Оглавление

[2 тема 2](#_Toc41307077)

[3 тема 11](#_Toc41307078)

[4 тема 18](#_Toc41307079)

[5 тема 29](#_Toc41307080)

[6 тема 48](#_Toc41307081)

[7 тема 58](#_Toc41307082)

[8 тема 65](#_Toc41307083)

[9 тема 78](#_Toc41307084)

[10 тема 85](#_Toc41307085)

[13 тема 99](#_Toc41307086)

[12+14 темы 107](#_Toc41307087)

[15 тема 130](#_Toc41307088)

[16 тема 142](#_Toc41307089)

[20 тема 166](#_Toc41307090)

[21 тема 178](#_Toc41307091)

[22 тема 197](#_Toc41307092)

[23 тема 210](#_Toc41307093)

## 1 тема

**<p><b>Алгоритмы, в соответствии с которыми решение поставленных задач сводится к арифметическим действиям, называются</b></p>**

1. <p><b>разветвляющимися</b></p>

2. <p><b>линейными</b></p>

3. <p><b>численными</b></p>

Ответ: 3

Комментарий: AP00101

**<p><b>Система четких однозначных указаний, которая определяет последовательность действий над некоторыми объектами и после конечного числа шагов приводит к получению требуемого результата – это</b></p>**

1. <p><b>алгоритм</b></p>

2. <p><b>схема алгоритма</b></p>

3. <p><b>тело функции</b></p>

4. <p><b>программа</b></p>

Ответ: 1

Комментарий: AP00102

**<p><b>Какое свойство алгоритма означает, что исполнение алгоритма должно закончиться за конечное число шагов?</b></p>**

1. <p><b>точность</b></p>

2. <p><b>результативность</b></p>

3. <p><b>дискретность</b></p>

4. <p><b>массовость</b></p>

Ответ: 2

Комментарий: AP00103

**<p><b>Всякий алгоритм может быть</b></p>**

1. <p><b>записан на естественном языке </b></p>

2. <p><b>изображен в виде рисунка</b></p>

3. <p><b>записан на русском языке</b></p>

Ответ: 1

Комментарий: AP00104

**<p><b>Структурный подход базируется на основополагающих принципах:</b></p>**

1. <p><b>использование функционального стиля программирования</b></p>

2. <p><b>применении к программному буферу подпрограмму подсчета количества слов в очередной строке, накапливая в цикле общее количество слов в файле</b></p>

3. <p><b>последовательная декомпозиция алгоритма решения задачи сверху вниз</b></p>

4. <p><b>обнаружение и обработка ошибок при считывании данных</b></p>

Ответ: 3

Комментарий: AP00105

**<p><b>Алгоритмическую декомпозицию можно представить как</b></p>**

1. <p><b>разделение модулей, где каждый система выполняет один из <br>этапов общего процесса</b></p>

2. <p><b>разделение алгоритмов, где каждый модуль системы выполняет <br>один из этапов общего процесса</b></p>

3. <p><b>нет понятия "алгоритмическая декомпозиция"</b></p>

4. <p><b>проверить, определяя начало и окончание записи файла</b></p>

Ответ: 2

Комментарий: AP00106

**<p><b>Что представляет собой механизм «заглушек»?</b></p>**

1. <p><b>процедуры, имитирующие вход и выход функций нижнего уровня</b></p>

2. <p><b>процедуры, имитирующие вход или выход процедур нижнего уровня</b></p>

3. <p><b>процедуры, выполняющие некоторые действия</b></p>

4. <p><b>"заглушки" не являются частью программирования</b></p>

Ответ: 2

Комментарий: AP00107

**<p><b>Структурный подхода базируются на:</b></p>**

1. <p><b>принципе решения сложных проблем путем их разбиения на множество меньших независимых задач, легких для понимания и решения</b></p>

2. <p><b>принципе составления простого кода программы без вложенных циклов</b></p>

3. <p><b>принципе организации составных частей проблемы в иерархические древовидные </b></p>

Ответ: 1, 3

Комментарий: AP00108

**<p><b>Объектная декомпозиция имеет несколько преимуществ перед алгоритмической:</b></p>**

1. <p><b>объектная декомпозиция уменьшает размер программных систем за счет повторного использования общих механизмов, что приводит к существенной экономии выразительных средств</b></p>

2. <p><b>объектная декомпозиция помогает нам разобраться в сложной программной системе, предлагая нам разумные решения относительно выбора подпространства большого пространства состояний</b></p>

3. <p><b>объектно-ориентированные системы более гибки и проще эволюционируют со временем, потому что их схемы базируется на устойчивых промежуточных формах</b></p>

Ответ: 1, 2, 3

Комментарий: AP00109

**<p><b>Назовите основные достоинства объектно-ориентированного подхода к программированию</b></p>**

1. <p><b>упрощение процесса проектирования программных систем</b></p>

2. <p><b>объектно-ориентированный подход не является достоинством программирования</b></p>

3. <p><b>программы могут быть подготовлены на различных языках и объединены перед выполнением в единый модуль</b></p>

Ответ: 1

Комментарий: AP00110

**<p><b>Что предлагает объектно-ориентированный подход?</b></p>**

1. <p><b>описывать программные системы в виде подсистем</b></p>

2. <p><b>использование процедурного стиля программирования</b></p>

3. <p><b>отказаться от процедур и функций</b></p>

4. <p><b>Нет правильного ответа</b></p>

Ответ: 4

Комментарий: AP00111

**<p><b>Что представляет собой объект?</b></p>**

1. <p><b>это понятие, сочетающее в себе совокупность массивов и действий над ними</b></p>

2. <p><b>это экземпляр класса</b></p>

3. <p><b>переменная, тип которой задается пользователем</b></p>

4. <p><b>это понятие, сочетающее в себе только совокупность данных</b></p>

Ответ: 2

Комментарий: AP00112

**<p><b>Для чего служит свойство объектов наследование?</b></p>**

1. <p><b>позволяет описать объект с учетом наследования возможностей объекта</b></p>

2. <p><b>позволяет повторно использовать уже созданную часть программного кода в других проектах</b></p>

3. <p><b>служит для сокращения избыточности кода</b></p>

4. <p><b>определяет значение модульного построения программ</b></p>

Ответ: 1, 2, 3

Комментарий: AP00113

**<p><b>Что называется инкапсуляцией?</b></p>**

1. <p><b>объединение в одном месте всех данных и методов объекта</b></p>

2. <p><b>запись результата по одному из адресов</b></p>

3. <p><b>совокупность правил образования цепочек символов, образующих идентификаторы</b></p>

Ответ: 1

Комментарий: AP00114

**<p><b>Чем характеризуется полиморфизм?</b></p>**

1. <p><b>основывается на возможности включения в данные объекта</b></p>

2. <p><b>основывается на абстрактной информации </b></p>

3. <p><b>позволяет при использовании объекта вызывать определенное свойство или метод, не заботясь о том, как объект выполняет задачу</b></p>

Ответ: 1, 3

Комментарий: AP00115

**<p><b>Что содержит в себе термин «класс»?</b></p>**

1. <p><b>объединяет объекты с одинаковыми возможностями (данными и методами)</b></p>

2. <p><b>упорядоченная структура однотипных данных</b></p>

3. <p><b>описывает характеристику функции</b></p>

Ответ: 1

Комментарий: AP00116

**<p><b>Какие типы свойств компонентов различают?</b></p>**

1. <p><b>свойства компонентов времени</b></p>

2. <p><b>свойства компонент скорости</b></p>

3. <p><b>ничего из предложенного</b></p>

Ответ: 3

Комментарий: AP00117

**<p><b>Что представляет собой константа?</b></p>**

1. <p><b>информация, представленная в формализованном виде и пригодная для обработки алгоритмом</b></p>

2. <p><b>данные, известные перед выполнением алгоритма</b></p>

3. <p><b>данные, значения которых могут изменяться в процессе выполнения алгоритма</b></p>

4. <p><b>данные, значения которых не меняются в процессе выполнения алгоритма</b></p>

Ответ: 4

Комментарий: AP00118

**<p><b>Последовательность букв и цифр, начинающаяся с буквы - это ...</b></p>**

1. <p><b>идентификатор</b></p>

2. <p><b>тип данных</b></p>

3. <p><b>данные</b></p>

4. <p><b>массив</b></p>

Ответ: 1

Комментарий: AP00119

**<p><b>Какие числа должны удовлетворять условию -2<sup>n-1</sup> &lt; х &lt; 2<sup>n-1</sup>?</b></p>**

1. <p><b>float</b></p>

2. <p><b>char</b></p>

3. <p><b>ничего из предложенного</b></p>

Ответ: 3

Комментарий: AP00120

**<p><b>Какие операции можно применять к данным типа int?</b></p>**

1. <p><b>арифметические операции</b></p>

2. <p><b>логические</b></p>

3. <p><b>операция конкатенации</b></p>

4. <p><b>операции отношения или сравнения</b></p>

Ответ: 1

Комментарий: AP00121

**<p><b>Какие операции можно применять к данным типа real?</b></p>**

1. <p><b>логические</b></p>

2. <p><b>арифметические</b></p>

3. <p><b>операция конкатенации</b></p>

4. <p><b>такого типа данных в языке программирования С не существует</b></p>

Ответ: 4

Комментарий: AP00122

**<p><b>Какие операции можно применять к данным типа char?</b></p>**

1. <p><b>операции сравнения</b></p>

2. <p><b>операции отношения</b></p>

3. <p><b>арифметические</b></p>

4. <p><b>ничего из предложенного</b></p>

Ответ: 4

Комментарий: AP00123

**<p><b>К целочисленным типам относятся типы:</b></p>**

1. <p><b>char<br></b></p>

2. <p><b>int</b></p>

3. <p><b>short</b></p>

4. <p><b>long</b></p>

Ответ: 1, 2, 3, 4

Комментарий: AP00124

**<p><b>Имена целочисленных типов могут использоваться в сочетании с парой модификаторов типа: </b></p>**

1. <p><b>signed</b></p>

2. <p><b>sign</b></p>

3. <p><b>unsigned</b></p>

Ответ: 1, 3

Комментарий: AP00125

**<p><b>Что представляет собой массив?</b></p>**

1. <p><b>группа расположенных друг за другом в памяти элементов одного типа и имеющих одно общее имя</b></p>

2. <p><b>упорядоченную структуру однотипных данных</b></p>

3. <p><b>набор однотипных или разнотипных данных, с которыми алгоритм должен работать как с одной именованной переменной</b></p>

Ответ: 1, 2

Комментарий: AP00126

**<p><b>Является ли данное выражение истинным const float pi = 3.14159265;</b></p>**

1. <p><b>да</b></p>

2. <p><b>нет</b></p>

Ответ: 1

Комментарий: AP00127

**<p><b>Переменная типа char имеет размер</b></p>**

1. <p><b>1 байт</b></p>

2. <p><b>2 байта</b></p>

3. <p><b>4 байта</b></p>

Ответ: 1

Комментарий: AP00128

**<p><b>К общим принципам, которые следует использовать при разработке ПО, относятся: принцип модульности, принцип функциональной избирательности, принцип генерируемости, принцип функциональной избыточности (выбрать недостающие)</b></p>**

1. <p><b>частотный принцип</b></p>

2. <p><b>частный принцип</b></p>

3. <p><b>принцип умолчания</b></p>

4. <p><b>принцип совместимости</b></p>

5. <p><b>принцип развития</b></p>

Ответ: 1

Комментарий: AP00129

**<p><b>Что содержит в себе принцип генерируемости ?</b></p>**

1. <p><b>учитывает возможность проведения одной и той же работы (функции) различными средствами</b></p>

2. <p><b>определяет способ представления ПО, который бы позволял осуществлять настройку на конкретную конфигурацию технических средств, круг решаемых проблем, условия работы пользователя</b></p>

3. <p><b>основан на выделении в алгоритмах и в обрабатываемых структурах действий и данных по частоте использования</b></p>

4. <p><b>применяется для облегчения организации связей с системой, как на стадии генерации, так и при работе с уже готовым ПО.</b></p>

Ответ: 2

Комментарий: AP00130

**<p><b>Что содержит в себе общесистемный принцип совместимости?</b></p>**

1. <p><b>состоит в том, что на всех стадиях создания, функционирования и развития ПО его целостность будет обеспечиваться связями между подсистемами</b></p>

2. <p><b>состоит в том, что язык, символы, коды и средства обеспечения ПО согласованы</b></p>

3. <p><b>определяет, что во всех подсистемах, средствах обеспечения и компонентах ПО используются единые термины, символы, условные обозначения и способы представления</b></p>

Ответ: 2

Комментарий: AP00131

**<p><b>Со скольких стадий состоит жизненный цикл программного обеспечения?</b></p>**

1. <p><b>3</b></p>

2. <p><b>4</b></p>

3. <p><b>5</b></p>

4. <p><b>6</b></p>

Ответ: 3

Комментарий: AP00132

**<p><b>На какой стадии жизненного цикла ПО происходит больше всего затрат?</b></p>**

1. <p><b>определение требований и спецификаций</b></p>

2. <p><b>проектирование</b></p>

3. <p><b>программирование</b></p>

4. <p><b>отладка</b></p>

5. <p><b>сопровождение</b></p>

Ответ: 5

Комментарий: AP00133

**<p><b>Что в алгебре логики понимают под повествовательным предложением, относительно которого имеет смысл говорить, истинно оно или ложно?</b></p>**

1. <p><b>выражение</b></p>

2. <p><b>логическая операция</b></p>

3. <p><b>высказывание</b></p>

4. <p><b>аксиома алгебры логики</b></p>

Ответ: 3

Комментарий: AP00134

**<p><b>Как принято обозначать высказывания в алгебре логики?</b></p>**

1. <p><b>большими буквами русского алфавита</b></p>

2. <p><b>малыми буквами греческого алфавита</b></p>

3. <p><b>словами True или False</b></p>

4. <p><b>малыми буквами латинского алфавита</b></p>

5. <p><b>большими буквами латинского алфавита</b></p>

Ответ: 5

Комментарий: AP00135

**<p><b>Что является операцией логического умножения?</b></p>**

1. <p><b>эквиваленция</b></p>

2. <p><b>дизъюнкция</b></p>

3. <p><b>инверсия</b></p>

4. <p><b>импликация</b></p>

5. <p><b>конъюнкция</b></p>

Ответ: 5

Комментарий: AP00136

**<p><b>Какая операция объединяет высказывания словами «если..., то»?</b></p>**

1. <p><b>эквиваленция</b></p>

2. <p><b>дизъюнкция</b></p>

3. <p><b>инверсия</b></p>

4. <p><b>импликация</b></p>

5. <p><b>конъюнкция</b></p>

Ответ: 4

Комментарий: AP00137

**<p><b>С помощью какой операции образованное сложное высказывание истинно, если истинно хотя бы одно из входящих в него высказываний?</b></p>**

1. <p><b>эквиваленция</b></p>

2. <p><b>дизъюнкция</b></p>

3. <p><b>инверсия</b></p>

4. <p><b>импликация</b></p>

5. <p><b>конъюнкция</b></p>

Ответ: 2

Комментарий: AP00138

**<p><b>Какая операция в логике высказываний обозначается А ~ В?</b></p>**

1. <p><b>эквиваленция</b></p>

2. <p><b>дизъюнкция</b></p>

3. <p><b>инверсия</b></p>

4. <p><b>импликация</b></p>

5. <p><b>конъюнкция</b></p>

Ответ: 1

Комментарий: AP00139

**<p><b>Как можно установить истинность сложных высказываний?</b></p>**

1. <p><b>методом индукции</b></p>

2. <p><b>используя таблицы истинности</b></p>

3. <p><b>наугад</b></p>

4. <p><b>с помощью операции побитового сравнения</b></p>

Ответ: 2

Комментарий: AP00140

**<p><b>Как называются высказывания, у которых совпадают таблицы истинности?</b></p>**

1. <p><b>сложные</b></p>

2. <p><b>тождественные</b></p>

3. <p><b>истинные</b></p>

4. <p><b>совместимые</b></p>

5. <p><b>равносильные</b></p>

Ответ: 2, 5

Комментарий: AP00141

**<p><b>Примером какого свойства операций дизъюнкции, конъюнкции и отрицания является выражение<br>A^(BvC) = (A^B)v(A^C)?</b></p>**

1. <p><b>сочетательного закона</b></p>

2. <p><b>свойства коммутативности</b></p>

3. <p><b>распределительного закона</b></p>

4. <p><b>закона распределения</b></p>

5. <p><b>закона де Моргана</b></p>

Ответ: 3

Комментарий: AP00142

**<p><b>Зачем в некоторых ЯП в логику высказываний вводится значение NULL?</b></p>**

1. <p><b>в дополнение к false и true</b></p>

2. <p><b>вместо false и true</b></p>

3. <p><b>на случай, когда отсутствует один из аргументов</b></p>

4. <p><b>на случай, когда значения логических аргументов неизвестны или не определены</b></p>

5. <p><b>на случай, когда значения логических аргументов противоречивы</b></p>

Ответ: 1, 4

Комментарий: AP00143

**<p><b>Сколько поколений языков программирования рассматривают в истории развития инструментального программного обеспечения?</b></p>**

1. <p><b>9</b></p>

2. <p><b>7</b></p>

3. <p><b>5</b></p>

4. <p><b>3</b></p>

Ответ: 3

Комментарий: AP00144

**<p><b>Как можно охарактеризовать языки программирования пятого поколения?</b></p>**

1. <p><b>мобильные, человеко-ориентированные, простые в освоении языки высокого уровня</b></p>

2. <p><b>ориентированые на повышение интеллектуального уровня ЭВМ и интерфейса с языками</b></p>

3. <p><b>ориентированые на использование в конкретной ЭВМ, сложные в освоении машинные языки</b></p>

4. <p><b>языки искусственного интеллекта, экспертных систем и баз знаний, естественные языки</b></p>

Ответ: 2, 4

Комментарий: AP00145

**<p><b>Что представляет собой синтаксис как уровень рассмотрения элементов языков программирования?</b></p>**

1. <p><b>совокупность правил образования языковых конструкций, или предложений ЯП — блоков, процедур, составных операторов, условных операторов, операторов цикла и пр.</b></p>

2. <p><b>совокупность правил образования цепочек символов, образующих идентификаторы (переменные и метки), операторы, операции и пр.</b></p>

3. <p><b>проверка смысловой правильности конструкции.</b></p>

4. <p><b>уровень, содержащий принцип вложенности (рекурсивность) правил построения конструкций.</b></p>

5. <p><b>совокупность символов, отображаемых на устройствах печати и экранах и/или вводимых с клавиатуры терминала.</b></p>

