинга текстового файла, содержащего информацию о геометрических фигурах. Создать структуры для каждой из фигур. Вывести сводную информацию: количество каждого типа фигур, общую площадь, общий периметр.

Критерии оценивания: соответствие программы заданию; оформление кода; наличие защиты от неправильного ввода данных.

Тема 6. Формулировка практического задания:

Существует просмотрщик файлов в режиме hex xxd , работа которого выглядит следующим образом:

~> echo Hello, the beautiful world! > test

~> xxd test

00000000: 4865 6c6c 6f2c 2074 6865 2062 6561 7574 Hello, the beaut

00000010: 6966 756c 2077 6f72 6c64 210a   iful world!.

xxd выводит начальный адрес строки, затем 16 байтов информации в 16-ричном виде, а затем ASCII-представление этих 16 байтов.

На следующей строке всё повторяется. Необходимо разработать аналог xxd .

Требования к программе:

1. Необходимо не только печатать вывод программы на консоль, но и сохранять файл с тем же содержимым на накопитель.

2. Если в файле встречаются непечатаемые символы, их необходимо выводить на печать в соответствии с обозначениями, данными в таблице по следующей ссылке: https://ru.wikipedia.org/wiki/ASCII#/media/Файл:ASCII_Code_Chart.svg

3. Программа должна иметь аргумент, указывающий, сколько байтов необходимо пропустить с начала файла. Т. е. при вводе xxd -s 5 test программа должна начать обрабатывать файл так, как будто бы он начинался с символа запятой.

5. УСЛОВИЯ РЕАЛИЗАЦИИ ДПП

5.1. Организационные условия реализации ДПП

Наименование аудитории	Вид занятия	Наименование оборудования, программного обеспечения
Компьютерный класс/вебинар	Лекции	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code. Дополнительное ПО устанавливается по необходимости из репозиториев Linux.
Компьютерный класс/вебинар	Практические занятия	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code. Дополнительное ПО устанавливается по необходимости из репозиториев Linux.
Компьютерный класс/вебинар	Самостоятельная работа	ПК с ОС GNU/Linux, либо Windows 10/11 + WSL, редактор VS Code. Дополнительное ПО устанавливается по необходимости из репозиториев Linux.
Компьютерный класс/вебинар	Итоговая аттестация	ПК с ОС GNU/Linux, либо Windows 10/11 + Microsoft Office.

5.2. Педагогические условия реализации ДПП

Реализация программы обеспечивается преподавательским составом, удовлетворяющим следующим условиям:

- наличие высшего профессионального образования, соответствующее профилю программы, из числа штатных преподавателей, или привлеченных на условиях почасовой оплаты труда;

- значительный опыт практической деятельности в соответствующей сфере из числа штатных преподавателей или привлеченных на условиях почасовой оплаты труда

5.3. Учебно-методическое обеспечение ДПП

Основная литература:

1. Огнева, М.В. Программирование на языке С++: практический курс: учебное пособие для вузов / М.В. Огнева, Е.В. Кудрина. – Москва: Издательство Юрайт, 2024. – 335 с.
2. Гниденко, И.Г. Технологии и методы программирования: учебное пособие для вузов / И.Г. Гниденко, Ф.Ф. Павлов, Д.Ю. Федоров. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 248 с.
3. Кудрявцева, И.А. Программирование: теория типов: учебное пособие для вузов / И.А. Кудрявцева, М.В. Швецкий. – 2-е изд., перераб. и доп. – Москва: Издательство Юрайт, 2024. – 652 с.

Дополнительная литература:

1. Прешерн К. Язык С. Мастерство программирования. Принципы, практики и паттерны / пер. с англ. А. Н. Слинкина – М.: ДМК Пресс, 2023. – 300 с.

2. Дональд Кнут: Искусство программирования. Том 1. Основные алгоритмы/ пер. с англ. Тригуб С. Г., Гордиенко Ю. Г., Красиков И. В. – М.: Вильямс, 2019. – 720 с

Интернет-источники:

1. https://habr.com/ru/companies/pvs-studio/articles/798675/.
2. https://habr.com/ru/companies/otus/articles/800381/.
3. https://habr.com/ru/companies/otus/articles/801123/.
4. https://habr.com/ru/companies/pvs-studio/articles/794997/.
5. https://habr.com/ru/articles/794630/.

