

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Косенок Сергей Михайлович
Должность: ректор
Дата подписания: 11.06.2026 10:49:18
Уникальный программный ключ:
e3a68f3eaa1e62674b54f4998099d3d6bfdcf836

Тестовое задание для диагностического тестирования по дисциплине

Структурное программирование
Семестр 2

Код, направление подготовки	27.03.04 Управление в технических системах
Направленность (профиль)	Инженерия автоматизированных, информационных и робототехнических систем
Форма обучения	очная
Кафедра-разработчик	автоматики и компьютерных систем
Выпускающая кафедра	автоматики и компьютерных систем

№	Проверяемая компетенция	Тип вопроса	Задание	Варианты ответов	Тип сложности вопроса
1.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	Какая конструкция программирования не допускается (является крайне нежелательной) в структурном программировании?	<ol style="list-style-type: none"> 1. оператор перехода типа goto 2. рекурсия 3. глобальные переменные 4. множественное ветвление типа оператора switch 	низкий
2.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	Какое определение термина «указатель» является верным в языке С?	<ol style="list-style-type: none"> 1. Указателем называют специальную конструкцию в виде → для обращения к полям структуры 2. Указатель – это специальный тип данных, используемый для представления адресов объектов, размещенных в памяти компьютера 3. Указатель – это номер элемента массива, который указывается в квадратных скобках при обращении к нему 4. Указателями называют имена параметров функции, потому что они указывают на значения параметров 5. Указателями называют ссылки на объекты классов и других структурных типов 	низкий
3.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Все или ничего	Какие утверждения о формальных параметрах функций в языке С являются истинными?	<ol style="list-style-type: none"> 1. являются локальными переменными 2. являются глобальными переменными 3. являются значениями выражений, которые нельзя изменять 4. могут быть изменены в теле функции, если не являются константными 5. изменение формальных параметров влечет за собой изменение фактических параметров 	низкий

4.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Множественный выбор	Что из перечисленного относится к базовым управляющим (алгоритмическим) конструкциям структурного программирования?	<ol style="list-style-type: none"> 1. последовательность 2. выбор (условие) 3. повторение (цикл) 4. рекурсия 5. исключение 6. произвольный переход 7. структурные типы данных 	низкий
5.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Множественный выбор	Какие способы определения разработчиком новых типов данных предусмотрены в языке C?	<ol style="list-style-type: none"> 1. переименование типов 2. перечисления 3. структуры 4. объединения (смеси) 5. отрезки (диапазоны) 6. классы 7. строки 8. множества 	низкий
6.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Имеется следующее объявление:</p> <pre>enum Fruit { Apple, Orange=5, Kiwi, Pinapple=10, Lemon };</pre> <p>Чему будет равно выражение Kiwi+2?</p>	<ol style="list-style-type: none"> 1. 0 2. 8 3. 11 4. Lemon 5. Kiwi 6. это выражение приведет к ошибке 	средний

7.	ОПК-6.1 ОПК-6.2 ОПК-11.2	На соответств ие	<p>Укажите соответствующее ключевое слово языка C, которое образует конструкцию в соответствии с приведенным описанием:</p> <ul style="list-style-type: none"> – Позволяет поставить в соответствие описанию типа некоторый идентификатор, используемый в дальнейшем как имя этого типа _____ – Позволяет определить множество целочисленных констант с уникальными именами _____ – Позволяет описать тип данных как некоторый набор переменных, называемых полями и имеющих одинаковый или различные типы _____ – Позволяет описать тип, который использует одну область памяти для обращения к ней как к данным разных типов _____ 	<ol style="list-style-type: none"> 1. typedef 2. enum 3. struct 4. union 5. define 6. array 	средний
8.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Что будет напечатано при выполнении следующего фрагмента программы?</p> <pre>int a = 5; int *b = &a; printf("%d", a**b);</pre>	<ol style="list-style-type: none"> 1. «мусорное» значение 2. ничего из-за ошибки компиляции 3. произойдет ошибка во время выполнения из-за неправильного использования указателя 4. 25 5. 0 	средний
9.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Какое утверждение о размере структуры в байтах является верным в любых случаях?</p>	<ol style="list-style-type: none"> 1. равен размеру наибольшего из всех полей структуры 2. не меньше суммы размеров всех полей структуры 3. равен сумме размеров всех полей структуры 4. равен размеру типа int 	средний

10.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Вставить слово	<p>Как в приведенном коде правильно передать массив <i>a</i> в функцию <i>fill</i>?</p> <pre>void fill(_____) { ... } int main() { double a[10]; fill(_____); ... }</pre>	<ol style="list-style-type: none"> 1. double *a 2. double *a[] 3. int a[10] 4. a: array 5. a 6. *a 7. &a 8. a[] 9. a[10] 10. &a[10] 11. double a[10] 	средний
11.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Какое значение должна возвращать главная функция в случае успешного завершения?</p>	<ol style="list-style-type: none"> 1. 0 2. -1 3. любое ненулевое значение 4. ERRORLEVEL 5. главная функция не может возвращать значение 	средний
12.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Множественный выбор	<p>Какие утверждения об объединениях в языке C являются истинными?</p>	<ol style="list-style-type: none"> 1. объединение позволяет посредством разных его полей обратиться к одной области памяти как к данным разных типов 2. поля объединения размещаются в памяти непосредственно друг за другом 3. тэг объединения является самостоятельным именем типа данных 4. тэг объединения необходимо использовать совместно с ключевым словом union в качестве типа данных 5. тэг объединения необходимо использовать совместно с ключевым словом enum в качестве типа данных 6. размер объединения определяется наибольшим из размеров его полей 	средний