Ответ: 1, 4

Комментарий: AP00146

**<p><b>Из каких частей, как правило, состоит исходная программа?</b></p>**

1. <p><b>раздел разделов</b></p>

2. <p><b>раздел 1, 2, 3</b></p>

3. <p><b>блоки операторов</b></p>

4. <p><b>раздел идентификации, раздел связи</b></p>

5. <p><b>набор машинных инструкций</b></p>

Ответ: 3, 4

Комментарий: AP00147

**<p><b>Что относится к типовым операторам управления вычислительным процессом?</b></p>**

1. <p><b>набор машинных инструкций</b></p>

2. <p><b>организация циклов</b></p>

3. <p><b>ветвление программы</b></p>

4. <p><b>блоки операторов</b></p>

Ответ: 2, 3, 4

Комментарий: AP00148

**<p><b>Какую функцию выполняют логические условия как компонент управляющих операторов?</b></p>**

1. <p><b>вычисление логических переменных в рамках правил образования логических выражений</b></p>

2. <p><b>выработка решения о продолжении цикла, переходе по ветви, выходе из блока и т.д.</b></p>

3. <p><b>пересылка значений переменных, констант, функций в принимающую переменную</b></p>

4. <p><b>сопоставление значения констант, функций, переменных отношениями различного типа</b></p>

5. <p><b>вычисление значений строчной (символьной) переменной путем соединения, пересечения, вычисления строк</b></p>

Ответ: 2, 4

Комментарий: AP00149

**<p><b>Что является средством, позволяющим многократно использовать в разных местах основной программы один раз описанный фрагмент алгоритма?</b></p>**

1. <p><b>оператор присваивания</b></p>

2. <p><b>функция</b></p>

3. <p><b>подпрограмма</b></p>

4. <p><b>формальные и фактические параметры</b></p>

Ответ: 2, 3

Комментарий: AP00150

**<p><b>Где осуществляются локальные объявления?</b></p>**

1. <p><b>внутри программы в целом и доступны как самой программе, так и всем ее подпрограммам</b></p>

2. <p><b>внутри подпрограммы и доступны как самой программе, так и всем ее подпрограммам</b></p>

3. <p><b>внутри подпрограммы и доступны только ей</b></p>

4. <p><b>внутри программы в целом и доступны ближайшей подпрограмме</b></p>

Ответ: 3

Комментарий: AP00151

**<p><b>Какими являются параметры, содержащиеся в объявлении подпрограммы?</b></p>**

1. <p><b>фактическими</b></p>

2. <p><b>подпрограммными</b></p>

3. <p><b>неформальными</b></p>

4. <p><b>формальными</b></p>

Ответ: 4

Комментарий: AP00152

**<p><b>Какие системы описания языков программирования используются?</b></p>**

1. <p><b>нотация NT</b></p>

2. <p><b>нотация Бэкуса</b></p>

3. <p><b>нотация IBM</b></p>

4. <p><b>нотация Гейтса</b></p>

5. <p><b>нотация Pentium</b></p>

Ответ: 2, 3

Комментарий: AP00153

**<p><b>Что представляет собой система программирования?</b></p>**

1. <p><b>совокупность средств разработки программ</b></p>

2. <p><b>совокупность средств, обеспечивающих автоматизацию составления и отладки программ пользователя</b></p>

3. <p><b>средства компьютера, среди которых находится операционная система и утилиты поддержки</b></p>

Ответ: 1, 2

Комментарий: AP00154

**<p><b>Какие признаки присущи системе программирования?</b></p>**

1. <p><b>набор исходных языков</b></p>

2. <p><b>автоматизация составления</b></p>

3. <p><b>возможность расширения</b></p>

4. <p><b>отладка программ пользователя</b></p>

Ответ: 1, 3

Комментарий: AP00155

**<p><b>Что включает в себя структура абстрактной многоязыковой, открытой, компилирующей системы программирования?</b></p>**

1. <p><b>трансляция</b></p>

2. <p><b>отладка</b></p>

3. <p><b>компиляция</b></p>

4. <p><b>преобразование</b></p>

5. <p><b>ввод</b></p>

Ответ: 1, 3, 5

Комментарий: AP00156

**<p><b>Что называют текстом программы на машинном языке, включающей машинные инструкции, словари, служебную информацию?</b></p>**

1. <p><b>загрузочный модуль</b></p>

2. <p><b>объектный модуль</b></p>

3. <p><b>предпроцессинг</b></p>

Ответ: 2

Комментарий: AP00157

**<p><b>С помощью каких специальных программных средств осуществляется загрузочный модуль?</b></p>**

1. <p><b>редакторов связей</b></p>

2. <p><b>объектного модуля</b></p>

3. <p><b>компоновщика</b></p>

4. <p><b>построителя задач</b></p>

Ответ: 1, 3, 4

Комментарий: AP00158

**<p><b>Какие проблемы возникают в процессе использования готовых подпрограмм?</b></p>**

1. <p><b>хранение имеющихся подпрограмм</b></p>

2. <p><b>алгоритм используется многократно при решении задач</b></p>

3. <p><b>размещение используемых подпрограмм в памяти ЭВМ</b></p>

4. <p><b>организация взаимодействия подпрограмм с основной программой</b></p>

Ответ: 1, 3, 4

Комментарий: AP00159

**<p><b>На какие библиотеки делятся библиотеки подпрограмм?</b></p>**

1. <p><b>библиотеки статического вызова</b></p>

2. <p><b>библиотеки логического вызова</b></p>

3. <p><b>библиотеки динамического вызова</b></p>

Ответ: 1, 3

Комментарий: AP00160

**<p><b>Какие специальные средства обработки исключительных ситуаций предоставляются программисту?</b></p>**

1. <p><b>размещение используемых подпрограмм в памяти ЭВМ</b></p>

2. <p><b>какой-нибудь аппарат</b></p>

3. <p><b>ошибки</b></p>

4. <p><b>ничего из предложенного</b></p>

Ответ: 4

Комментарий: AP00161

**<p><b>Что обозначают ключевые слова try и finally?</b></p>**

1. <p><b>организацию взаимодействия подпрограмм с основной программой</b></p>

2. <p><b>размещение в оперативной памяти и заполнение некоторыми значениями массив целых чисел</b></p>

3. <p><b>два программных блока обработки завершения</b></p>

4. <p><b>обработку элементов массива</b></p>

Ответ: 3

Комментарий: AP00162

**<p><b>Что пропущено в синтаксисе обработки исключений:<br><b>try</b><br>&lt;Защищенный блок&gt;<br><b>exept</b> &lt;…&gt;<br>&lt;обработчик исключений&gt; ?</b></p>**

1. <p><b>аппарат обнаружения</b></p>

2. <p><b>фильтр исключений</b></p>

3. <p><b>обработка исключений</b></p>

Ответ: 2

Комментарий: AP00163

**<p><b>Отметьте правильное высказывание:</b></p>**

1. <p><b>загрузочный модуль после сборки помещается в качестве раздела в пользовательскую библиотеку программ</b></p>

2. <p><b>подпрограммы, которые удовлетворяют всем требованиям выбранной системы, называются логическими</b></p>

3. <p><b>в случае динамической компоновки компоновщик просто использует информацию о подпрограмме для настройки соответствующих таблиц в исполняемом файле</b></p>

4. <p><b>в хорошо написанной программе обычно должно быть обращение к недопустимому адресу памяти</b></p>

Ответ: 1, 3

Комментарий: AP00164

**<p><b>Что представляет собой внешний файл?</b></p>**

1. <p><b>устройство ввода-вывода</b></p>

2. <p><b>поименованный файл на диске</b></p>

3. <p><b>последовательность данных определённой длины</b></p>

Ответ: 1

Комментарий: AP00165

**<p><b>Какие операции включает цикл обработки файла?</b></p>**

1. <p><b>закрытие файла</b></p>

2. <p><b>открытие файла</b></p>

3. <p><b>сохранение</b></p>

4. <p><b>организация цикла, управляемого файлом</b></p>

5. <p><b>изменение информации</b></p>

Ответ: 1, 2, 4

Комментарий: AP00166

**<p><b>На какие типы подразделяются устройства, на которые осуществляется вывод данных из программы, или с которых осуществляется ввод?</b></p>**

1. <p><b>по способу выборки информации</b></p>

2. <p><b>по расширению</b></p>

3. <p><b>посимвольный обмен информацией</b></p>

4. <p><b>передача информации порциями (записями)</b></p>

Ответ: 3, 4

Комментарий: AP00167

**<p><b>Если говорить о типе файлов, расположенных на том или ином устройстве - какие типы могут быть рассмотрены?</b></p>**

1. <p><b>по типу записей</b></p>

2. <p><b>по имени</b></p>

3. <p><b>по способу выборки информации</b></p>

Ответ: 1, 3

Комментарий: AP00168

**<p><b>Что выполняет метод сопоставления?</b></p>**

1. <p><b>обнаружение и обработка ошибок при считывании данных</b></p>

2. <p><b>«привязывание» данных</b></p>

3. <p><b>организацию цикла</b></p>

Ответ: 1

Комментарий: AP00169

**<p><b>Какими могут быть файлы по способу выборки информации ?</b></p>**

1. <p><b>файлы последовательного доступа</b></p>

2. <p><b>файлы, образующие байтовый или битовый поток</b></p>

3. <p><b>файлы прямого доступа</b></p>

Ответ: 1, 3

Комментарий: AP00170

**<p><b>Что нужно уметь при считывании информации из файла?</b></p>**

1. <p><b>определять начало и окончание элементарного данного</b></p>

2. <p><b>обеспечить циклическое чтение строк файла в программный буфер</b></p>

3. <p><b>определять начало и окончание записи файла</b></p>

Ответ: 1, 3

Комментарий: AP00171

**<p><b>Какие операции характерны для файлов последовательного действия?</b></p>**

1. <p><b>операции последовательного чтения</b></p>

2. <p><b>операции параллельного чтения</b></p>

3. <p><b>операции последовательного записи в конец файла</b></p>

Ответ: 1, 3

Комментарий: AP00172

**<p><b>Какой длины могут быть записи в файлах последовательного доступа?</b></p>**

1. <p><b>фиксированной длины</b></p>

2. <p><b>определённой длины</b></p>

3. <p><b>переменной (неопределенной) длины</b></p>

Ответ: 1, 3

Комментарий: AP00173

**<p><b>Какие операции характерны для файлов прямого доступа?</b></p>**

1. <p><b>записи по произвольному адресу</b></p>

2. <p><b>последовательного записи в конец файла</b></p>

3. <p><b>чтения по произвольному адресу</b></p>

Ответ: 1, 3

Комментарий: AP00174

## 2 тема

<p><b> Компоновщик - это</b></p>

<p><b>1) Программа, которая принимает несколько объектных модулей и на их основе создает загрузочный модуль.</b></p>

<p><b>2) Транслятор с исходного языка на языке ассемблера в программу на машинном коде</b></p>

<p><b>3) Оптимизатор кода</b></p>

<p><b>4) Редактор кода</b></p>

Ответ: 1

Комментарий: AP0002001

<p><b> Результатом работы компоновщика является:</b></p>

<p><b>1) Объектный модуль</b></p>

<p><b>2) Загрузочный модуль </b></p>

<p><b>3) Исходный модуль</b></p>

<p><b>4) Абсолютный модуль</b></p>

Ответ: 2

Комментарий: AP0002002

<p><b> Для второго поколения ЯП характерно:</b></p>

<p><b>1) Машинно-зависим</b></p>

<p><b>2) ЯП носят непроцедурный характер</b></p>

<p><b>3) Ориентированы на использование конкретной ЭВМ</b></p>

<p><b>4) Характеризуются созданием языков ассемблерного типа</b></p>

Ответ: 1, 4

Комментарий: AP0002003

<p><b> Результатом работы транслятора является:</b></p>

<p><b>1) Объектный модуль</b></p>

<p><b>2) Загрузочный модуль </b></p>

<p><b>3) Исходный модуль</b></p>

<p><b>4) Абсолютный модуль</b></p>

Ответ: 1

Комментарий: AP0002004

<p><b> Набор инструкций, выполняющийся непосредственно центральным процессором компьютера - это ...</b></p>

<p><b>1) Код ассемблера</b></p>

<p><b>2) Исходный код</b></p>

<p><b>3) Машинный код</b></p>

<p><b>4) Загрузочный код</b></p>

Ответ: 3

Комментарий: AP0002005

<p><b> Система программирования это -</b></p>

<p><b>1) Совокупность программных средств для программирования состоящая из компилятора, библиотек, компоновщика, загрузчика, отладчика и др.</b></p>

<p><b>2) Программа преобразующая исходный код в машинный код</b></p>

<p><b>3) Набор символов, разрешенных к использованию языком программирования.</b></p>

<p><b>4) Программа позволяющая подготовку исходного кода</b></p>

Ответ: 1

Комментарий: AP0002006

<p><b> Что такое объектный модуль?</b></p>

<p><b>1) Текст программы на машинном языке, включающий машинные инструкции, словари, служебную информацию. </b></p>

<p><b>2) Результат работы компоновщика.</b></p>

<p><b>3) Программа, обычно входящая в состав операционной системы, предназначенная для запуска процесса операционной системы на основе загрузочного модуля. </b></p>

<p><b>4) Компонент системы программирования (или IDE) - программа, позволяющая контролировать ход выполнения программы, просматривать и изменять области памяти. </b></p>

Ответ: 1

Комментарий: AP0002007

<p><b> Алфавит языка программирования - это</b></p>

<p><b>1) Набор зарезервированных слов ЯП.</b></p>

<p><b>2) Набор символов, разрешенных к использованию языком программирования.</b></p>

<p><b>3) Набор переменных, созданных во время разработки кода</b></p>

<p><b>4) Символы латинского алфавита</b></p>

Ответ: 2

Комментарий: AP0002008

<p><b> Указатель в С++ - это </b></p>

<p><b>1) Переменная в которой хранится адрес другого объекта</b></p>

<p><b>2) Переменная который работает как псевдоним другого объекта или значения.</b></p>

<p><b>3) Значение переменной(то, что стоит после знака “=”)</b></p>

<p><b>4) Курсор мыши</b></p>

Ответ: 1

Комментарий: AP0002009

<p><b> Исключение - это:</b></p>

<p><b>1) Алгоритм, записанный на языке программирования;</b></p>

<p><b>2) Событие при выполнении программы, при котором ее дальнейшее выполнение становится бессмысленным.</b></p>

<p><b>3) Протокол взаимодействия компонентов компьютерной сети;</b></p>

<p><b>4) Ориентированный граф, указывающий порядок исполнения команд компьютера;</b></p>

Ответ: 2

Комментарий: AP0002010

<p><b> Директивы препроцессора не могут:</b></p>

<p><b>1) Заменить какие-то лексемы в исходном тексте; </b></p>

<p><b>2) Вставить содержимое других файлов в указанном месте; </b></p>

<p><b>3) Подавить компиляцию части файла. </b></p>

<p><b>4) Удалить переменную с содержимым в указанном месте.</b></p>

Ответ: 4

Комментарий: AP0002011

<p><b> #pragma once :</b></p>

<p><b>1) Контролирует, чтобы конкретный включаемый файл при компиляции подключался строго один раз.</b></p>

<p><b>2) Вставляет содержимое заданного файла в место расположения этой директивы в исходном тексте программы. </b></p>

<p><b>3) Определяет идентификатор и последовательность символов, которая заменит этот идентификатор в тексте программы.</b></p>

<p><b>4) Заменяет все вхождения идентификатора макроса в исходном файле на последовательность символов. </b></p>

Ответ: 1

Комментарий: AP0002012

<p><b> Директива препроцессора #define:</b></p>

<p><b>1) Контролирует, чтобы конкретный включаемый файл при компиляции подключался строго один раз.</b></p>

<p><b>2) Заменяет все вхождения идентификатора макроса в исходном файле на последовательность символов.</b></p>

<p><b>3) Сохраняет код в определённый файл.</b></p>

<p><b>4) Вставляет содержимое заданного файла в место расположения этой директивы в исходном тексте программы.</b></p>

Ответ: 2

Комментарий: AP0002013

<p><b> Директива #include:</b></p>

<p><b>1) Удаляет все указанные переменные и их содержимое.</b></p>

<p><b>2)Заменяет все вхождения идентификатора макроса в исходном файле на последовательность символов.</b></p>

<p><b>3) Вставляет содержимое заданного файла в место расположения этой директивы в исходном тексте программы.</b></p>

<p><b>4) Отменяет предыдущее определение.</b></p>

Ответ: 4

Комментарий: AP0002014

<p><b> Директивы условной компиляции #ifdef, #ifndef, #if, defined(). Выберите верное утверждение:</b></p>

<p><b>1) Для одной директивы #ifdef можно иметь несколько соответствующих закрывающих директив #endif.</b></p>

<p><b>2) Директивы условной компиляции #ifdef и #ifndef не принимают участие в компиляции исходного файла.</b></p>

<p><b>3) Каждая директива #ifdef в исходном коде должна иметь соответствующую закрывающую директиву #endif. </b></p>

<p><b>4) Директива условной компиляции #if может не иметь соответствующую закрывающую директиву #endif.</b></p>

Ответ: 3

Комментарий: AP0002015

<p><b> Что из себя представляет string?</b></p>

<p><b>1)Массив</b></p>

<p><b>2)Тип данных</b></p>

<p><b>3)Структуру</b></p>

<p><b>4)Класс </b></p>

Ответ: 4

Комментарий: AP0002016

<p><b> В чем принципиальное отличие компилятора от интерперетатора? </b></p>

<p><b>1) Компилятор генерирует загрузочный модуль, интерпретатор переводит исходный код в код, близкий к машинному.</b></p>

<p><b>2) Компилятор переводит исходный код сразу в машинный, интерпретатор переводит исходный код в код, близкий к машинному</b></p>

<p><b>3) Компилятор переводит исходный код в код, близкий к машинному, интерпретатор переводит исходный код сразу в машинный </b></p>

<p><b>4) Нет отличий</b></p>

Ответ: 3

Комментарий: AP0002017

<p><b> Для пятого поколения ЯП характерно:</b></p>

<p><b>1) Мобильны</b></p>

<p><b>2) Ориентированы на непрофессионального пользователя и на ЭВМ с параллельной структурой</b></p>

<p><b>3)Ориентированы на использование в конкретной ЭВМ</b></p>

<p><b>4)Ориентированы на повышение интеллектуального уровня ЭВМ и интерфейса с языками </b></p>

Ответ: 4

Комментарий: AP0002018

<p><b> Компиляция - многоступенчатый процесс, включающий фазы :</b></p>

<p><b>1) Синтаксический анализ</b></p>

<p><b>2) Семантический анализ</b></p>

<p><b>3) Генерация объектного кода</b></p>

<p><b>4) Все вышеперечисленное</b></p>

Ответ: 4

Комментарий: AP0002019

<p><b> Выберите возможности, которые можно включить в категории General(общие параметры)</b></p>

<p><b>1) База данных программы</b></p>

<p><b>2) Инкрементная компоновка</b></p>

<p><b>3) Генерация карты компоновки</b></p>

<p><b>4) Вывод сообщений процесса</b></p>

Ответ: 2, 3

Комментарий: AP0002020

<p><b> Выберите возможности, которые можно включить в категории Customization</b></p>