5.4. Методические рекомендации

ДПП построена по тематическому принципу, каждый раздел представляет собой логически завершенный материал.

Преподавание программы основано на личностно-ориентированной технологии образования, сочетающей два равноправных аспекта этого процесса: обучение и учение. Личностно-ориентированный подход развивается при участии слушателей в активной работе на практических занятиях. Личностно-ориентированный подход направлен, в первую очередь, на развитие индивидуальных способностей обучающихся, создание условий для развития творческой активности слушателя и разработке инновационных идей, а также на развитие самостоятельности мышления при решении учебных задач разными способами, нахождение рационального варианта решения, сравнения и оценки нескольких вариантов их решения и т.п. Это способствует формированию приемов умственной деятельности по восприятию новой информации, ее запоминанию и осознанию, созданию образов для сложных понятий и процессов, приобретению навыков поиска решений в условиях неопределенности.

Практические занятия проводятся для приобретения навыков решения практических задач в предметной области модуля. Задания, выполняемые на практических занятиях, выполняются с использованием активных и интерактивных методов обучения.

Самостоятельная работа слушателей предназначена для проработки дополнительной литературы. Результаты практических заданий слушателей учитываются на итоговой аттестации.

При изучении курса предусмотрены следующие методы организации и осуществления учебно-познавательной деятельности:

- объяснительно-иллюстративный метод;
- репродуктивный метод;
- частично-поисковый метод.

6. ФОРМЫ ИТОГОВОЙ АТТЕСТАЦИИ ДПП

Итоговая аттестация проводится в форме зачета для проверки сформированности компетенций, полученных в рамках ДПП.

Зачет проводится в формате тестирования и ответов на вопросы билета. Результатом зачета служат правильные ответы на вопросы билета.

По результатам итоговой аттестации обучающемуся выставляется оценка «ЗАЧТЕНО/НЕ ЗАЧТЕНО»:

Оценка «ЗАЧТЕНО» выставляется обучающемуся, который:

- правильно ответил не менее, чем на 60% вопросов теста;
- правильно ответил на вопрос билета;
- продемонстрировал необходимые систематизированные знания и достаточную степень владения принципами предметной области программы, понимание их особенностей и взаимосвязь между ними в течение всего срока обучения по ДПП.

Оценка «НЕ ЗАЧТЕНО» ставятся обучающемуся, который:

- правильно ответил менее, чем на 60% вопросов теста;
- неправильно ответил на вопрос билета;
- имеет крайне слабые теоретические и практические знания, обнаруживает неспособность к построению самостоятельных заключений.

7. ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ИТОГОВОЙ АТТЕСТАЦИИ

7.1. Паспорт комплекта оценочных средств

Предметы оценивания	Объекты оценивания	Показатели оценки
ОПК-8. Способен разрабатывать алгоритмы и программы, пригодные для практического применения	Ответы на вопросы	Правильные ответы на вопросы теста

7.2. Комплект оценочных средств

7.2.1. Темы для подготовки к зачету:

1. Правила оформления кода.
2. Инструменты разработчика.
3. Процесс сборки программ.
4. Переменные и выражения.
5. Работа с памятью.
6. Работа с функциями.
7. Рекурсивные функции.
8. Массивы.
9. Работа с указателями.
10. Понятия стека и кучи.
11. Динамическое выделение памяти.
12. Структуры.
13. Работа с файлами, чтение и запись.
14. Поддержка систем счисления в С.
15. Макросы.

7.2.2. Вопросы теста для проведения зачета:

1. Какой размер имеет тип long int ?

1. 2 байта

2. 4 байта

3. 8 байтов

4. зависит от архитектуры и платформы

2. Что не может содержать структура struct foo ?