13.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	Какое значение возвращает функция <code>fopen</code> в случае, если попытка открытия потока оказалась неуспешной?	<ol style="list-style-type: none"> 1. NULL 2. EOF 3. SEEK_END 4. -1 5. любое ненулевое значение 	средний
14.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Для представления точки в трехмерной системе координат была определена структура <code>Point</code> с тремя полями <code>x</code>, <code>y</code>, <code>z</code>:</p> <pre>struct Point { double x, y, z; };</pre> <p>Как правильно определить массив из 30 таких структур?</p>	<ol style="list-style-type: none"> 1. <pre>struct Point { double x, y, z; }; struct Point a[30];</pre> 2. <pre>struct Point { double x[30]; double y[30]; double z[30]; }; struct Point a;</pre> 3. <pre>struct Point[30] { double x, y, z; }; struct Point a;</pre> 4. В языке C не допускается определение массива элементов структурного типа 	средний

15.	ОПК-6.1 ОПК-6.2 ОПК-11.2	На соответствие	<p>Какие функции стандартной библиотеки языка C выполняют следующие действия с буферизированными потоками ввода-вывода:</p> <ul style="list-style-type: none"> – чтение массива двоичных данных _____ – запись массива двоичных данных _____ – чтение данных в текстовом виде _____ – запись данных в виде текста _____ 	<ol style="list-style-type: none"> 1. fread 2. fwrite 3. fscanf 4. fprintf 5. fseek 6. ftell 7. feof 8. fopen 9. fclose 	средний
16.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Что означает приведенное ниже объявление? typedef int (*a);</p>	<ol style="list-style-type: none"> 1. объявляется тип, который представляет собой указатель на данные типа int 2. объявляется тип, который представляет собой указатель на функцию, возвращающую значение типа int 3. объявляется массив элементов типа int с заранее не заданным количеством элементов 4. объявляется указатель на переменную a типа int 5. объявляется функция, которая возвращает указатель на данные типа int 	высокий
17.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Что выполняет данная функция? void func(double *a, int n) { double * b, t; for (b = a+n-1; a < b; a++, b--) t = *a, *a = *b, *b = t; }</p>	<ol style="list-style-type: none"> 1. меняет порядок значений элементов массива на обратный 2. меняет местами значения в каждой паре соседних элементов массива 3. находит сумму значений элементов массива 4. невозможно сказать, так как функция реализована некорректно 5. код функции не скомпилируется из-за ошибок 	высокий

18.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Один из	<p>Что выведет на экран приведенная ниже функция?</p> <pre>void out(double *a, int n) { while (n--) printf("%lg\n", *a++); }</pre>	<ol style="list-style-type: none"> 1. расположенные «в столбик» n значений элементов массива a 2. расположенные «в столбик» $n-1$ значений элементов массива a 3. расположенные «в строку» $n-1$ значений элементов массива a 4. «мусорные» значения из-за неправильной работы с указателем 5. код функции не скомпилируется из-за ошибки 	высокий
19.	ОПК-6.1 ОПК-6.2 ОПК-11.2	Все или ничего	<p>Какие описания типа являются правильными в языке C?</p>	<ol style="list-style-type: none"> 1. <pre>struct Point { double x, y; };</pre> 2. <pre>typedef struct Point { double x; double y; } Point;</pre> 3. <pre>typedef tagPoint struct { double x, y; };</pre> 4. <pre>struct { double x; double y; } typedef Point;</pre> 	высокий

20.	ОПК-6.1 ОПК-6.2 ОПК-11.2	На соответств ие	<p>Укажите характеристики, соответствующие режимам открытия файла функцией fopen:</p> <p>– "r" _____(1–6) _____(7–8)</p> <p>– "w" _____(1–6) _____(7–8)</p> <p>– "a" _____(1–6) _____(7–8)</p> <p>– "r+" _____(1–6) _____(7–8)</p> <p>– "w+" _____(1–6) _____(7–8)</p> <p>– "a+" _____(1–6) _____(7–8)</p>	<ol style="list-style-type: none"> 1. только для чтения 2. только для записи 3. только для дополнения 4. чтение с возможностью записи 5. запись с возможностью чтения 6. дополнение с возможностью чтения 7. файл должен существовать 8. файл будет создан, если он не существует 	<p style="text-align: center;">ВЫСОКИЙ</p>
-----	--------------------------------	------------------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------