<p><b>1) Имя выходного файла</b></p>

<p><b>2) Отмена библиотек, включаемых по умолчанию</b></p>

<p><b>3) Профилирование</b></p>

<p><b>4) Заставка</b></p>

Ответ: 1, 4

Комментарий: AP0002021

<p><b> Что позволяет категория Output?</b></p>

<p><b>1) Позволяет указать объектные модули и библиотеки</b></p>

<p><b>2) Позволяет задать имя файла ресурсов</b></p>

<p><b>3) Позволяет установить базовый адрес, точку входа, размер стека и информацию о версии проекта</b></p>

<p><b>4) Позволяет задать имя выходного файла, имя выходного заголовочного файла, определения для препроцессора и заставку</b></p>

Ответ: 3

Комментарий: AP0002022

<p><b> Прототипом языка программирования C был язык:</b></p>

<p><b>1) A</b></p>

<p><b>2) B</b></p>

<p><b>3) D</b></p>

<p><b>4) E</b></p>

Ответ: 2

Комментарий: AP0002023

<p><b> Отладчик (debugger)-</b></p>

<p><b>1) Программа, позволяющая контролировать ход выполнения программы (приостанавливать, выполнять пошагово)</b></p>

<p><b>2) Программа, предназначенная для запуска процесса операционной системы на основе загрузочного модуля.</b></p>

<p><b>3) Программа, принимающая один или несколько объектных модулей и формирующая на их основе загрузочный модуль.</b></p>

<p><b>4) Программа, преобразующая исходный код на одном языке программирования в исходный код на другом языке.</b></p>

Ответ: 1

Комментарий: AP0002024

<p><b> Выберите назначение подкаталога BIN</b></p>

<p><b>1) Исполняемые файлы и средства для построения 32-разрядных приложений</b></p>

<p><b>2) Файлы справки</b></p>

<p><b>3) Заголовочные файлы C++</b></p>

<p><b>4) Библиотеки языка C++ и Win32 SDK</b></p>

Ответ: 1

Комментарий: AP0002025

<p><b> Разработчик языка C:</b></p>

<p><b>1) Билл Гейтс</b></p>

<p><b>2) Деннис Ритчи</b></p>

<p><b>3) Кен Томпсон</b></p>

<p><b>4) Билл Клинтон</b></p>

Ответ: 2

Комментарий: AP0002026

<p><b> Выберите назначение подкаталога LIB</b></p>

<p><b>1) Файлы библиотеки Microsoft Foundation Classes (MFS)</b></p>

<p><b>2) Библиотеки языка C++ и Win32 SDK</b></p>

<p><b>3) Файлы для построения приложений OLE</b></p>

<p><b>4) Исполняемые файлы и средства для построения 32-разрядных приложений</b></p>

Ответ: 2

Комментарий: AP0002027

<p><b> Интерпретатор - это разновидность:</b></p>

<p><b>1) Компилятора</b></p>

<p><b>2) Транслятора</b></p>

<p><b>3) Компоновщика</b></p>

<p><b>4) Отладчика</b></p>

Ответ: 2

Комментарий: AP0002028

<p><b> Из чего состоит Документация по Visual C++</b></p>

<p><b>1) QuickReference</b></p>

<p><b>2) BooksOnline</b></p>

<p><b>3) QuickReference и BooksOnline</b></p>

<p><b>4) MicrosoftFoundationClasses</b></p>

Ответ: 3

Комментарий: AP0002029

<p><b> Компонента системы программирования, позволяющая подготовить исходный код программы -</b></p>

<p><b>1) Текстовый редактор</b></p>

<p><b>2) Транслятор</b></p>

<p><b>3) Компилятор</b></p>

<p><b>4) Компоновщик</b></p>

Ответ: 1

Комментарий: AP0002030

<p><b> Выберите назначение утилиты WinDiff.</b></p>

<p><b>1) Позволяет существенно уменьшить размер программы и повысить скорость ее выполнения на величину до 60 процентов</b></p>

<p><b>2) Позволяет быстро устанавливать параметры, необходимые для отслеживания выполняемых процессов</b></p>

<p><b>3) Позволяет быстро тестировать ваши управляющие элементы</b></p>

<p><b>4) Позволяет в графическом виде сравнивать и изменять два файла или два каталога</b></p>

Ответ: 4

Комментарий: AP0002031

<p><b> Какая закладка позволяет задать дополнительные инструменты для построения программы?</b></p>

<p><b>1) Закладка Debug</b></p>

<p><b>2) Закладка General</b></p>

<p><b>3) Закладки CustomBuild</b></p>

<p><b>4) Закладка C/C++</b></p>

Ответ: 3

Комментарий: AP0002032

<p><b> Выберите назначение утилиты ProcessViewer.</b></p>

<p><b>1) Позволяет быстро тестировать ваши управляющие элементы</b></p>

<p><b>2) Позволяет быстро устанавливать параметры, необходимые для отслеживания выполняемых процессов</b></p>

<p><b>3) Позволяет в графическом виде сравнивать и изменять два файла или два каталога</b></p>

<p><b>4) Позволяет существенно уменьшить размер программы и повысить скорость ее выполнения на величину до 60 процентов</b></p>

Ответ: 2

Комментарий: AP0002033

## 3 тема

**<p><b> Какова функция команды Hide??? </b></p>**

1. <p><b> делит текущее окно на две части </b></p>

2. <p><b> делает текущее окно скрытым</b></p>

3. <p><b> создаёт в текущем проекте новое окно </b></p>

Ответ: 2

Комментарий: AP0003001

**<p><b> Как работает команда Build??? </b></p>**

1. <p><b> В процессе сборки анализируются все файлы проекта и затем компилируются и компонуются лишь те из них, которые были изменены после создания исполняемого файла проекта </b></p>

2. <p><b> В процессе сборки анализируются только те файлы проекта, которые были изменены после создания исполняемого файла проекта и только потом они компилируются и компонуются </b></p>

3. <p><b> В процессе сборки анализируются все файлы проекта, а затем все эти файлы компилируются и компонуются </b></p>

Ответ: 1

Комментарий: AP0003002

**<p><b> Какова функция команды Undo??? </b></p>**

1. <p><b> Позволяет отменить последнее изменение, которое было внесено в процессе редактирования</b></p>

2. <p><b> Позволяет вернуть те изменения, которые по ошибке отменили </b></p>

3. <p><b> Позволяет отменить все изменения, которые были внесены в процессе редактирования </b></p>

Ответ: 1

Комментарий: AP0003003

**<p><b> Что обозначает данная кодовая последовательность &f??? </b></p>**

1. <p><b> Создать новый файл </b></p>

2. <p><b> Вставить имя файла </b></p>

3. <p><b> Открыть предыдущий файл</b></p>

Ответ: 2

Комментарий: AP0003004

**<p><b> Чтобы получить на бумаге копию содержимого активного окна, нужно выполнить команду… </b></p>**

1. <p><b> Print</b></p>

2. <p><b> Print Active Window</b></p>

3. <p><b> Print Range </b></p>

Ответ: 1

Комментарий: AP0003005

**<p><b> Что делает команда Properties? </b></p>**

1. <p><b> Вызывает консольное окно, в котором выводятся характеристики текущего окна </b></p>

2. <p><b> Показывает свойства данного файла </b></p>

3. <p><b> Вызывает окно диалога, в котором задаются характеристики текущего окна </b></p>

Ответ: 3

Комментарий: AP0003006

**<p><b> Что обозначает данная кодовая последовательность &r??? </b></p>**

1. <p><b> Выровнять вправо</b></p>

2. <p><b> Открыть файл для чтения</b></p>

3. <p><b> Открывает окно, отображающее содержимое памяти </b></p>

Ответ: 1

Комментарий: AP0003007

**<p><b> Для чего нужна команда Redo??? </b></p>**

1. <p><b> Позволяет быстро находить и исправлять текст в активном окне</b></p>

2. <p><b> Позволяет отменить последнее изменение, которое было внесено в процессе редактирования</b></p>

3. <p><b> Чтобы вернуть изменения, которые мы по ошибке отменили </b></p>

Ответ: 3

Комментарий: AP0003008

**<p><b> Что обозначает шаблон “^”, который можно использовать в стоке Find What???</b></p>**

1. <p><b> Отмечает начало строки</b></p>

2. <p><b> Задаёт определённый набор символов </b></p>

3. <p><b> Задаёт любую последовательность символов </b></p>

Ответ: 1

Комментарий: AP0003009

**<p><b> Что обозначает шаблон “+”, который можно использовать в стоке Find What??? </b></p>**

1. <p><b> Задаёт последовательность символов</b></p>

2. <p><b> Задаёт любое число символов перед искомым словом</b></p>

3. <p><b> Задаёт один (любой) символ </b></p>

Ответ: 2

Комментарий: AP0003010

**<p><b> Что делает команда Toolbars??? </b></p>**

1. <p><b> Вызывает диалоговое окно, позволяющее выбрать панели инструментов, которые должны отображаться в рабочей среде </b></p>

2. <p><b> Позволяет создать свой рабочий инструмент</b></p>

3. <p><b> Дублирует код </b></p>

Ответ: 1

Комментарий: AP0003011

**<p><b> для чего используется команда Open Workspace </b></p>**

1. <p><b> для открытия ранее сохраненного рабочего пространства </b></p>

2. <p><b> для создания рабочего пространства </b></p>

3. <p><b> для закрытия рабочего пространства </b></p>

Ответ: 1

Комментарий: AP0003012

**<p><b> что делает команда debug </b></p>**

1. <p><b> используется для выбора конфинурации проекта </b></p>

2. <p><b> запускает интегрированный отладчик </b></p>

3. <p><b> запускает программу </b></p>

Ответ: 2

Комментарий: AP0003013

**<p><b> что делает команда compile </b></p>**

1. <p><b> заставляет среду компилировать текст в текущем режиме </b></p>

2. <p><b> создаёт в текущем проекте новое окно </b></p>

3. <p><b> запускает программу </b></p>

Ответ: 1

Комментарий: AP0003014

**<p><b> что делает команда split </b></p>**

1. <p><b> делит текущее окно на две части </b></p>

2. <p><b> создаёт в текущем проекте новое окно </b></p>

3. <p><b> скрывает текущее окно </b></p>

Ответ: 1

Комментарий: AP0003015

**<p><b> для чего используется команда execute </b></p>**

1. <p><b> для запуска программы </b></p>

2. <p><b> для запуска интегрированного отладчика </b></p>

3. <p><b> для выбора конфигурации проектов </b></p>

Ответ: 1

Комментарий: AP0003016

**<p><b> в какое окно помещаются оповещения об обнаруженных во время компиляции ошибках </b></p>**

1. <p><b> results </b></p>

2. <p><b> build </b></p>

3. <p><b> output </b></p>

Ответ: 3

Комментарий: AP0003017

**<p><b> какая команда позволяет создавать, удалять точки останова </b></p>**

1. <p><b> properties </b></p>

2. <p><b> build </b></p>

3. <p><b> breakpoints </b></p>

Ответ: 3

Комментарий: AP0003018

**<p><b> какая команда позволяет создавать, удалять закладки </b></p>**

1. <p><b> properties </b></p>

2. <p><b> bookmark </b></p>

3. <p><b> breakpoints </b></p>

Ответ: 2

Комментарий: AP0003019

**<p><b> шаблон “\*” команды Find </b></p>**

1. <p><b> Задаёт любую последовательность символов </b></p>

2. <p><b> Задаёт один символ </b></p>

3. <p><b> Отмечает начало строки </b></p>

Ответ: 1

Комментарий: AP0003020

**<p><b> шаблон “$” команды Find </b></p>**

1. <p><b> Задаёт любую последовательность символов </b></p>

2. <p><b> Задаёт один символ </b></p>

3. <p><b> Отмечает начало строки </b></p>

4. <p><b> Ни один из вышеперечисленных вариантов не является верным </b></p>

Ответ: 4

Комментарий: AP0003021

**<p><b> выберите верное утверждение </b></p>**

1. <p><b> Команду Find можно настроить для поиска с учетом регистра, без учета регистра, а также поиска только полного слова </b></p>

2. <p><b> Команда Find позволяет указать направление поиска </b></p>

3. <p><b> Всё вышеперечисленное верно </b></p>

4. <p><b> Ни один из вышеперечисленных вариантов не является верным </b></p>

Ответ: 3

Комментарий: AP0003022

**<p><b> Команда Find в меню Edit позволяет настроить поиск с учётом регистра??? </b></p>**

1. <p><b> да </b></p>

2. <p><b> нет</b></p>

3. <p><b> такой команды не существует</b></p>

Ответ: 1

Комментарий: AP000323

**<p><b> Какую команду в меню Edit нужно использовать, чтобы найти строку с описанием константы, переменной или функции??? </b></p>**

1. <p><b> Find</b></p>

2. <p><b> Replace</b></p>

3. <p><b> Go To </b></p>

Ответ: 3

Комментарий: AP000324

**<p><b>Если название пункта меню изображено серым цветом, это значит, что соответствующая команда в данный момент??? </b></p>**

1. <p><b> невыполнима </b></p>

2. <p><b> выполняется </b></p>

3. <p><b> готовится к выполнению</b></p>

4. <p><b> уже выполнена</b></p>

Ответ: 1

Комментарий: AP000325

**<p><b>Bookmark используются для того, чтобы…??? </b></p>**

1. <p><b> отмечать некоторые строки в тексте Программы, к которым придется обратиться позже</b></p>

2. <p><b> выделять некоторые строки в тексте Программы, к которым придётся обратиться позже</b></p>

3. <p><b> удалять некоторые строки в тексте Программы, к которым уже не понадобятся</b></p>

Ответ: 1

Комментарий: AP000326

**<p><b>Какая команда располагает все открытые на экране окна одно за другим со сдвигом вправо и вниз??? </b></p>**

1. <p><b> Split </b></p>

2. <p><b> Hide</b></p>

3. <p><b> Cascade</b></p>

Ответ: 3

Комментарий: AP000327

**<p><b>Какая команда выдаст хронологический список объектов или ссылок на них, которые вы искали ранее??? </b></p>**

1. <p><b> View</b></p>

2. <p><b> Browse </b></p>

3. <p><b> Split</b></p>

Ответ: 2

Комментарий: AP0003328

**<p><b>Какая команда открывает список всех имеющихся в среде окон, позволяя быстро найти нужное??? </b></p>**

1. <p><b> Windows</b></p>

2. <p><b> Browse </b></p>

3. <p><b> All</b></p>

Ответ: 1

Комментарий: AP000329

**<p><b>Какая команда делит текущее окно на две части??? </b></p>**

1. <p><b> Browse</b></p>

2. <p><b> Split</b></p>

3. <p><b> Cascade</b></p>

Ответ: 2

Комментарий: AP000330

**<p><b>Какая команда делает текущее окно скрытым??? </b></p>**

1. <p><b> Split</b></p>

2. <p><b> Cascade</b></p>

3. <p><b> Hide</b></p>

Ответ: 3

Комментарий: AP000331

**<p><b>Какая команда позволяет создать копию содержимого активного окна под другим именем?? </b></p>**

1. <p><b> Save As</b></p>

2. <p><b> Save All</b></p>

3. <p><b> Save</b></p>

Ответ: 1

Комментарий: AP000332

**<p><b>?Какая команда открывает окно, позволяющее увидеть активные в данный момент переменные, локальные переменные или указатель this??? </b></p>**

1. <p><b> Output</b></p>

2. <p><b> Variables</b></p>

3. <p><b> Watch</b></p>

Ответ: 2

Комментарий: AP000333

## 4 тема

<p><b>Кто создал язык С</b></p>

1)<p><b>[Бьёрн Страуструп](https://www.google.com/search?client=opera&hs=PUD&sxsrf=ALeKk038_8j55HQ6fqIMy5_l13Z6HaWQaQ:1589209986315&q=%D0%91%D1%8C%D1%91%D1%80%D0%BD+%D0%A1%D1%82%D1%80%D0%B0%D1%83%D1%81%D1%82%D1%80%D1%83%D0%BF&stick=H4sIAAAAAAAAAONgVuLQz9U3MDRPSnnEaMwt8PLHPWEprUlrTl5jVOHiCs7IL3fNK8ksqRQS42KDsnikuLjgmngWscpfmHix5-LEiw0X9ipcWHixCcjYcLH5YiOIdbH5wn4ASKxp1WQAAAA)</b></p>

2)<p><b>Кен Томпсон и Денис Ритчи</b></p>

3)<p><b>[Джеймс Гослинг](https://www.google.com/search?client=opera&hs=SWD&sxsrf=ALeKk00hN8hLDpwGlkgnOwVD6FH1Z2x4yw:1589210113246&q=%D0%94%D0%B6%D0%B5%D0%B9%D0%BC%D1%81+%D0%93%D0%BE%D1%81%D0%BB%D0%B8%D0%BD%D0%B3&stick=H4sIAAAAAAAAAONgVuLQz9U3SMkzjH_EaMwt8PLHPWEprUlrTl5jVOHiCs7IL3fNK8ksqRQS42KDsnikuLjgmngWsUpfmHJh24WtF3Ze2HOxUeHC5Av7LjZe2H1hx4W9FzYDAKbrrQ5gAAAA) и [Брендан Эйх](https://www.google.com/search?client=opera&hs=Ors&sxsrf=ALeKk03EQ6zgbzCUc-t9btZHBhZffWqQBA:1589210133600&q=%D0%91%D1%80%D0%B5%D0%BD%D0%B4%D0%B0%D0%BD+%D0%AD%D0%B9%D1%85&stick=H4sIAAAAAAAAAONgVuLUz9U3MDE2KrZ8xGjCLfDyxz1hKe1Ja05eY1Tl4grOyC93zSvJLKkUEudig7J4pbi5ELp4FrGKXph4seHC1gt7L2y5sOHCXoULay_svNgKAF18oRZcAAAA)</b></p>

Ответ: 2

Комментарий: AP0004001

<p><b> Выберите заголовочный файл </b></p>

1)<p><b> main.h </b></p>

2)<p><b> main.cpp</b></p>

3)<p><b>main.sln</b></p>

Ответ: 1

Комментарий: AP0004002

<p><b>Что такое выполняемый файл?</b></p>

1)<p><b>Файл проекта. Хранит информацию, относящуюся к каждому проекту. </b></p>

2)<p><b>Файл скрипта ресурса для проекта.</b></p>

3)<p><b> файл, содержащий весь необходимый машинный код, требующийся компьютеру для выполнения задания.</b></p>

Ответ: 3

Комментарий: AP0004003

<p><b>В какие годы разрабатывался язык С С</b></p>

1)<p><b>[1969-1973](https://www.google.com/search?client=opera&hs=PUD&sxsrf=ALeKk038_8j55HQ6fqIMy5_l13Z6HaWQaQ:1589209986315&q=%D0%91%D1%8C%D1%91%D1%80%D0%BD+%D0%A1%D1%82%D1%80%D0%B0%D1%83%D1%81%D1%82%D1%80%D1%83%D0%BF&stick=H4sIAAAAAAAAAONgVuLQz9U3MDRPSnnEaMwt8PLHPWEprUlrTl5jVOHiCs7IL3fNK8ksqRQS42KDsnikuLjgmngWscpfmHix5-LEiw0X9ipcWHixCcjYcLH5YiOIdbH5wn4ASKxp1WQAAAA)</b></p>