1. указатели на функции

2. массив без указания размера в конце структуры (например, int a[];)

3. указатель на саму себя (struct foo* pfoo;)

4. саму себя (struct foo bar;)

3. Какому типу данных соответствует литерал 0x1000UL ?

1. int

2. unsigned int

3. long unsigned int

4. long long int

4. Чему будет равна переменная b ?

```
unsigned char a = 250;
unsigned char b = a + 10;
```

1. 4

2. 5

3. 260

4. код содержит ошибку (неопределённое поведение)

5. Сколько раз выполнится цикл?

```
unsigned int a = 0, b = 1;
do {
    b++;
} while (a);
```

1. ни одного раза

2. один раз

3. будет работать бесконечно

4. код содержит ошибку компиляции

6. Чем является переменная str ?

```
const char* str;
```

1. указателем на строку

2. изменяемым указателем на неизменяемую строку

3. неизменяемым указателем на изменяемую строку

4. неизменяемым указателем на неизменяемую строку

7. Какой логической операции соответствует оператор ^?

1. и

2. или

3. исключающее или

4. не

8. Чему равна переменная a = (2 | 4) | 5 ?

1. 3

2. 5

3. 7

4. 1

9. Двусвязный список это ...

1. структура данных, содержащая указатель на предыдущий элемент

2. структура данных, содержащая указатель на следующий элемент

3. структура данных, содержащая указатель на первый элемент

4. структура данных, содержащая указатель на предыдущий и следующий элементы

10. Какое их объявлений битового поля не является правильным, если sizeof (int) == 4 ?

1. signed int :5;

2. unsigned int :0;

3. unsigned int b :30;

4. signed int c :40;

11. Какая из операций допустима над битовыми полями?

1. взятие адреса (оператор &)

2. получение размера (оператор sizeof)

3. переполнение беззнакового битового поля

4. объявление битового поля вне структуры или объединения

12. Чему равна переменная val ?

```
union {
  struct {
    uint32_t field1 :8;
    uint32_t field2 :8;
```

```
        uint32_t field3 :8;
        uint32_t field4 :8;
    } s;
    uint32_t reg;
} u;
u.reg = 0xFFFF0000;
uint32_t val = u.s.field3;
```

1. 0x0
2. 0xFF
3. 0xF0
4. код содержит ошибку

13. Минимальная адресуемая единица информации это

1. бит
2. байт
3. ниббл
4. машинное слово

14. Переменная, объявленная со спецификатором static в функции

```
void foo(void) {
   static int a = 0;
   a++;
}
```

1. инициализируется один раз при старте программы и уничтожается при завершении программы
2. инициализируется при каждом вызове функции и уничтожается при завершении программы
3. инициализируется при каждом вызове функции и уничтожается при выходе из функции
4. инициализируется при первом вызове функции, повторный вызов функции приведёт к ошибке

15. Что такое единица трансляции?

1. любой файл с исходным кодом
2. любой заголовочный файл
3. файл с исходным кодом, прошедший этап препроцессинга при сборке программы

4. файл с исходным кодом, прошедший этап компиляции при сборке программы

16. В каком случае функция будет скомпилирована и слинкована без ошибок?

```
void foo(void) {
   extern int a;
a = 10;
}
```

1. в любом
2. если переменная a объявлена ранее в текущей единице трансляции
3. если переменная a объявлена в текущей или другой единице трансляции
4. если переменная a объявлена в другой функции

17. В каком случае включение одного заголовочного файла в две разные единицы трансляции приведёт к ошибкам сборки?

1. если заголовочный файл содержит прототипы функций
2. если заголовочный файл содержит объявления структур
3. если заголовочный файл содержит объявления перечислений или объединений
4. если заголовочный файл содержит нестатические переменные

18. Чему равно значение переменной a ? uint32_t* a = 0x1000; a += 2;

1. 0x1002
2. 0x1004
3. 0x1008
4. 0x100A

19. Что напечатает следующий код, если int val = 1 ?

```
switch(val) {
   case 1: printf("1\n");
   case 2: printf("2\n");
   case 3: printf("3\n");
   default: printf("4\n");
}
```

1. 1
2. 234
3. 123

4. 1234

20. Какая из строк кода содержит ошибку в использовании типа void ?