2)<p><b>1962-1964</b></p>

3)<p><b>1970-1975</b></p>

Ответ: 1

Комментарий: AP0004004

<p><b>Какого уровня язык С</b></p>

1)<p><b> Низкий </b></p>

2)<p><b>Высокий</b></p>

3)<p><b>Сверхвысокий</b></p>

Ответ: 2

Комментарий: AP0004005

<p><b>Что такое внешний файл?</b></p>

1)<p><b> Файл на диске или устройство ввода-вывода (например, клавиатура или дисплей).</b></p>

2)<p><b>Файл, хранящийся в одной папке с проектом</b></p>

3)<p><b> Специально организованная структура данных, распознаваемая компьютером как единое целое</b></p>

Ответ: 1

Комментарий: AP0004006

<p><b>Кто создал язык С++</b></p>

1)<p><b>[Бьёрн Страуструп](https://www.google.com/search?client=opera&hs=PUD&sxsrf=ALeKk038_8j55HQ6fqIMy5_l13Z6HaWQaQ:1589209986315&q=%D0%91%D1%8C%D1%91%D1%80%D0%BD+%D0%A1%D1%82%D1%80%D0%B0%D1%83%D1%81%D1%82%D1%80%D1%83%D0%BF&stick=H4sIAAAAAAAAAONgVuLQz9U3MDRPSnnEaMwt8PLHPWEprUlrTl5jVOHiCs7IL3fNK8ksqRQS42KDsnikuLjgmngWscpfmHix5-LEiw0X9ipcWHixCcjYcLH5YiOIdbH5wn4ASKxp1WQAAAA)</b></p>

2)<p><b>Кен Томпсон и Денис Ритчи</b></p>

3)<p><b>[Джеймс Гослинг](https://www.google.com/search?client=opera&hs=SWD&sxsrf=ALeKk00hN8hLDpwGlkgnOwVD6FH1Z2x4yw:1589210113246&q=%D0%94%D0%B6%D0%B5%D0%B9%D0%BC%D1%81+%D0%93%D0%BE%D1%81%D0%BB%D0%B8%D0%BD%D0%B3&stick=H4sIAAAAAAAAAONgVuLQz9U3SMkzjH_EaMwt8PLHPWEprUlrTl5jVOHiCs7IL3fNK8ksqRQS42KDsnikuLjgmngWsUpfmHJh24WtF3Ze2HOxUeHC5Av7LjZe2H1hx4W9FzYDAKbrrQ5gAAAA) и [Брендан Эйх](https://www.google.com/search?client=opera&hs=Ors&sxsrf=ALeKk03EQ6zgbzCUc-t9btZHBhZffWqQBA:1589210133600&q=%D0%91%D1%80%D0%B5%D0%BD%D0%B4%D0%B0%D0%BD+%D0%AD%D0%B9%D1%85&stick=H4sIAAAAAAAAAONgVuLUz9U3MDE2KrZ8xGjCLfDyxz1hKe1Ja05eY1Tl4grOyC93zSvJLKkUEudig7J4pbi5ELp4FrGKXph4seHC1gt7L2y5sOHCXoULay_svNgKAF18oRZcAAAA)</b></p>

Ответ: 1

Комментарий: AP0004007

<p><b>Когда создавался язык с++</b></p>

1)<p><b>1980-1983 </b></p>

2)<p><b>1982-1985</b></p>

3)<p><b>1980-1984</b></p>

Ответ: 1

Комментарий: AP0004008

<p><b>Выберите характерную черту частотного принципа разработки ПО</b></p>

1)<p><b> учитывает возможность проведения одной и той же работы различными средствами </b></p>

2)<p><b> используется при проектировании ПО, объем которого существенно превосходит имеющийся объем оперативной памяти </b></p>

3)<p><b> выделение в алгоритмах и в обрабатываемых структурах действий и данных по частоте использования </b></p>

Ответ: 3

Комментарий: AP0004009

<p><b>Благодаря каким проблемам ООП получило широкое распространение</b></p>

1)<p><b> Развитие языков и методов программирования не успевало за все более растущими потребностями в прикладных программах. Единственным реальным способом снизить временные затраты на разработку было проектирование новой программной системы на базе разработанных и отлаженных ранее модуле.</b></p>

2)<p><b> Ускорение разработки программного обеспечения требовало решения проблемы упрощения их сопровождения и модификации.</b></p>

3)<p><b> Не все задачи поддаются алгоритмическому описанию по требованиям структурного программирования</b></p>

Ответ: 1, 2, 3

Комментарий: AP0004010

<p><b>Что такое файлы последовательного доступа?</b></p>

1)<p><b>Файлы с операциями последовательного чтения и записи в конец файла </b></p>

2)<p><b>Файлы с операциями чтения и записи по произвольному адресу. </b></p>

3)<p><b>Файлы с операциями чтения и записи по конкретному адресу </b></p>

Ответ: 1

Комментарий: AP0004011

<p><b>Что такое файлы прямого доступа?</b></p>

1)<p><b>Файлы с операциями последовательного чтения и записи в конец файла </b></p>

2)<p><b>Файлы с операциями чтения и записи по произвольному адресу. </b></p>

3)<p><b>Файлы с операциями чтения и записи по конкретному адресу </b></p>

Ответ: 2

Комментарий: AP0004012

<p><b>Что такое ANSI?</b></p>

1)<p><b> Язык программирования, разработанный и реализованный в 1972 году сотрудником фирмы AT&T Bell Laboratories</b></p>

2)<p><b> стандарт [языка C](https://ru.wikipedia.org/wiki/%D0%A1%D0%B8_(%D1%8F%D0%B7%D1%8B%D0%BA_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D1%8F)), опубликованный Американским национальным институтом стандартов</b></p>

3)<p><b> стандарт языка С, принятый международной организацией по стандартизации [ISO](https://ru.wikipedia.org/wiki/ISO)</b></p>

Ответ: 2

Комментарий: AP0004013

<p><b>Сколько крупный переработок потерпел язык С++</b></p>

1)<p><b>три</b></p>

2)<p><b>две</b></p>

3)<p><b>четыре</b></p>

Ответ: 1

Комментарий: AP0004014

<p><b>Что делает оператор NEW</b></p>

1)<p><b>выделяет память на стеке и возвращает указатель на выделенную память</b></p>

2)<p><b>выделяет память в куче и возвращает указатель на нее</b></p>

3)<p><b>создает новый файл </b></p>

Ответ: 2

Комментарий: AP0004015

<p><b>Что делает оператор DELETE</b></p>

1)<p><b> программа освобождает выделенную память. </b></p>

2)<p><b>программа удаляет переменную/функцию</b></p>

3)<p><b>программа удаляет все, что было выведено на консоль</b></p>

Ответ: 1

Комментарий: AP0004016

<p><b>Что делает конструктор?</b></p>

1)<p><b> при вызове очищает память, выделенную под объект </b></p>

2)<p><b> при вызове собирает проект быстрее</b></p>

3)<p><b> при вызове создает и инициализирует объект класса <b></p>

Ответ: 3

Комментарий: AP0004017

<p><b>К достоинствам языка С нельзя отнести:</b></p>

1)<p><b> быстродействие</b></p>

2)<p><b> переносимость</b></p>

3)<p><b> возможности поразрядной обработки</b></p>

4)<p><b> строгость типизацию</b></p>

Ответ: 4

Комментарий: AP0004018

<p><b>Кто дал первое описание языка C?</b></p>

1)<p><b> Американский национальный институт стандартов (ANSI)</b></p>

2)<p><b> Брайан Керниган и Деннис Ритчи</b></p>

3)<p><b> Международная организация по стандартизации (ISO)</b></p>

4)<p><b> Бьерн Страуструп</b></p>

Ответ: 2

Комментарий: AP0004019

<p><b>К потокам вывода, добавленным в C++ относятся</p></b>

1)<p><b> cerr </p></b>

2)<p><b> clog </p></b>

3)<p><b> cbuf </p></b>

4)<p><b> cout </p></b>

Ответ: 1,2,4

Комментарий: AP0004020

<p><b>Объекты взаимодействуют между собой при помощи механизма</p></b>

1)<p><b> обратных вызовов</p></b>

2)<p><b> семафоров</p></b>

3)<p><b> сообщений</p></b>

4)<p><b> инкапсуляции</p></b>

Ответ: 3

Комментарий: AP0004021

<p><b>Где можно описывать анонимные объединения?</p></b>

1)<p><b> только в заголовочных файлах</p></b>

2)<p><b> только внутри функций</p></b>

3)<p><b> только в глобальной области</p></b>

4)<p><b> везде где и именованные </p></b>

Ответ: 4

Комментарий: AP0004022

<p><b>Какое ограничение на длину идентификатора устанавливает стандарт ANSI C?</p></b>

1)<p><b> 32 символа</p></b>

2)<p><b> неограниченно</p></b>

3)<p><b> 31 символ</p></b>

4)<p><b> 16 символов</p></b>

Ответ: 3

Комментарий: AP0004023

<p><b>Язык Си</b></p>

1)<p><b>компилируется в машинный код</b></p>

2)<p><b>компилируется в байт-код</b></p>

3)<p><b>транслируется</b></p>

Ответ: 1

Комментарий: AP0004024

<p><b>Значение какого типа возвращает sizeof</b></p>

1)<p><b>size\_t</b></p>

2)<p><b>long double</b></p>

3)<p><b>unsigned short</b></p>

4)<p><b>int</b></p>

Ответ: 1

Комментарий: AP0004025

<p><b>Какое ограничение установлено на размер COM файла?</b></p>

1)<p><b>64 мб</b></p>

2)<p><b>64 кб</b></p>

3)<p><b>16 мб</b></p>

4)<p><b>16 кб</b></p>

Ответ: 2

Комментарий: AP0004026

<p><b>Использование чего гарантирует инициализацию данных, описанных внутри объекта?</b></p>

1)<p><b>идиомы RAII</b></p>

2)<p><b>виртуальных функций</b></p>

3)<p><b>конструкторов и деструкторов</b></p>

4)<p><b>списков инициализации</b></p>

Ответ: 3  
Комментарий: AP0004027

<p><b>Какая концепция подразумевает запрет доступа к внутренней структуре объекта извне?</b></p>

1)<p><b>иерархия</b></p>

2)<p><b>инкапсуляция</b></p>

3)<p><b>концепция дружественных классов</b></p>

4)<p><b>полиморфизм</b></p>

Ответ 2

Комментарий: AP0004028

<p><b>Какой способ передачи аргументов в функцию отличает C++ от C</b></p>

1)<p><b>использование указателя</b></p>

2)<p><b>использование ссылочного типа</b></p>

3)<p><b>передача по значению</b></p>

4)<p><b>использование свойств классов памяти</b></p>

Ответ: 2

Комментарий: AP0004029

<p><b>Какой способ передачи аргументов в функцию отличает C++ от C</b></p>

1)<p><b>использование указателя</b></p>

2)<p><b>использование ссылочного типа</b></p>

3)<p><b>передача по значению</b></p>

4)<p><b>использование свойств классов памяти</b></p>

Ответ: 2

Комментарий: AP0004030

<p><b>Как имя было первоначально дано C++?</b></p>

1)<p><b>Objective-C</b></p>

2)<p><b>C with classes</b></p>

3)<p><b>D</b></p>

4)<p><b>C+</b></p>

Ответ: 2

Комментарий: AP0004031

<p><b>Какой префикс используется с шестнадцатеричными константами?</b></p>

1)<p><b>0b</b></p>

2)<p><b>u</b></p>

3)<p><b>0b</b></p>

4)<p><b>0x</b></p>

Ответ: 4

Комментарий: AP0004032

<p><b>Какой будет результат данной строки кода? </br>

printf(" %o\n", 336); </b></p>

1)<p><b>336</b></p>

2)<p><b>150</b></p>

3)<p><b>520</b></p>

4)<p><b>-336</b></p>

Ответ: 3

Комментарий: AP0004033

<p><b> Название С++ предложил </b> </p>

1)<p><b> Кэм Томпсон</b> </p>

2)<p><b> Бьерн Страуструп</b> </p>

3)<p><b> Рик Массити </b> </p>

4)<p><b> Дональд Кнут </b> </p>

Ответ:3

Комментарий: AP00004034

<p><b> Укажите объектно-ориентированный язык программирования </b> </p>

1)<p><b> Все варианты ответов </b> </p>

2)<p><b> С++</b> </p>

3)<p><b> Java </b> </p>

4)<p><b> Eiffel </b> </p>

Ответ:1

Комментарий: AP00004035

<p><b>Какой будет результат данной строки кода? </br>

printf(" %d\n", 336); </b></p>

1)<p><b>336</b></p>

2)<p><b>150</b></p>

3)<p><b>520</b></p>

4)<p><b>-336</b></p>

Ответ: 1

Комментарий: AP0004036

<p><b>Какой будет результат данной строки кода? </br>

printf(" %x\n", 336); </b></p>

1)<p><b>336</b></p>

2)<p><b>150</b></p>

3)<p><b>520</b></p>

4)<p><b>-336</b></p>

Ответ: 3

Комментарий: AP0004037

<p><b>Какой будет результат данной строки кода? </br>

printf(" %d\n", -336); </b></p>

1)<p><b>336</b></p>

2)<p><b>150</b></p>

3)<p><b>520</b></p>

4)<p><b>-336</b></p>

Ответ: 4

Комментарий: AP0004038

<p><b>Какой будет вывод у данной функции? int main()

<p><b>{ int age,ves; </b></p>

<p><b>printf(" Vas vozrast i ves.\n");</b></p>

<p><b>scanf\_s(" %d ", &age);//*20*</b></p>

<p><b>scanf\_s("%d", &ves);//*70*</b></p>

<p><b>printf(" Vas Vozrast %d\n" , age);}</b></p> </b></p>

1)<p><b>70</b></p>

2)<p><b>20</b></p>

3)<p><b>25</b></p>

4)<p><b>-75</b></p>

Ответ: 2

Комментарий: AP0004039

<p><b>Какой будет вывод у данной функции? int main()

<p><b>{ int age,ves; </b></p>

<p><b>printf(" Vas vozrast i ves.\n");</b></p>

<p><b>scanf\_s(" %d ", &age);//*20*</b></p>

<p><b>scanf\_s("%d", &ves);//*70*</b></p>

<p><b> printf(" Vas Ves. %d\n", ves); } </b></p> </b></p>

1)<p><b>70</b></p>

2)<p><b>20</b></p>

3)<p><b>25</b></p>

4)<p><b>-75</b></p>

Ответ: 1

Комментарий: AP0004040

<p><b>В какой год началась разработка языка С? </b></p>

1)<p><b>1962</b></p>

2)<p><b>1968</b></p>

3)<p><b>1969</b></p>

4)<p><b>1970</b></p>

Ответ: 3

Комментарий: AP0004041

<p><b>В каком году закончилась разработка языка С?</b></p>

1)<p><b>1972</b></p>

2)<p><b>1973</b></p>

3)<p><b>1974</b></p>

4)<p><b>1970</b></p>

Ответ: 2

Комментарий: AP0004042

<p><b>В каком году началась разработка С++? </b></p>

1)<p><b>1979</b></p>

2)<p><b>1982</b></p>

3)<p><b>1980</b></p>

4)<p><b>1981</b></p>

Ответ: 3

Комментарий: AP0004043

<p><b>В каком году закончилась разработка С++? </b></p>

1)<p><b>1983</b></p>

2)<p><b>1984</b></p>

3)<p><b>1980</b></p>

4)<p><b>1982</b></p>

Ответ: 1

Комментарий: AP0004044

<p><b>Какая из операций является более приоритетной: унарная операция расширения области действия или постфиксальный инкремент? </b></p>

1)<p><b>::</b></p>

2)<p><b>++</b></p>

3)<p><b>--</b></p>

4)<p><b>==</b></p>

Ответ: 1

Комментарий: AP0004045

<p><b>Какая из операций является более приоритетной умножение или постфиксальный инкремент? </b></p>

1)<p><b>::</b></p>

2)<p><b>++</b></p>

3)<p><b>\*</b></p>

4)<p><b>==</b></p>

Ответ: 2

Комментарий: AP0004046

<p><b>Какая из операций является более приоритетной: унарная операция расширения области действия или сложение? </b></p>

1)<p><b>::</b></p>

2)<p><b>+</b></p>

3)<p><b>-</b></p>

4)<p><b>=</b></p>

Ответ: 1

Комментарий: AP0004047

<p><b>Какая из операций является более приоритетной: унарная операция расширения области действия или постфиксальный инкремент? </b></p>

1)<p><b>::</b></p>

2)<p><b>++</b></p>

3)<p><b>--</b></p>

4)<p><b>==</b></p>

Ответ: 1

Комментарий: AP0004048

<p><b>Какая из операций является более приоритетной: деление с присваиванием или постфиксальный декремент? </b></p>

1)<p><b>:=</b></p>

2)<p><b>++</b></p>

3)<p><b>--</b></p>

4)<p><b>==</b></p>

Ответ: 3

Комментарий: AP0004049

<p><b>Какая из операций является более приоритетной: запятая или логическое и? </b></p>

1)<p><b>,</b></p>

2)<p><b>::</b></p>

3)<p><b>&&</b></p>

4)<p><b>==</b></p>

Ответ: 3

Комментарий: AP0004050

## 5 тема

<p><b>Какие из следующих идентификаторов записаны правильно?</p></b>

1. <p><b>\_\_\_\_</b></p>
2. <p><b>Not\_Identify! </b></p>
3. <p><b>OOOO</b></p>
4. <p><b>1st\_year</b></p>

Ответ: 1, 3

Комментарий: AP005001

<p><b>В какой строке сумма размеров типов данных минимальна? </p></b>

1. <p><b>char, float, long int</b></p>
2. <p><b>short, bool, double</b></p>
3. <p><b>unsigned long int, float, double</b></p>
4. <p><b>bool, char, double</b></p>

Ответ: 1

Комментарий: AP005002

<p><b>Выберите верное утверждение: </p></b>

1. <p><b>unsigned int занимает меньше байт памяти, чем signed int </b></p>
2. <p><b>unsigned int не может принимать значение 0, а signed int - может </b></p>
3. <p><b>unsigned int не может содержать отрицательные значения, а signed int - может</b></p>
4. <p><b>unsigned int практически не отличается от signed int</b></p>

Ответ: 3

Комментарий: AP005003

<p><b>С какими из нижеперечисленных типов данных нельзя использовать модификатор unsigned? </p></b>

1. <p><b>float</b></p>
2. <p><b>short</b></p>
3. <p><b>char</b></p>
4. <p><b>double</b></p>