1. int foo(void);
2. void a;
3. void* a;
4. void foo(int a);

21. Для немедленного перехода к следующей итерации любого цикла используется оператор

1. break
2. continue
3. return
4. goto

22. Какой прототип должна иметь функция, вызываемая следующим образом:

```
int* ptr = NULL;
foo(&ptr);
```

1. void foo(int ptr)
2. void foo(int* ptr)
3. void foo(int** ptr)
4. void foo(int& ptr)

23. Чем будет равна переменная c ?

```
int a = 5, b = 10;
int c = (a > 4) ? ((b < 10) ? 100 : 200) : 300;
```

1. 100
2. 200
3. 300
4. ошибка компиляции

24. Чему будут равны значения массива int a[3] = { 1 } ?

1. a[0] == a[1] == a[2] == 1
2. a[0] == 1, a[1] и a[2] содержат мусор
3. a[0] == 1, a[1] == a[2] == 0

4. зависит от контекста

25. Чему равно значение arrSize ?

```
uint32_t arr[5] = {};
uint32_t* ptr = arr;
arrSize = sizeof(ptr);
```

1. заданному размеру массива (5)
2. сумме размеров всех элементов массива (sizeof(arr))
3. размеру типа массива (sizeof(uint32_t))
4. размеру обобщённого указателя (sizeof(void*))

26. Какая операция допустима в С?

1. многократное изменение одной переменной в пределах одной точки следования
2. разыменование нулевого указателя
3. переполнение беззнаковой переменной
4. попытка освободить уже освобождённую память (функцией free())

27. Каким способом можно вернуть из функции больше одного значения?

1. возвращать структуру, содержащую нужные значения
2. передать в функцию указатель на первое значение, а другое значение вернуть оператором return
3. передать в функцию указатели на все нужные значения, возвращаемый тип функции сделать void
4. всё вышеперечисленное

28. Квалификатор volatile нужен для

1. отключения оптимизаций компилятора при обращении к переменной
2. доступа к регистрам микроконтроллера
3. подсказки программисту, что переменная может быть изменена извне
4. ускорения работы кода с указанной переменной

29. Поддержки какой системы счисления нет в стандарте С11?

1. двоичной
2. восьмеричной
3. десятичной

4. шестнадцатеричной

30. В чём отличие функций выделения памяти malloc() и calloc() ?

1. выбор функции зависит от платформы
2. только в том, что malloc() принимает один аргумент, а calloc() — два
3. calloc() принимает два аргумента и очищает выделенную память
4. calloc() оптимизирована для микроконтроллеров

7.2.3. Вопросы для проведения зачета:

1. Чему будет равно значение переменных a , b и c ? Дайте ответ в шестнадцатеричной системе счисления.

   ```
   uint16_t a = ~0xFF00;
   uint32_t b = (1U << 3U) | 0x3;
   uint32_t c = (5 & 3) ^ 2;
   ```

2. Разработайте макрос, принимающий два аргумента и возвращающий больший из них?

3. Что напечатает данный код?

   ```
   #include <stdio.h>
   int a = 100;
   int foo(void) {
      static int a = 5;
      a *= 2;
      return a;
   }
   int main(void) {
      int res = foo();
      printf("%d\n", res);
   return 0; }
   ```

4. Что напечатает данный код?

   ```
   #include <stdio.h>
   #include <stbool.h>
   ```

```
bool foo(void) {
   printf("foo\n");
   return true;
}
bool bar(void) {
   printf("bar\n");
   return false;
}
bool baz(void) {
   printf("baz\n");
   return false;
}
int main(void) {
   if ((foo() || bar()) && baz())
      printf("main\n");
   return 0;
}
```

5. Что напечатает данный код?

```
#include <stdio.h>
void rec(int n) {
   if(n > 0)
      rec(n - 1);
   printf("%d\n", n);
}
int main(void) {
  rec(4);
return 0;
}?
```