Ответ: 1, 4

Комментарий: AP005004

<p><b>Выберите верные утверждения: </p></b>

1. <p><b>константам, содержащимся в перечисляемом типе enum, значения присваиваются автоматически и эти значения нельзя изменить вручную </b></p>
2. <p><b>константам, содержащимся в перечисляемом типе enum, можно задавать значения вручную </b></p>
3. <p><b>при автоматическом присваивании значений константам в перечисляемом типе enum первой константе всегда присваивается значение 1 </b></p>
4. <p><b>константы, содержащиеся в перечисляемом типе enum, имеют значения типа int</b></p>

Ответ: 2, 4

Комментарий: AP005005

<p><b>Выберите верные утверждения: </p></b>

1. <p><b>С не относится к числу языков, в котором разрешены двойное, тройное и т.д. присваивания</b></p>
2. <p><b>В результате выполнения строки printf(“%d”, 2.4 \* 5) в консоль будет выведено число 12</b></p>
3. <p><b>В результате выполнения строки printf("%d", 2 << 2) в консоль будет выведено число 8</b></p>
4. <p><b>Переменная bool a = !1 < 0 будет содержать false</b></p>

Ответ: 3, 4

Комментарий: AP005006

<p><b>Операция ?: является аналогом: </p></b>

1. <p><b>цикла for()</b></p>
2. <p><b>цикла do while()</b></p>
3. <p><b>конструкции if-else</b></p>
4. <p><b>операции двоичного ИЛИ</b></p>

Ответ: 3

Комментарий: AP005007

<p><b>Наивысшим приоритетом среди ниже перечисленных обладает операция: </p></b>

1. <p><b>операция \*</b></p>
2. <p><b>операция, взятая в скобки</b></p>
3. <p><b>операция ++</b></p>
4. <p><b>операция /</b></p>

Ответ: 2

Комментарий: AP005008

<p><b>К операторам передачи управления относятся операторы: </p></b>

1. <p><b>goto</b></p>
2. <p><b>break</b></p>
3. <p><b>return</b></p>
4. <p><b>if</b></p>

Ответ: 1,2,3

Комментарий: AP005009

<p><b> Как определяется очерёдность выполнения операций, если они имеют одинаковый приоритет?</b></p>

1. <p><b> cогласно свойству ассоциативности</b></p>
2. <p><b> cлева направо независимо от приоритета</b></p>
3. <p><b> cправа налево независимо от приоритета</b></p>

Ответ: 1

Комментарий: AP00050010

<p><b> Какая операция имеет самый высокий приоритет? </b></p>

1. <p><b> операция указания области видимости (::)</b></p>
2. <p><b> операция взятия адреса (&)</b></p>
3. <p><b> побитная операция ИЛИ ( | ) </b></p>
4. <p><b> операция выбора элемента по ссылке (.)</b></p>

Ответ: 1

Комментарий: AP00050011

<p><b> Операторы можно разделить на следующие группы:</b></p>

1. <p><b> арифметические, реляционные, присваивания</b></p>
2. <p><b> арифметические, реляционные, присваивания, логические, побитовые</b></p>
3. <p><b> арифметические, реляционные, побитовые, логические</b></p>
4. <p><b> арифметические, реляционные, присваивания, логические</b></p>

Ответ: 2

Комментарий: AP0005012

<p><b> К какому виду можно отнести следующие операторы : /, ++, % </b></p>

1. <p><b> реляционные</b></p>
2. <p><b> побитовые</b></p>
3. <p><b> арифметические</b></p>
4. <p><b> логические</b></p>

Ответ: 3

Комментарий: AP0005013

<p><b> Для чего использует оператор Сast </b></p>

1. <p><b> для преобразования одного типа данных в другой</b></p>
2. <p><b> для заключения в группу объявлений классов, переменных и функций для использования только в текущем модуле</b></p>
3. <p><b> для передачи управления на помеченный оператор</b></p>

Ответ: 1

Комментарий: AP0005014

<p><b> Для чего используется оператор & </b></p>

1. <p><b> для определения адреса переменной</b></p>
2. <p><b> для определения указателя на переменную</b></p>
3. <p><b> для обращения к члену структуры</b></p>

Ответ: 1

Комментарий: AP0005015

<p><b> Для чего предназначен оператор namespace? </b></p>

1. <p><b> для использования классов, переменных и функций из других модулей программы без использования заголовочных файлов </b></p>
2. <p><b> для заключения в группу объявлений классов, переменных и функций в отдельный контекст со своим именем</b></p>
3. <p><b> для заключения в группу объявлений классов, переменных и функций для использования только в текущем модуле</b></p>

Ответ : 2

Комментарий: AP0005016

<p><b> Если в арифметическом выражении участвуют целый и вещественный операнды, тогда </b></p>

1. <p><b> происходит ошибка компиляции</b></p>
2. <p><b> целый тип приводится к вещественному</b></p>
3. <p><b> вещественный тип приводится к целому</b></p>

Ответ 2

Комментарий: AP0005017

<p><b> К операторам передачи управления относятся: </b></p>

1. <p><b> break, continue, return, goto, throw</b></p>
2. <p><b> break, continue, cast</b></p>
3. <p><b> goto, namespace, throw, continue</b></p>
4. <p><b> continue, cast</b></p>

Ответ: 1

Комментарий: AP0005018

<p><b> К какому виду можно отнести следующие операторы: ==, >=, < </b></p>

1. <p><b> арифметические</b></p>
2. <p><b> реляционные</b></p>
3. <p><b> логические</b></p>
4. <p><b> побитовые</b></p>

Ответ: 2

Комментарий: AP0005019

<p><b> Для чего используется оператор continue? </b></p>

1. <p><b> для пропуска всех операторов, оставшихся до конца тела цикла, и передачи управления на модификацию параметров цикла и начало следующей итерации</b></p>
2. <p><b> для завершения выполнения функции и передает управление в точку ее вызова</b></p>
3. <p><b> для передачи управления на помеченный оператор</b></p>

Ответ: 1

Комментарий: AP0005020

<p><b> Цикл for называется циклом с …</b></p>

1. <p><b> с заданным количеством шагов</b></p>
2. <p><b> с предусловием</b></p>
3. <p><b> с постусловием</b></p>
4. <p><b> с предусловием и постусловием</b></p>

Ответ: 1

Комментарий: AP0005021

<p><b> Каким может быть цикл for? </b></p>

1. <p><b> с предусловием</b></p>
2. <p><b> с предусловием и постусловием</b></p>
3. <p><b> декрементным и инкрементным</b></p>

Ответ: 3

Комментарий: AP0005022

<p><b> Оператор цикла for предназначен для циклических алгоритмов, когда переменная цикла явно выражена и изменяется от начального до конечного значений с … </b></p>

1. <p><b> переменным шагом</b></p>
2. <p><b> постоянным шагом</b></p>
3. <p><b> показателем</b></p>

Ответ: 2

Комментарий: AP0005023

<p><b> Цикл заканчивается, когда параметр:</b></p>

1. <p><b> принимает определенное значение</b></p>
2. <p><b> принимает отрицательное значение</b></p>
3. <p><b> исчерпал себя</b></p>

Ответ: 1

Комментарий: AP0005024

<p><b> Какая операция имеет самый низкий приоритет?</b></p>

<p><b> операция запятая (,)</b></p>

<p><b> операция присваивания с побитовой операцией И (&=) </b></p>

<p><b> операция деления с присваиванием (/=)</b></p>

<p><b> операция сложения (+) </b></p>

Ответ: 1

Комментарий: AP0005025

<p><b> Для чего используется оператор sizeof ?</b></p>

<p><b> для определения размера переменной</b></p>

<p><b> для выделения памяти</b></p>

<p><b> для освобождения памяти</b></p>

Ответ: 1

Комментарий: AP0005026

<p><b> Какая из следующих операций выполняется справа налево? </b></p>

<p><b> =</b></p>

<p><b> [ ] </b></p>

<p><b> -> </b></p>

Ответ: 1

Комментарий: AP0005027

<p><b> Что такое ассоциативность?</b></p>

<p><b> очерёдность выполнения операций в выражении, при условии, что в выражении нет явного указания порядка следования выполнения операций</b></p>

<p><b> направление выполнения операций в случае, если операции имеют одинаковый приоритет</b></p>

<p><b> направление выполнения операций в случае, если в выражении присутствуют 3 и более операции</b></p>

Ответ: 2

Комментарий: AP0005028

<p><b> Какой горячей клавишей можно поставить точку останова в Visual Studio </b></p>

1. <p><b> F7 </b></p>

2. <p><b> F3 </b></p>

3. <p><b> F9 </b></p>

4. <p><b> Home </b></p>

Ответ: 3

Комментарий: AP0005029

<p><b> Шаг с заходом в отладчике осуществляется с помощью клавиши</b></p>

1. <p><b> F10 </b></p>

2. <p><b> F11 </b></p>

3. <p><b> F9 </b></p>

4. <p><b> F6 </b></p>

Ответ: 2

Комментарий: AP0005030

<p><b> Шаг с обходом в отладчике осуществляется с помощью клавиши</b></p>

1. <p><b> F10 </b></p>

2. <p><b> F11 </b></p>

3. <p><b> F9 </b></p>

4. <p><b> F6 </b></p>

Ответ: 1

Комментарий: AP0005031

<p><b> Первое, что нужно сделать для написания программы</b></p>

1. <p><b> Объявить переменные </b></p>

2. <p><b> Открыть новый файл </b></p>

3. <p><b> Скомпилировать её</b></p>

4. <p><b> Поставить точку останова </b></p>

Ответ: 2

Комментарий: AP0005032

<p><b> Минимальное количество выполнений цикла do-while </b></p>

1. <p><b> 0 </b></p>

2. <p><b> 1 </b></p>

3. <p><b> Сколько укажет программист </b></p>

Ответ: 2

Комментарий: AP0005033

<p><b> Компилятор C/C++ является регистрозависимым</b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 1

Комментарий: AP0005034

<p><b> Чем отличаются команды Build и Rebuild All ? </b></p>

1. <p><b> Rebuild All заново компилирует и компонует все файлы проекта, не проверяя даты создания файлов </b></p>

2. <p><b> Build заново компилирует и компонует все файлы проекта, не проверяя даты создания файлов </b></p>

3. <p><b> Ничем не отличаются</b></p>

4. <p><b> Rebuild All работает быстрее, чем Build</b></p>

Ответ: 1

Комментарий: AP0005035

<p><b> Главная функция в программе на языке Си должна называться</b></p>

1. <p><b> principal() </b></p>

2. <p><b> main() </b></p>

3. <p><b> major() </b></p>

4. <p><b> Без разницы </b></p>

Ответ: 2

Комментарий: AP0005036

<p><b> Отличительным признаком имени функции служат </b></p>

1. <p><b> [] </b></p>

2. <p><b> “ ” </b></p>

3. <p><b> () </b></p>

4. <p><b> Прописные буквы </b></p>

Ответ: 3

Комментарий: AP0005037

<p><b> Каждый набор операторов в языке Си оканчивается символом </b></p>

1. <p><b> // </b></p>

2. <p><b> …(многоточие) </b></p>

3. <p><b> : (двоеточие)</b></p>

4. <p><b> ;(точка с запятой) </b></p>

Ответ: 4

Комментарий: AP0005038

<p><b> На что указывает символ “#” в языке Си</b></p>

1. <p><b> Адрес переменной</b></p>

2. <p><b> Ссылку на объект </b></p>

3. <p><b> Что строка должны быть обработана «препроцессором» </b></p>

4. <p><b> Указатель </b></p>

Ответ: 3

Комментарий: AP0005039

<p><b> Синтаксис многострочного комментария в языке Си </b></p>

1. <p><b> //комментарий </b></p>

2. <p><b>/\*комментарий\*/ </b></p>

3. <p><b> <--!комментарий--> </b></p>

4. <p><b> /комментарий/ </b></p>

Ответ: 2

Комментарий: AP0005040

<p><b> Синтаксис однострочного комментария в языке Си </b></p>

1. <p><b> //комментарий </b></p>

2. <p><b>/\*комментарий\*/ </b></p>

3. <p><b> <--!комментарий--> </b></p>

4. <p><b> /комментарий/ </b></p>

Ответ: 1

Комментарий: AP0005041

<p><b> Начало и конец функции в языке Си определяется символами </b></p>

1. <p><b> { } </b></p>

2. <p><b> [] </b></p>

3. <p><b> / / </b></p>

4. <p><b> | | </b></p>

Ответ: 1

Комментарий: AP0005042

<p><b> Выберите правильные имена переменных на языке Си </b></p>

1. <p><b> man1 </b></p>

2. <p><b> 1man </b></p>

3. <p><b> $A\*\* </b></p>

4. <p><b> hElLo123 </b></p>

Ответ: 1, 4

Комментарий: AP0005043

<p><b> Символ новой строки в языке Си обозначается </b></p>

1. <p><b> \t </b></p>

2. <p><b> \n </b></p>

3. <p><b> \v </b></p>

4. <p><b> \s </b></p>

Ответ: 2

Комментарий: AP0005044

<p><b> Вывести строку в языке Си можно с помощью функции </b></p>

1. <p><b> printf()</b></p>

2. <p><b> message() </b></p>

3. <p><b> print() </b></p>

4. <p><b> cout </b></p>

Ответ: 1

Комментарий: AP0005045

<p><b> Для быстрого перемещения по строке нужно нажать сочетание клавиш: </b></p>

1. <p><b> SHIFT + стрелка вверх/вниз </b></p>

2. <p><b> ALT + стрелка влево/вправо </b></p>

3. <p><b> CTRL + стрелка влево/вправо </b></p>

4. <p><b> CTRL + стрелка вверх/вниз </b></p>

Ответ: 3

Комментарий: AP0005046

<p><b> Для удаления слова, расположенного справа от курсора, нужно нажать сочетание клавиш: </b></p>

1. <p><b> SHIFT + стрелка вверх/вниз </b></p>

2. <p><b> ALT + DELETE </b></p>

3. <p><b> BACKSPACE + SHIFT</b></p>

4. <p><b> CTRL + DELETE </b></p>

Ответ: 4

Комментарий: AP0005047

<p><b> Для удаления слова, расположенного слева от курсора, нужно нажать сочетание клавиш: </b></p>

1. <p><b> SHIFT + стрелка вверх/вниз </b></p>

2. <p><b> ALT + DELETE </b></p>

3. <p><b> CTRL + BACKSPACE</b></p>

4. <p><b> CTRL + DELETE </b></p>

Ответ: 3

Комментарий: AP0005048

<p><b> Выберите операторы передачи управления </b></p>

1. <p><b> goto </b></p>

2. <p><b> continue </b></p>

3. <p><b> break </b></p>

4. <p><b> control </b></p>

Ответ: 1, 2, 3

Комментарий: AP0005049

<p><b> Выберите черты, характерные для оператора continue </b></p>

1. <p><b> Он использует метки </b></p>

2. <p><b> Переходит к следующей итерации цикла </b></p>

3. <p><b> Выходит из цикла </b></p>

4. <p><b> Пропускает все операторы, оставшиеся до конца тела функции</b></p>

Ответ: 2, 4

Комментарий: AP0005050

<p><b> Операторами цикла в языке Си являются: </b></p>

1. <p><b> operator </b></p>

2. <p><b> do-while </b></p>

3. <p><b> while </b></p>

4. <p><b> for </b></p>

Ответ: 2, 3, 4

Комментарий: AP0005051

<p><b> Каждое сообщение об ошибке начинается с.. </b></p>

1. <p><b> Приветствия </b></p>

2. <p><b> Имени исходного файла </b></p>

3. <p><b> Номер строки, где обнаружена ошибка </b></p>

1. <p><b> Слова “error” </b></p>

Ответ: 2

Комментарий: AP0005052

<p><b> Обнаружить ошибки в программе можно с помощью </b></p>

1. <p><b> Транслятора </b></p>

2. <p><b> Компилятора </b></p>

3. <p><b> Отладчика(debugger) </b></p>

4. <p><b> С помощью функции error() </b></p>

Ответ: 3

Комментарий: AP0005053

**<p><b>Цикл с постусловием, где истинность выражения, проверяющего условие** **проверяется после выполнения блока операций**, **заключенного в фигурные скобки.</b></p>**

1. **<p><b>**for**</b></p>**

2. **<p><b>**while**</b></p>**

3. **<p><b>**do…while**</b></p>**

Ответ: 3

Комментарий: AP0005054

**<p><b>Какой оператор позволяет пропустить часть операторов тела цикла и начать новую итерацию? </b></p>**

1. **<p><b>**break**</b></p>**

2. **<p><b>**continue**</b></p>**

3. **<p><b>**try…catch**</b></p>**

4. **<p><b>**return**</b></p>**

Ответ: 2

Комментарий: AP0005055

**<p><b>Какой тип данных не является вещественным?</b></p>**

1. **<p><b>**long double**</b></p>**

2. **<p><b>**short**</b></p>**

3. **<p><b>**double**</b></p>**

4. **<p><b>**float **</b></p>**

Ответ: 2

Комментарий: AP0005056

**<p><b>Сколько байт в Windows занимает wchar\_t?</b></p>**

1. **<p><b>**1**</b></p>**

2. **<p><b>**2**</b></p>**

3. **<p><b>**4**</b></p>**

4. **<p><b>**16**</b></p>**

5. **<p><b>**32**</b></p>**

Ответ: 2

Комментарий: AP0005055

**<p><b>Ключевое слово языка С++, которое позволяет объявить новое имя существующего типа</b></p>**

1. **<p><b>**define**</b></p>**

2. **<p><b>**const**</b></p>**

3. **<p><b>**static**</b></p>**

4. **<p><b>**typedef**</b></p>**

Ответ: 4

Комментарий: AP0005056

**<p><b>В какой из функций допущена ошибка?</b></p>**

1. **<p><b>**int f1() { return 1;}</**b></p>**

2. **<p><b>**short f2() { return 1;}**</b></p>**

3. **<p><b>**double f3() { return 1;}**</b></p>**

4. **<p><b>**void f4() { return 1;}**</b></p>**

Ответ: 4

Комментарий: AP0005057

<p><b>Как фирма Microsoft в своей версии языка С++ расширила область видимости переменной, объявленной внутри оператора for?</b></p>

1. <p><b>Внутри цикла и за его пределы</b></p>

2. <p><b>До конца блока этого for</b></p>

3. <p><b>Только в блоке объявления переменных </b></p>

Ответ: 2

Комментарий: AP0005058

<p><b> Для чего используется операция запятая внутри оператора for?</b></p>

1. <p><b>Чтобы отделить друг от друга объявление переменных, условие, инкремент/декремент счётчика</b></p>

2. <p><b> Чтобы в тех места, где допустимо использование только одного оператора, размещать несколько операторов</b></p>

3. <p><b> Для размещения оператора на несколько строк кода</b></p>

Ответ: 2

Комментарий: AP0005059

<p><b>Что делает функция abort?</b></p>

1. <p><b>аварийно завершает работу программы</b></p>

2. <p><b>генерирует SIGABTR сигнал </b></p>

3. <p><b>вызывает завершение программы без вызова деструкторов для статических или динамических объектов </b></p>

4. <p><b>вызывает завершение программы без вызова функции atexit </b></p>

Ответ: 1, 2, 3, 4

Комментарий: AP0005060

<p><b>Что относится к лексемам языка Си?</b></p>

1. <p><b>Комментарии</b></p>

2. <p><b>Разделители</b></p>

3. <p><b>Константы </b></p>

4. <p><b>Знаки операций</b></p>

Ответ: 2, 3, 4

Комментарий: AP0005061

<p><b> Какие символы в идентификаторе являются значимыми? </b></p>

1. <p><b>Все символы идентификатора </b></p>

2. <p><b>Первые 32 символа </b></p>

3. <p><b>Первый символ идентификатора </b></p>

Ответ: 2

Комментарий: AP0005062

<p><b> Выберите строки, где все слова являются ключевыми в языке Си</b></p>

1. <p><b>Volatile, return, false </b></p>

2. <p><b>Include, inline, catch </b></p>

3. <p><b>Auto, map, mutable </b></p>

4. <p><b>Vector, char, void </b></p>

5. <p><b>For, goto, break </b></p>

Ответ: 1, 5

Комментарий: AP0005063

<p><b> В какой библиотеке содержатся функции форматированного ввода и вывода данных?</b></p>

1. <p><b>Stdio.lib </b></p>

2. <p><b>Conio.h </b></p>

3. <p><b>Iostream.h </b></p>

4. <p><b>Iomanip </b></p>

Ответ: 1

Комментарий: AP0005064

<p><b> Чем является clog в стандартной библиотеке <iostream>? </b></p>

1. <p><b>Поток стандартной ошибки </b></p>

2. <p><b> Стандартный поток ввода</b></p>

3. <p><b>Буферизируемый поток стандартных ошибок </b></p>

4. <p><b>Стандартный поток вывода </b></p>

Ответ: 3

Комментарий: AP0005065

<p><b> Для чего используется манипулятор endl? </b></p>

1. <p><b>Для перехода на новую строку</b></p>

2. <p><b> Для переноса строки и сброса буферов потока вывода</b></p>

3. <p><b>Для горизонтальной и вертикальной табуляции </b></p>

Ответ: 3

Комментарий: AP0005066

<p><b>Какой флаг форматированного вывода отвечает за вывод вещественных чисел в экспоненциальной форме? </b></p>

1. <p><b>Fixed </b></p>

2. <p><b> Showpos</b></p>

3. <p><b>Scientific</b></p>

4. <p><b>Dec</b></p>

Ответ: 3

Комментарий: AP0005067

<p><b>Для чего используется данная конструкция: cout.setf(ios::…) </b></p>

1. <p><b>Для установки флага форматированного вывода </b></p>

2. <p><b> Для снятия флага форматированного вывода</b></p>

3. <p><b>Для вывод вещественных чисел в экспоненциальной форме </b></p>

4. <p><b>Для вывода величин в десятичной форме исчисления</b></p>

Ответ: 1

Комментарий: AP0005068

<p><b>Какой из перечисленных символов нужно использовать для установки нескольких флагов форматированного ввода/вывода</b></p>

1. <p><b>;</b></p>

2. <p><b> &</b></p>

3. <p><b>|</b></p>

4. <p><b>&& </b></p>

Ответ: 3

Комментарий: AP0005069

<p><b>Работают ли файлы проекта одной IDE в другой IDE </b></p>

1. <p><b>Работают</b></p>

2. <p><b> Не работают </b></p>

3. <p><b>Зависит от самих IDE</b></p>

Ответ: 2

Комментарий: AP0005070

<p><b>Что такое литералы?</b></p>

1. <p><b>Фиксированное значение, которое записывается непосредственно в исходном коде </b></p>

2. <p><b> Данные, с которыми работает выражение</b></p>

3. <p><b>Символы, которые сообщают компилятору выполнить определенные математические или логические манипуляции </b></p>

Ответ: 1, 2

Комментарий: AP0005071

<p><b>Почему переменные не могут быть типа void? </b></p>

1. <p><b>Переменные могут быть типа void </b></p>

2. <p><b> Потому что void используется только для функций </b></p>

3. <p><b>Потому что он обозначает “отсутствие любого типа данных” </b></p>

Ответ: 3

Комментарий: AP0005072

<p><b>Какое значение имеет данный суффикс численного литерала: l или L? </b></p>

1. <p><b>long long </b></p>

2. <p><b> long</b></p>

3. <p><b>long double </b></p>

4. <p><b>unsigned long </b></p>

Ответ: 2, 3

Комментарий: AP0005073

<p><b>Какие из ключевых слов являются типами оператора управления порядком выполнения программы “переход” ? </b></p>

1. <p><b> Break</b></p>

2. <p><b>Continue </b></p>

3. <p><b> Switch</b></p>

4. <p><b> Goto</b></p>

Ответ: 1, 2, 4

Комментарий: AP0005074

<p><b>Выберете функции оператора break? </b></p>

1. <p><b> В контексте оператора switch обычно используется в конце каждого кейса для его завершения</b></p>

2. <p><b>В контексте циклов используется для завершения работы цикла раньше времени </b></p>

3. <p><b>Может использоваться для выхода из бесконечного цикла </b></p>

4. <p><b> Завершает выполнение всей функции, в которой находится цикл, а выполнение продолжается в точке после вызова функции:</b></p>

Ответ: 1, 2, 3

Комментарий: AP0005075

<p><b>С какими операндами может работать оператор деления с остатком %? </b></p>

1. <p><b> Только с целочисленными операндами</b></p>

2. <p><b>С любыми операндами </b></p>

3. <p><b> С целочисленными и вещественными операндами</b></p>

Ответ: 1

Комментарий: AP0005076

**<p><b>Описание функции состоит из </b></p>**

1. <p><b>заголовка</b></p>

2. <p><b>переменных</b></p>

3. <p><b>тела</b></p>

4. <p><b>операторов</b></p>

Ответ: 1,3

Комментарий: AP0005077

**<p><b>Чем оканчивается каждый оператор в функциях </b></p>**

1. <p><b>пробелом</b></p>

2. <p><b>символом «точка с запятой»</b></p>

3. <p><b>фигурными скобками </b></p>

Ответ: 2

Комментарий: AP0005078

**<p><b>Какое расширение имеет заголовочный файл </b></p>**

1. <p><b>.h</b></p>

2. <p><b>.cpp</b></p>

3. <p><b>.sln</b></p>

Ответ: 1

Комментарий: AP0005079

**<p><b>Отличием команд Build и Rebuild All является</b></p>**

1. <p><b> Команда Rebuild All заново не компилирует и не компонует все файлы проекта </b></p>

2. <p><b> Команда Rebuild All проверяет даты создания файлов </b></p>

3. <p><b> Команда Rebuild All не проверяет даты создания файлов, заново компилирует и компонует все файлы проекта</b></p>

Ответ: 3

Комментарий: AP0005080

**<p><b>В каком случае можно запустить программу</b></p>**

1. <p><b>Без предупреждений, но с одной ошибкой</b></p>

2. <p><b>С одними предупреждениями </b></p>

3. <p><b>Без предупреждений и ошибок</b></p>

Ответ: 2, 3

Комментарий: AP0005081

**<p><b>Если все функции, имеющие прототипы в файле МАТН.Н, имеют формальные параметры типа double и возвращают значение типа double, но программа вызовет какую-либо из этих функций с аргументом типа float то </b></p>**

1. <p><b>компилятор выдаст предупреждение</b></p>

2. <p><b>компилятор выдаст ошибку</b></p>

3. <p><b>компилятор ничего не выдаст и запустит программу</b></p>

Ответ: 1

Комментарий: AP0005082

**<p><b>Что выполняет опция Replace All</b></p>**

1. <p><b>подстановку новой строки</b></p>

2. <p><b>непрерывно просматривает весь текст </b></p>

3. <p><b>поиск следующего вхождения строки</b></p>

4. <p><b>осуществляет поиск и замену требуемой строки</b></p>

Ответ: 2,4

Комментарий: AP0005083

**<p><b>Что выполняет опция Find Next** **</b></p>**

1. <p><b>подстановку новой строки</b></p>

2. <p><b>непрерывно просматривает весь текст </b></p>

3. <p><b>поиск следующего вхождения строки</b></p>

4. <p><b>осуществляет поиск и замену требуемой строки</b></p>

Ответ: 3

Комментарий: AP0005084

**<p><b>Что выполняет опция Replace </b></p>**

1. <p><b>подстановку новой строки</b></p>

2. <p><b>непрерывно просматривает весь текст </b></p>

3. <p><b>поиск следующего вхождения строки</b></p>

4. <p><b>осуществляет поиск и замену требуемой строки</b></p>

Ответ: 1

Комментарий: AP0005085

## 6 тема

<p><b>Что такое идентификатор?</b></p>

1. <p><b>Идентификатор — это имя переменной, функции, класса или другого объекта в C++</b></p>
2. <p><b>Идентификатор - это ключевое слово, например bool, int и тд</b></p>
3. <p><b>Идентификатор - это значение, которое мы задаем переменной</b></p>

Ответ: 1

Комментарий: AP0006001

<p><b>Выберите правильное указание идентификатора:</b></p>

1. <p><b>int my p;</b></p>
2. <p><b>double .oranges;</b></p>
3. <p><b>char letter;</b></p>
4. <p><b>int struct;</b></p>

Ответ: 3

Комментарий: AP0006002

<p><b>Выберите верные утверждения об идентификаторах:</b></p>

1. <p><b>Идентификатор не может быть ключевым словом. Ключевые слова зарезервированы.</b></p>
2. <p><b>Идентификатор может состоять только из букв (нижнего или верхнего регистра), цифр или символов подчеркивания. Это означает, что все другие символы и пробелы — запрещены.</b></p>
3. <p><b>Идентификатор должен начинаться с буквы (нижнего или верхнего регистра). Он не может начинаться с цифры.</b></p>
4. <p><b>C++ различает нижний регистр от верхнего. nvalue отличается от nValue и отличается от NVALUE.</b></p>
5. <p><b>Все варианты верны</b></p>

Ответ: 5

Комментарий: AP0006003

<p><b>Идентификатор не может начинаться с :</b></p>

1. <p><b>Буквы нижнего или верхнего регистра</b></p>
2. <p><b>С ключевого слова</b></p>
3. <p><b>С цифры</b></p>

Ответ: 2, 3

Комментарий: AP0006004

<p><b>int integer\_value = 10;

printf("%D", integer\_value);

Что выведет данный код?</b></p>

1. <p><b>Сообщение об ошибке</b></p>
2. <p><b>D</b></p>
3. <p><b>10</b></p>

Ответ: 2

Комментарий:AP0006005

<p><b>Что такое ключевые слова в С++.Выберите правильный вариант:</b></p>

1. <p><b>C++ имеет зарезервированный набор из слов для собственного использования. Эти слова называются ключевыми словами, каждое из которых имеет свое особое значение.</b></p>
2. <p><b>Ключевое слово — это имя переменной, функции, класса или другого объекта в C++</b></p>

Ответ: 1

Комментарий: AP0006006

<p><b>Как выглядят ключевые слова в С++</b></p>

1. <p><b>/r /t //</b></p>
2. <p><b>class, int, do </b></p>
3. <p><b>ivalue, dnumber, cletter</b></p>

Ответ:2

Комментарий AP0006007

<p><b>int, long, short, unsigned, char, float, double. Какие типы данных являются целочисленными?</b></p>

1. <p><b>int</b></p>
2. <p><b>long</b></p>
3. <p><b>short</b></p>
4. <p><b>unsigned</b></p>
5. <p><b>char</b></p>
6. <p><b>float</b></p>
7. <p><b>double</b></p>
8. <p><b>Все</b></p>

Ответ: 1,2,3,4

Комментарий: AP0006008

<p><b>Какой тип данных в C описывает символы?</b></p>

1. <p><b>int</b></p>
2. <p><b>char</b></p>
3. <p><b>float</b></p>
4. <p><b>double</b></p>
5. <p><b>void</b></p>

Ответ: 2

Комментарий: AP0006009

<p><b>Как описываются символы в C?</b></p>

1. <p><b>“ “ - двойные кавычки</b></p>
2. <p><b>` ` - обратные кавычки</b></p>
3. <p><b>‘ ’ - одиночные кавычки</b></p>

Ответ: 3

Комментарий: AP0006010

<p><b>Как выглядят управляющие в символы в C?</b></p>

1. <p><b>/t, /b, /n.</b></p>
2. <p><b>|, &, !=.</b></p>
3. <p><b>-, +, =</b></p>

Ответ: 1

Комментарий: AP0006011

<p><b>Выберите верные утверждения:</b></p>

1. <p><b>Значение константы можно изменять</b></p>
2. <p><b>Значения константы и переменной можно изменять</b></p>
3. <p><b>Значение переменной можно изменять</b></p>
4. <p><b>Значение константы нельзя изменять</b></p>
5. <p><b>Значения константы и переменной нельзя изменять</b></p>

Ответ: 3,4

Комментарий: AP0006012

<p><b>Запись unsigned int означает:</b></p>

1. <p><b>Что число будет положительным (от 0 и больше)</b></p>
2. <p><b>Что число будет отрицательным (до 0 и меньше)</b></p>
3. <p><b>Слово unsigned не влияет на значения, которые принимает int</b></p>

Ответ: 1

Комментарий:AP0006013

<p><b>Тип данных переменной указывается:</b></p>

1. <p><b>При объявлении переменной</b></p>
2. <p><b>При инициализации переменной</b></p>
3. <p><b>Не обязательно указывать </b></p>
4. <p><b>Нет верных вариантов ответа</b></p>

Ответ: 1,2

Комментарий: AP0006014

<p><b>Выберите верные утверждения:</b></p>

1. <p><b>int - это единственный целочисленный тип данных</b></p>
2. <p><b>char - может хранить массивы данных</b></p>
3. <p><b>double хранит числа с плавающей точкой</b></p>
4. <p><b>short и int хранят только целые числа </b></p>
5. <p><b>bool принимает только 2 значения</b></p>

Ответ: 3,4,5

Комментарий: AP0006015

<p><b>

int iresult = 0;

float fx = 7.0;

float fy = 2.0;

iresult = 4.0 + fx/fy;

Результатом операции является:

</b></p>

1. <p><b>3.5</b></p>
2. <p><b>7.5</b></p>
3. <p><b>7</b></p>
4. <p><b>8</b></p>

Ответ: 3

Комментарий: AP0006016

<p><b> int num = 5; double apples = 10;

К какому типу преобразуется результат операции num\*apples</b></p>

1. <p><b>int</b></p>
2. <p><b>double</b></p>

Ответ: 2

Комментарий: AP0006017

<p><b>1 #include <stdio.h>

2 main

3 {

4 float g; h;

5 float tax, rate;

6 g = e21;

7 tax = rate\*g;

8 }

Укажите в каких строчках допущены ошибки</b></p>

1. <p><b>1</b></p>
2. <p><b>2</b></p>
3. <p><b>4</b></p>
4. <p><b>5</b></p>
5. <p><b>6</b></p>
6. <p><b>7</b></p>
7. <p><b>Ошибок нет</b></p>

Ответ: 2, 4, 6

Комментарий: AP0006018

<p><b>short dogs, cats = 92;

Выберите верные утверждения</b></p>

1. Переменная cats объявлена, а переменная dogs инициализирована
2. Переменная dogs объявлена, а переменная cats инициализирована
3. Нет верного утверждения

Ответ: 2

Комментарий: AP0006019

<p><b>Выберите верные утверждения:</b></p>

1. Целые числа не имеют дробной части, в то время как числа с плавающей точкой могут представлять как целые, так и дробные числа.
2. Числа с плавающей точкой дают возможность представлять величины из более широкого диапазона, чем целые.
3. При некоторых арифметических операциях, например при вычитании одного большого числа из другого, использование чисел с плавающей точкой приводит к большей потере точности.
4. Операции над числами с плавающей точкой выполняются, как правило, медленнее, чем операции над целыми числами.
5. Все утверждения верны

Ответ: 5

Комменатрий: AP0006020

<p><b>enum air\_supply {EMPTY,

USEABLE,

FULL=5 } instructor\_tank

enum air\_supply student\_tank;

instructor\_tank = FULL;

student\_tank = EMPTY;

Чему будет равна переменная student\_tank

</b></p>

1. <p><b>0</b></p>
2. <p><b>5</b></p>
3. <p><b>1</b></p>

Ответ: 1

Комментарий AP0006021

<p><b>В С++ есть способы определения констант. Укажите их:</b></p>

1. <p><b>const </b></p>
2. <p><b>#define</b></p>
3. <p><b>unsigned</b></p>
4. <p><b>class</b></p>
5. <p><b>volatile</b></p>

Ответ: 1,2

Комментарий: AP0006022

<p><b>Выберите правильную инициализацию переменной letter</b></p>

1. <p><b>char letter</b></p>
2. <p><b>char 12.letter = “k”;</b></p>
3. <p><b>char letter = ‘S’;</b></p>

Ответ: 3

Комментарий: AP0006023

<p><b>К каким типам данных можно применить модификатор unsigned?</b></p>

1. <p><b>char, short, int, long</b></p>
2. <p><b>char, int, float, double</b></p>
3. <p><b>short, int, float, double</b></p>

Ответ: 1

Комментарий: AP0006024

<p><b>Диапазон значений для типа данных int:</b></p>

1. <p><b>от -120 до 127</b></p>
2. <p><b>от -32768 до 32767</b></p>
3. <p><b>от 0 до 65535</b></p>

Ответ: 2

Комментарий: AP0006025

<p><b>Как обозначается перечисляемый тип данных?</b></p>

1. <p><b>double</b></p>
2. <p><b>unsigned</b></p>
3. <p><b>enum</b></p>

Ответ: 3

Комментарий: AP0006026

<p><b>Какая инициализация является верной?</b></p>

1. <p><b>unsigned short ribs = 6;</b></p>
2. <p><b>unsigned long apples = -1;</b></p>
3. <p><b>unsigned short .dots;</b></p>

Ответ: 1

Комментарий: AP0006027

<p><b>Как можно объявить константу в C++?</b></p>

1. <p><b>Директива препроцессора #define</b></p>
2. <p><b>Ключевое слово const</b></p>
3. <p><b>Ключевое слово enum</b></p>
4. <p><b>Все утверждения верные</b></p>

Ответ: 1, 2

Комментарий: AP0006028

<p><b>Какая инициализация является верной?</b></p>

1. <p><b>float planck = 6.63e-34;</b></p>
2. <p><b>double apples = 1.56Е+ 12</b></p>
3. <p><b>float links = e21</b></p>

Ответ: 1

Комментарий: AP0006029

<p><b>Какая инициализация является верной?</b></p>

1. <p><b>float planck = 6.63e-34;</b></p>
2. <p><b>double apples = 1.56Е+ 12</b></p>
3. <p><b>float links = e21</b></p>

Ответ: 1

Комментарий: AP0006030

<p><b>Какое число будет записано в переменную result?

int a = 5;

double k = 0.3

double result = (int) a/k</b></p>

1. <p><b>16</b></p>
2. <p><b>16.7</b></p>
3. <p><b>1.5</b></p>

Ответ: 1

Комментарий: AP0006031

<p><b>Какое ключевое слово означает, что переменная может неожиданно измениться в результате каких-то внешних событий?</b></p>

1. <p><b>enum</b></p>
2. <p><b>volatile</b></p>
3. <p><b>const</b></p>

Ответ: 2

Комментарий: AP0006032

<p><b>Какие нижеперечисленные ключевые слова являются модификаторами памяти?</b></p>

1. <p><b>near</b></p>
2. <p><b>long</b></p>
3. <p><b>huge</b></p>
4. <p><b>far</b></p>
5. <p><b>short</b></p>

Ответ: 1, 3, 4

Комментарий: AP0006033

<p><b>Выберите верные утверждения об константах</b></p>

1. <p><b>Начальные значения констант изменить нельзя</b></p>
2. <p><b>Константы заданные ключевым словом define можно переопределять в любой момент </b></p>
3. <p><b>Константы заданные ключевым словом const можно переопределять в любой момент</b></p>
4. <p><b>Константы не обязательно инициализировать, можно просто объявить</b></p>
5. <p><b>#define SALES\_TEAM 10 - явл правильным примером использования define</b></p>
6. <p><b>#define SALES\_TEAM = 10 - явл правильным примером использования define</b></p>

Ответ: 1, 2, 5

Комментарий: AP0006034

<p><b>Диапазон значений типа данных char:</b></p>

1. <p><b>от -120 до 127</b></p>
2. <p><b>от 0 до 255 </b></p>
3. <p><b>от -32768 до 32767</b></p>

Ответ: 1

Комментарий: AP0006035

<p><b>Диапазон значений типа данных unsigned char:</b></p>

1. <p><b>от -120 до 127</b></p>
2. <p><b>от 0 до 255 </b></p>
3. <p><b>от -32768 до 32767</b></p>

Ответ: 1

Комментарий: AP0006036

<p><b>Какой модификатор используют для обеспечения совместимости с языком C?</b></p>

1. <p><b>huge</b></p>
2. <p><b>pascal</b></p>
3. <p><b>cdecl</b></p>

Ответ: 3

Комментарий: AP0006037

<p><b>Как использовать символ, который невозможно набрать на клавиатуре?</b></p>

1. <p><b>Использовать его ASCII код</b></p>
2. <p><b>Использовать специальный набор символов, например ‘\\’</b></p>
3. <p><b>Никак, его же невозможно набрать</b></p>

Ответ: 1,2

Комментарий: AP0006038

<p><b>Укажите ошибки в данном коде: const char = “K”</b></p>

1. <p><b>char не может быть константой</b></p>
2. <p><b>Пропущено имя переменной</b></p>
3. <p><b>Кавычки должны быть одинарными</b></p>
4. <p><b>Все варианты верные</b></p>

Ответ: 2,3

Комментарий: AP0006039

<p><b>Неявный преобразование типов:</b></p>

1. <p><b>Когда компилятор автоматически конвертирует один фундаментальный тип данных в другой.</b></p>
2. <p><b>когда разработчик использует один из операторов casts для выполнения конвертации объекта из одного типа данных в другой.
3. </b>Все варианты верные</p>

Ответ: 1

Комментарий: AP0006040

## 7 тема

**<p><b>Что выведет на печать следующий программный фрагмент в предположении, что он является частью некоторой полной программы?**

**printf("Oн набрал покупок на $%2.2f.\n", 2.345e2);** **</b></p>**

1. <p><b> Он набрал покупок на 234.50 долл.</b></p>

2 <p><b> Он набрал покупок на 23.45.</b></p>

3. <p><b> Он набрал покупок за 234.50.</b></p>

4. <p><b> Он набрал покупок за 23.450.</b></p>

5. . <p><b> Он набрал покупок за 2.34.</b></p>

Ответ: 3

Комментарий: AP00701

**<p><b>Что такое “широкий” и “узкий” набор символов?</b></p>**

1. <p><b>“Широкий” набор символов – символы в верхнем регистре, а ”узкий” – символы в нижнем регистре </b></p>

2. <p><b>“Широкий” набор символов – это ASCII, а ”узкий” – символы UNICODE. </b></p>

3. <p><b>“Узкий” набор символов – это 16-битовый набор символов, а ”широкий” – 8-битовый набор символов. </b></p>

4. <p><b> “Широкий” набор символов – это 16-битовый набор символов, а ”узкий” – 8-битовый набор символов.</b></p>

5. <p><b>“Широкий” набор символов – символы в нижнем регистре, а ”узкий” – символы в верхнем регистре </b></p>

Ответ: 4

Комментарий: AP00702

**<p><b>strrev – это…** **</b></p>**

1. <p><b>реверс строки</b></p>

2. <p><b>сравнивает строку 1 со строкой 2, не различая прописные и строчные буквы</b></p>

3. <p><b>возвращает позицию первого вхождения символа в строку</b></p>

4. <p><b>присоединяет строку2 в конец строки 1</b></p>

Ответ: 1

Комментарий: AP00703

**<p><b>Функция strtod ()**…**</b></p>**

1. <p><b> возвращает указатель на лексему, ограниченную заданным разделителем

</b></p>

2. <p><b>преобразует строку символов в число с плавающей точкой типа double </b></p>

3. <p><b>преобразует все буквы строки в прописные буквы</b></p>

4. <p><b>помещает заданный символ в заданное число позиций строки</b></p>

Ответ: 2

Комментарий: AP00704

**<p><b>Для использования функции printf() нужно подключить…</b></p>**

1. <p><b>#include <iostream></</b></p>

2. <p><b> #include <stdio.h></b></p>

3. <p><b>>#include <iomanip></b></p>

4. <p><b>>#include <stdlib.h></b></p>

Ответ: 2

Комментарий: AP00705

**<p><b>Функция нахождения символа в** строке strchr() имеет следующий прототип: **</b></p>**

1. <p><b> char strchr(const char\* string, int c)</b></p>

2. <p><b>char\* strchr(const char\* string, int c)</b></p>

3. <p><b> char\* strchr(int c, const char\* string)</b></p>

4. <p><b> char\* strchr(char\* string, int c)</b></p>

Ответ: 2

Комментарий: AP00706

**<p><b>** **Функция преобразования ultoa…</b></p>**

1. <p><b> преобразует число типа unsigned long в строку символов</b></p>

2. <p><b> преобразует строку символов в число типа unsigned long</b></p>

3. <p><b> преобразует строку символов в число типа long</b></p>

4. <p><b> преобразует число типа long в строку символов </b></p>

Ответ: 1

Комментарий: AP00707

**<p><b>** **Функция printf () имеет следующий прототип:** **</b></p>**

1. <p><b> int printf(const char \*format[, argument,...]);</b></p>

2. <p><b> int printf(const char format[, argument,...]);b></p>

3. <p><b> char printf(const char \*format[, argument,...]);</b></p>

4. <p><b> char printf(char \*format[, argument,...]);</b></p>

Ответ: 1

Комментарий: AP00708

**<p><b> Что выведет на печать следующий программный фрагмент в предположении, что он является частью некоторой полной программы?**

**printf("/%2d/\n", 336);</b></p>**

1. <p><b>/336/ </b></p>

2. <p><b>/ 336/ </b></p>

3. <p><b>/336 /</b></p>

Ответ: 1

Комментарий: AP00709

**<p><b>Прототип функции scanf(): </b></p>**

1. <p><b>char scanf(const char \*format[, address,...]);</b></p>

2. <p><b>Int scanf(char format[, address,...]);</b></p>

3. <p><b>Int scanf(const char \*format[, address,...]);</b></p>

4. <p><b>char scanf(const char \*format[, address,...]);</b></p>

Ответ: 3

Комментарий: AP00710

**<p><b>Функция strlwr()… </b></p>**

1. <p><b>преобразует строку символов к нижнему регистру</b></p>

2. <p><b>проводит сравнение символов одной строки с символами другой </b></p>

3. <p><b> меняет порядок следования символов на обратный </b></p>

4. <p><b> осуществляет поиск заданного символа с конца строки</b></p>

Ответ: 1

Комментарий: AP00711

**<p><b>Прототип функции strncat(): </b></p>**

1. <p><b> char\* strncat(const char\* str1, const char\* str2, size\_t num)</b></p>

2. <p><b> char strncat(char\* str1, char\* str2, size\_t num)</b></p>

3. <p><b> char\* strncat(char str1, char\* str2, size\_t num)</b></p>

4. <p><b> char\* strncat(char\* str1, const char\* str2, size\_t num)</b></p>

Ответ: 4

Комментарий: AP00712

**<p><b>Какие два параметра принимает функция getline()?</b></p>**

1. <p><b>строка, в которую осуществляется ввод</b></p>

2. <p><b>число символов, подлежащих вводу</b></p>

3. <p><b>строка, в которую осуществляется вывод</b></p>

Ответ: 1,2

Комментарий: AP00713

**<p><b>”Символьная строка”– это…</b></p>**

1. <p><b>это последовательность, состоящая из одного и более символов </b></p>

2. <p><b>это последовательность, состоящая только из прописных и строчных букв</b></p>

3. <p><b>это последовательность, состоящая только из цифр и пробелов </b></p>

Ответ: 1

Комментарий: AP00714

**<p><b>Что является завершение массива символов?</b></p>**

1. <p><b>нуль(0) </b></p>

2. <p><b>последний символ </b></p>

3. <p><b>символ нулевым значением(‘\0’)</b></p>

Ответ: 3

Комментарий: AP00715

**<p><b>Чем является [40] в данной вырезке из кода:**

**сhar name[40];</b></p>**

1. <p><b>является название массива</b></p>

2. <p><b>размер элементов массива name</b></p>

3. <p><b>число элементов массива name</b></p>

Ответ: 3

Комментарий: AP00716

**<p><b>Когда функция scanf() прекратит ввод символов?</b></p>**

1. <p><b>при встрече с разделителем(пробел, символ табуляции или перевода строки)</b></p>

2. <p><b>когда встретит заглавную букву</b></p>

3. <p><b>когда встретит цифру</b></p>

Ответ: 1

Комментарий: AP00717

**<p><b>Что выполняет функция strlen()?</b></p>**

1. <p><b>реверс строки</b></p>

2. <p><b>возвращает длину строки в байтах, не учитывая нулевой терминатор </b></p>

3. <p><b>присоединяет одну строку в конец другой </b></p>

4. <p><b>возвращает позицию первого вхождения символа из заданного набора символов </b></p>

Ответ: 2

Комментарий: AP00718

**<p><b> Что выполняет функция streat()?</b></p>**

1. <p><b>возвращает позицию первого вхождения символа из заданного набора символов</b></p>

2. <p><b>возвращает длину строки в байтах, не учитывая нулевой терминатор </b></p>

3. <p><b>процедуры, выполняющие некоторые действия</b></p>

4. <p><b>присоединяет одну строку в конец другой </b></p>

Ответ: 4

Комментарий: AP00719

**<p><b> Что выполняет функция atoi()?</b></p>**

1. <p><b> преобразует строку символов в строку типа int </b></p>

2. <p><b> преобразует строку символов в число типа long </b></p>

3. <p><b>преобразует строку символов в число с плавающей точкой</b></p>

Ответ: 1

Комментарий: AP00720

**<p><b>Тип выводимой информации формата %с? </b></p>**

1. <p><b> Один символ </b></p>

2. <p><b> Число с плавающей точкой, десятичная запись </b></p>

3. <p><b> Десятичное целое число без знака </b></p>

Ответ: 1

Комментарий: AP00721

**<p><b>Что позволяет осуществлять функция printf()</b></p>**

1. <p><b>Она позволяет осуществлять вывод в стандартный поток вывода stdout</b></p>

2. <p><b>Она позволяет осуществлять форматированный вывод в стандартный поток вывода stdout</b></p>

Ответ: 2

Комментарий: AP00722

**<p><b>Что позволяет осуществлять функция scanf()</b></p>**

1. <p><b>Она позволяет осуществлять ввод в стандартный поток вывода stdout</b></p>

2. <p><b>Она позволяет осуществлять форматированный ввод информации из стандартного потока ввода</b></p>

Ответ: 2

Комментарий: AP00723

**<p><b> Что выполняет функция ltoa()?</b></p>**

1. <p><b> преобразует число типа int в строку символов </b></p>

2. <p><b>преобразует строку символов в число типа unsigned long

</b></p>

3. <p><b> преобразует строку символов в число типа long </b></p>

4. <p><b> преобразует число типа long в строку символов </b></p>

Ответ: 4

Комментарий: AP00724

<p><b>**Как называется 8-битовый набор символов?**</b></p>

1. <p><b>”широким”</b></p>

2. <p><b>”узким”</b></p>

3. <p><b>”длинным”</b></p>

Ответ: 2

Комментарий: AP00725

<p><b>**Как называется 16-битовый набор символов?**</b></p>

1. <p><b>”порядочным”</b></p>

2. <p><b>”числовым”</b></p>

3. <p><b>”широким”</b></p>

Ответ: 3

Комментарий: AP00726

<p><b>**Что называют ASCIIZ-строками?**</b></p>

1. <p><b> Строки – которые заканчивающиеся терминатором строки - символом с нулевым значением ('\0')</b></p>

2. <p><b>Строки – которые начинающиеся терминатором строки - символом с нулевым значением ('\0')</b></p>

3. <p><b> Строки – которые и начинающиеся и заканчивающиеся терминатором строки - символом с нулевым значением ('\0')</b></p>

Ответ: 1

Комментарий: AP00727

<p><b>**Для чего служит «нуль-символ»?**</b></p>

1. <p><b>Для того, чтобы начать отсчет.</b></p>

2. <p><b>Для того, чтобы отмечать начало строки.</b></p>

3. <p><b>Для того, чтобы отмечать конец строки.</b></p>

Ответ: 3

Комментарий: AP00728

<p><b>**Что такое «массив»**?</b></p>

1. <p><b>Массив — это упорядоченная последовательность элементов разного типа</b></p>

2. <p><b>Массив — это упорядоченная последовательность элементов данных одного типа.</b></p>

Ответ: 2

Комментарий: AP00729

<p><b>**Что выполняет функция ltoi()**?</b></p>

1. <p><b>Преобразует строку символов в число типа unsigned long.</b></p>

2. <p><b>Преобразует число типа long в строку символов.</b></p>

3. <p><b>Преобразует число типа unsigned long в строку символов.</b></p>

Ответ: 2

Комментарий: AP00730

<p><b>**Что выведет на печать следующий программный фрагмент в предположения?**

printf(" %x\n", 336);</b></p>

1. <p><b>336</b></p>

2. <p><b>65200</b></p>

3. <p><b>520</b></p>

4. <p><b>150</b></p>

5. <p><b>-336</b></p>

Ответ: 4

Комментарий: AP00731

<p><b>**Прототип функции printf ()** ?</b></p>

1. <p><b> >Int printf(char format[, address,...]) </b></p>

2. <p><b>Int printf(const char \*format[, argument,...]);</b></p>

3. <p><b>Char printf(const char \*format[, address,...] </b></p>

Ответ: 2

Комментарий: AP00732

<p><b> **Символы-заполнители - это ?** </b></p>

1. <p><b>Это ASCII-символы, за исключением знака процента. </b></p>

2. <p><b>Это символы которые заполняют. </b></p>

3. <p><b>Это пробел, символ табуляции и перевод строки. </b></p>

Ответ: 3

Комментарий: AP00733

<p><b> **Прототип функции sscanf()** ?</b></p>

1. <p><b>Char sscanf(const char \*buffer, const char \*format[, address,...]);</b></p>

2. <p><b>Int sscanf(const char \*buffer, const char \*format[, address,...]);</b></p>

3. <p><b>Int sscanf(const char \*buffer, const char \*format[, argument,...]);</b></p>

4. <p><b>Char sscanf(const char \*buffer, const char \*format[, argument,...]);</b></p>

Ответ: 2

Комментарий: AP00734

## 8 тема

<p><b>Операция преобразования переменной int F к типу float\* верно записана в строке:</p></b>

1. <p><b>(float)\*F</b></p>
2. <p><b>(&float)F</b></p>
3. <p><b>(float\*)F</b></p>
4. <p><b>\*(float)F</b></p>

Ответ: 3

Комментарий: AP008001

<p><b>Операция преобразования переменной int F к типу float\* верно записана в строке:</p></b>

1. <p><b>(float)\*F</b></p>
2. <p><b>(&float)F</b></p>
3. <p><b>(float\*)F</b></p>
4. <p><b>\*(float)F</b></p>

Ответ: 3

Комментарий: AP008002

<p><b>Укажите операции, приоритет которых ниже, чем у операции постфиксального инкремента: </p></b>

1. <p><b>динамическое выделение памяти</b></p>
2. <p><b>логическое отрицание</b></p>
3. <p><b>вызов функции</b></p>
4. <p><b>обращение по адресу</b></p>

Ответ: 1, 2, 4

Комментарий: AP008003

<p><b>Выберите строки, являющиеся выражениями:</p></b>

1. <p><b>4;</b></p>
2. <p><b>q += 2;</b></p>
3. <p><b>a > 3;</b></p>
4. <p><b>5 + 7;</b></p>

Ответ: 1, 2, 3, 4

Комментарий: AP008004

<p><b>Наиболее низким по приоритету среди нижеперечисленных операторов является: </p></b>

1. <p><b>int</b></p>
2. <p><b>char</b></p>
3. <p><b>short</b></p>
4. <p><b>unsigned int</b></p>

Ответ: 2

Комментарий: AP008005

<p><b>Результатом выполнения выражения int mice = (int)2.7 + (float)3.3 будет являться:</p></b>

1. <p><b>сообщение об ошибке</b></p>
2. <p><b>присвоение переменной mice значения 0</b></p>
3. <p><b>присвоение переменной mice значения 5</b></p>
4. <p><b>присвоение переменной mice значения 6</b></p>

Ответ: 3

Комментарий: AP008006

<p><b>Как происходит явное преобразование?</b></p>

1. <p><b> происходит автоматически, по правилам, заложенным в языке Си</b></p>
2. <p><b> происходит только при указании необходимого типа перед выражением</b></p>
3. <p><b> в зависимости от типа данных, к которому происходит преобразование</b></p>

Ответ: 2

Комментарий: AP0008007

<p><b> Можно ли производить операцию деления по модулю над числами с плавающей запятой? </b></p>

1. <p><b>Да </b></p>

2. <p><b>Нет</b></p>

Ответ: 2

Комментарий: AP0008008

<p><b> Означает ли знак равенства в СИ «равно»? </b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 2

Комментарий: AP0008009

<p><b> Значением выражения 13 % 5 будет </b></p>

1. <p><b> 1 </b></p>

2. <p><b> 5 </b></p>

3. <p><b> 3 </b></p>

Ответ: 3

Комментарий: AP00080010

<p><b> Значением выражения 13 / 5 будет </b></p>

1. <p><b> 2 </b></p>

2. <p><b> 3 </b></p>

3. <p><b> 2.6 </b></p>

Ответ: 1

Комментарий: AP0008011

<p><b> Значением выражения 3 << 1 будет </b></p>

1. <p><b> 4 </b></p>

2. <p><b> 6 </b></p>

3. <p><b> 1 </b></p>

Ответ: 2

Комментарий: AP0008012

<p><b> Значением выражения (5 > 2) ? 7:3 будет </b></p>

1. <p><b> 1 </b></p>

2. <p><b> 0 </b></p>

3. <p><b> 7 </b></p>

4. <p><b> 3 </b></p>

Ответ: 3

Комментарий: AP0008013

<p><b> Операнд - это </b></p>

1. <p><b> тип данных </b></p>

2. <p><b> совокупность операций </b></p>

3. <p><b> то, над чем выполняются операции </b></p>

3. <p><b> то, что выполняет операцию </b></p>

Ответ: 3

Комментарий: AP0008014

<p><b> Выберите операцию с наивысшим приоритетом </b></p>

1. <p><b> % </b></p>

2. <p><b> [] </b></p>

3. <p><b> , </b></p>

4. <p><b> = </b></p>

Ответ: 2

Комментарий: AP0008015

<p><b> Операция запятая «,» имеет самый низкий приоритет</b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 1

Комментарий: AP0008016

<p><b> Может ли «понижение» типа привести к некорректности данных?</b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 1

Комментарий: AP0008017

<p><b> Значение константы можно изменять </b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 2

Комментарий: AP0008018

<p><b> Операнды могут быть как переменными, так и константными </b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 1

Комментарий: AP0008019

<p><b> Выберите унарные операторы</b></p>

1. <p><b> ! </b></p>

2. <p><b> % </b></p>

3. <p><b> ~ </b></p>

4. <p><b> sizeof </b></p>

Ответ: 1, 3, 4

Комментарий: AP0008020

<p><b> Выберите логические операторы</b></p>

1. <p><b> || </b></p>

2. <p><b> = </b></p>

3. <p><b> == </b></p>

4. <p><b> && </b></p>

Ответ: 1, 4

Комментарий: AP0008021

<p><b> Сколько(примерно) операций имеется в языке СИ</b></p>

1. <p><b> около 20 </b></p>

2. <p><b> около 40 </b></p>

3. <p><b> больше 100 </b></p>

4. <p><b> меньше 10 </b></p>

Ответ: 2

Комментарий: AP0008022

<p><b> Выберите операцию приведения типа в языке Си</b></p>

1. <p><b> typeof</b></p>

2. <p><b> new </b></p>

3. <p><b> casting </b></p>

4. <p><b> type</b></p>

Ответ: 3

Комментарий: AP0008023

<p><b> Генерацию исключения в языке Си производит оператор: </b></p>

1. <p><b> throw </b></p>

2. <p><b> new </b></p>

3. <p><b> elimination </b></p>

4. <p><b> exception </b></p>

Ответ: 1

Комментарий: AP0008024

<p><b> Выберите операторы, имеющие постфиксную и префиксную формы </b></p>

1. <p><b> + </b></p>

2. <p><b> -- </b></p>

3. <p><b> ++ </b></p>

4. <p><b> ~ </b></p>

Ответ: 2, 3

Комментарий: AP0008025

<p><b> Операция инкремент «++» служит для </b></p>

1. <p><b> Выделения памяти </b></p>

2. <p><b> Сдвига вправо на 1 бит </b></p>

3. <p><b> Увеличения значения переменной на единицу </b></p>

4. <p><b> Сложения двух операндов </b></p>

Ответ: 3

Комментарий: AP0008026

<p><b> Оператор new предназначен для.. </b></p>

1. <p><b> Создания нового объекта </b></p>

2. <p><b> Логического отрицания </b></p>

3. <p><b> Динамического выделения памяти </b></p>

4. <p><b> Статического выделения памяти</b></p>

Ответ: 3

Комментарий: AP0008027

<p><b> Выражение - это </b></p>

1. <p><b> Директива препроцессора </b></p>

2. <p><b> Объединение операций и операндов </b></p>

3. <p><b> Символ “;” </b></p>

4. <p><b> Функция </b></p>

Ответ: 2

Комментарий: AP0008028

<p><b> Примерами выражений являются </b></p>

1. <p><b> 24; </b></p>

2. <p><b> a+b; </b></p>

3. <p><b> #include <iostream> </b></p>

1. <p><b> x = 1 + y; </b></p>

Ответ: 1, 2, 4

Комментарий: AP0008030

<p><b> Дополнительными операциями являются: </b></p>

1. <p><b> % </b></p>

2. <p><b> (тип) – определение типа </b></p>

3. <p><b> sizeof </b></p>

1. <p><b> ++ </b></p>

Ответ: 2, 3

Комментарий: AP0008031

<p><b> Как распознать оператор? </b></p>

1. <p><b> По верхнему регистру </b></p>

2. <p><b> По весу </b></p>

3. <p><b> По символу </b></p>

4. <p><b> По нижнему регистру </b></p>

Ответ: 3

Комментарий: AP0008032

<p><b> Как вычисляются выражения со смешенными типами данных? </b></p>

1. <p><b> С помощью автоматического преобразования типов</b></p>

2. <p><b> С помощью оператора присваивания «=» </b></p>

3. <p><b> Не вычисляются (выдаётся ошибка)</b></p>

4. <p><b> С помощью ключевого слова type </b></p>

Ответ: 1

Комментарий: AP0008033

<p><b> Можно ли использовать цикл без фигурных скобок «{}» в языке Си? </b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 1

Комментарий: AP0008034

<p><b> Выберите верное утверждение </b></p>

1. <p><b> Значение операции запятая – значение выражения, стоящего справа </b></p>

2. <p><b> Значение операции запятая – значение выражения, стоящего слева </b></p>

Ответ: 1

Комментарий: AP0008035

<p><b> Если результат сравнения истина, то результатом будет</b></p>

1. <p><b> Левый операнд</b></p>

2. <p><b> true</b></p>

3. <p><b> false</b></p>

4. <p><b> Правый операнд</b></p>

Ответ: 2

Комментарий: AP0008036

<p><b> Каждый бит может принимать значение </b></p>

1. <p><b> 0 </b></p>

2. <p><b> 1 </b></p>

3. <p><b> 255 </b></p>

4. <p><b> любое</b></p>

Ответ: 1, 2

Комментарий: AP0008037

<p><b> Каждый бит может принимать значение </b></p>

1. <p><b> 0 </b></p>

2. <p><b> 1 </b></p>

3. <p><b> 255 </b></p>

4. <p><b> любое</b></p>

Ответ: 1, 2

Комментарий: AP0008038

<p><b> Биты в байте считаются.. </b></p>

1. <p><b> Слева направо </b></p>

2. <p><b> Справа налево </b></p>

3. <p><b> Без разницы </b></p>

4. <p><b> Сверху вниз</b></p>

Ответ: 2

Комментарий: AP0008039

<p><b> В операции поразрядного И сравниваются два разряда и в результате получается 1, если оба разряда или один из них равны 1 </b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 2

Комментарий: AP0008040

<p><b> В Си есть функция, осуществляющая возведение в квадрат</b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 2

Комментарий: AP0008041

<p><b> Синтаксис использования функции printf()</b></p>

1. <p><b> printf(text)</b></p>

2. <p><b> printf(“text”) </b></p>

3. <p><b> printf(‘text’) </b></p>

4. <p><b> printf(/text/) </b></p>

Ответ: 2

Комментарий: AP0008042

<p><b> Сколько циклов можно вложить в цикл for? </b></p>

1. <p><b> Не больше 10 </b></p>

2. <p><b> 1 </b></p>

3. <p><b> 0 </b></p>

4. <p><b> Сколько угодно </b></p>

Ответ: 4

Комментарий: AP0008043

**<p><b>Что делает оператор sizeof?</b></p>**

1. <p><b>возвращает длину строки</b></p>

2. <p><b>возвращает размер в байтах объекта или типа данных</b></p>

3. <p><b>возвращает размер в битах объекта или типа данных</b></p>

4. <p><b>возвращает размер массива в битах</b></p>

Ответ: 2

Комментарий: AP0008044

**<p><b>Выберите верные утверждения</b></p>**

1. <p><b>операнды могут быть константами, переменными или их сочетаниями </b></p>

2. <p><b>в языке Си указанием на наличие оператора служит символ «точка с запятой», стоящий в конце него</b></p>

3. <p><b>операции в Си не имеют порядка выполнения</b></p>

4. <p><b>выражение представляет собой объединение операций и операндов</b></p>

Ответ: 1, 2, 4

Комментарий: AP0008045

**<p><b>Какой из перечисленных операторов имеет наивысший приоритет </b></p>**

1. <p><b>префиксальный инкремент</b></p>

2. <p><b>постфиксальный инкремент</b></p>

3. <p><b>оператор разыменования</b></p>

4. <p><b>оператор побитового сдвига</b></p>

Ответ: 2

Комментарий: AP0008046

**<p><b>Что является результатом операции 12 & 5?<b></p>**

1. <p><b>2</b></p>

2. <p><b>17</b></p>

3. <p><b>7</b></p>

4. <p><b>4</b></p>

Ответ: 4

Комментарий: AP0008047

**<p><b>Какая операция имеет наивысший приоритет?</b></p>**

1. **<p><b>**\***</b></p>**

2. **<p><b>**||**</b></p>**

3. **<p><b>**+**</b></p>**

4. **<p><b>**=**</b></p>**

5. **<p><b>**<=**</b></p>**

Ответ: 1

Комментарий: AP0008048

**<p><b>Какой операции не существует в языке С++?</b></p>**

1. **<p><b>**&&**</b></p>**

2. **<p><b>**||**</b></p>**

3. **<p><b>**,**</b></p>**

4. **<p><b>**!!**</b></p>**

5. **<p><b>**/=**</b></p>**

5. **<p><b>**?:**</b></p>**

Ответ: 4

Комментарий: AP0008049

**<p><b>Для чего используется спецификатор** %d**</b></p>**

1. **<p><b>**десятичное число без знака**</b></p>**

2. **<p><b>**десятичное число со знаком**</b></p>**

3. **<p><b>**одиночный символ **</b></p>**

4. **<p><b>**десятичноечисло с плавающей точкой **</b></p>**

Ответ: 2

Комментарий: AP0008050

**<p><b>Кто(что) выполняет неявное преобразование типов данных?</b></p>**

1. **<p><b>**программист**</b></p>**

2. **<p><b>**компилятор**</b></p>**

3. **<p><b>**отладчик**</b></p>**

4. **<p><b>**все варианты неверные**</b></p>**

Ответ: 2

Комментарий: AP0008051

<p><b>Что гласит закон Де Моргана? </b></p>

1. <p><b>!(x && y) эквивалентно !x || !y </b></p>

2. <p><b> !(x || y) эквивалентно !x && !y</b></p>

3. <p><b>!(x && y) эквивалентно !x && !y </b></p>

4. <p><b>!(x || y) эквивалентно !x || !y </b></p>

Ответ: 1, 2

Комментарий: AP0008052

**<p><b>Оператор обработки исключений</b></p>**

1. <p><b>**try-catch**</b></p>

2. <p><b>**switch-case**</b></p>

3. <p><b>if-else</b></p>

Ответ: 1

Комментарий: AP0008053

**<p><b>Оператор-переключатель </b></p>**

1. <p><b>**try-catch**</b></p>

2. <p><b>**switch-case**</b></p>

3. <p><b>if-else</b></p>

Ответ: 2

Комментарий: AP0008054

**<p><b>Оператор-переключатель </b></p>**

1. <p><b>**try-catch**</b></p>

2. <p><b>**switch-case**</b></p>

3. <p><b>if-else</b></p>

Ответ: 3

Комментарий: AP0008055

**<p><b>Оператор продолжения</b></p>**

1. <p><b>**break**</b></p>

2. <p><b>while</b></p>

3. <p><b>continue</b></p>

Ответ: 3

Комментарий: AP0008056

**<p><b>Оператор разрыва прерывает выполнение операторов</b></p>**

1. <p><b>while, do, for и switch</b></p>

2. <p><b>while, do, for</b></p>

3. <p><b>try, switch, for</b></p>

4. <p><b>try и switch</b></p>

Ответ: 1

Комментарий: AP0008057

**<p><b>Константа соответствующего типа, задающая значение, которое присваивается переменной при создании это</b></p>**

1. <p><b>Макроопределение</b></p>

2. <p><b>Оператор</b></p>

3. <p><b>Инициализатор</b></p>

Ответ: 3

Комментарий: AP0008058

**<p><b>Программа, которая обрабатывает текст вашей программы до компилятора</b></p>**

1. <p><b>Макрос</b></p>

2. <p><b>Препроцессор </b></p>

3. <p><b> Идентификатор</b></p>

Ответ: 3

Комментарий: AP0008059

**<p><b>Выражение - это</b></p>**

1. <p><b>константа соответствующего типа, задающая значение, которое присваивается переменной при создании</b></p>

2. <p><b>любая комбинация операндов или элементов данных (чисел, констант, переменных, функций и других операндов) и знаков операций</b></p>

3. <p><b>символы, которые используются для соединения операндов в выражении</b></p>

Ответ: 2

Комментарий: AP0008060

**<p><b>Знаки операций - это</b></p>**

1. <p><b>символы, которые используются для соединения операндов в выражении</b></p>

2. <p><b>наименьшая исполняемая единица программного кода VBA</b></p>

3. <p><b>константа соответствующего типа, задающая значение, которое присваивается переменной при создании</b></p>

Ответ: 1

Комментарий: AP0008061

**<p><b>Оператор - это</b></p>**

1. <p><b>символ, который используются для соединения операндов в выражении</b></p>

2. <p><b>наименьшая исполняемая единица программного кода VBA</b></p>

3. <p><b>константа соответствующего типа, задающая значение, которое присваивается переменной при создании</b></p>

Ответ: 2

Комментарий: AP0008062

**<p><b>Оператор присваивания - это</b></p>**

1. <p><b>символы, которые используются для соединения операндов в выражении</b></p>

2. <p><b>связанное знаком равенства выражение, в котором значение выражения, указанное справа, присваивается переменной, находящейся слева от знака равенства</b></p>

3. <p><b>константа соответствующего типа, задающая значение, которое присваивается переменной при создании</b></p>

Ответ: 2

Комментарий: AP0008063

**<p><b> Какому оператору соответствует таблица истинности?</b></p>**

|  |  |  |
| --- | --- | --- |
| **Левый операнд** | **Правый операнд** | **Результат** |
| **false** | **false** | **false** |
| **false** | **true** | **true** |
| **true** | **false** | **true** |
| **true** | **true** | **true** |

1. <p><b> НЕ (!)</b></p>

2. <p><b> И (&&)</b></p>

3. <p><b> ИЛИ ||</b></p>

Ответ: 3

Комментарий: AP0008064

**<p><b> Какому оператору соответствует таблица истинности?</b></p>**

|  |  |
| --- | --- |
| **Операнд** | **Результат** |
| **true** | **false** |
| **false** | **true** |

1. <p><b> НЕ (!)</b></p>

2. <p><b> И (&&)</b></p>

3. <p><b> ИЛИ ||</b></p>

Ответ: 1

Комментарий: AP0008065

**<p><b> Какому оператору соответствует таблица истинности?</b></p>**

|  |  |  |
| --- | --- | --- |
| **Левый операнд** | **Правый операнд** | **Результат** |
| **false** | **false** | **false** |
| **false** | **true** | **false** |
| **true** | **false** | **false** |
| **true** | **true** | **true** |

1. <p><b> НЕ (!)</b></p>

2. <p><b> И (&&)</b></p>

3. <p><b> ИЛИ ||</b></p>

Ответ: 2

Комментарий: AP0008066

## 9 тема

**<p><b> Какой символ означает: не равно? </b></p>**

1. <p><b>/=</b></p>

2. <p><b> !=</b></p>

3.<p><b> =! </b></p>

Ответ: 2

Комментарий: AP00901

**<p><b>Какие параметры включает в себя функция char \*\_itoa\_s();?</b></p>**

1. <p><b>преобразуемое число</b></p>

2. <p><b>строка в которую преобразуется число</b></p>

3. <p><b> основание степени счисления </b></p>

4. <p><b>длина преобразуемого числа</b></p>

Ответ: 1, 2, 3

Комментарий: AP000902

**<p><b>Какая библиотека используется для работы с файлами в C++?</b></p>**

1. <p><b>files.h</b></p>

2. <p><b>fstream</b></p>

3. <p><b>string</b></p>

4. <p><b>conio.h </b></p>

Ответ: 2

Комментарий: AP000903

**<p><b>Что выполняет функция .is\_open() при работе с файлами?</b></p>**

1. <p><b>возвращает значение true</b></p>

2. <p><b>открывает файл</b></p>

3. <p><b>делает файл доступным для чтения</b></p>

4. <p><b> возвращает значение false</b></p>

Ответ: 1, 4

Комментарий: AP000904

**<p><b>Библиотека <iostream> определяет такие стандартные потоки как: </b></p>**

1. <p><b> cin стандартный входной поток</b></p>

2. <p><b> cout стандартный выходной поток </b></p>

3. <p><b> getline() стандартный поток ввода текста </b></p>

4. <p><b> cerr стандартный поток вывода сообщения об ошибке</b></p>

Ответ: 1, 2, 4

Комментарий: AP000905

**<p><b>Какие метода работы с портами ввода-вывода применяются в языке С++?</b></p>**

1. <p><b> Определение портов и линий ввода-вывода с помощью стандартных библиотек </b></p>

2. <p><b> Определение портов и линий ввода-вывода с помощью препроцессора </b></p>

3. <p><b> Передача порта в код, который его использует, посредствам указателя </b></p>

4. <p><b> Виртуальные порты </b></p>

Ответ: 2, 3, 4

Комментарий: AP000906

**<p><b>Какие потоки используются для записи в файл?</b></p>**

1. <p><b>ofstream</b></p>

2. <p><b>ifstream</b></p>

3. <p><b>fstream</b></p>

4. <p><b>cin</b></p>

Ответ: 1, 3

Комментарий: AP000907

**<p><b> Какие параметры включает в себя функция fseek() </b></p>**

1. <p><b> указатель на объект типа file</b></p>

2. <p><b> количество байт для смещения </b></p>

3. <p><b> позиция указателя относительно которой будет выполняться смещение</b></p>

4. <p><b>номер позиции указателя по счёту</b></p>

Ответ: 1, 2, 3

Комментарий: AP00908

**<p><b>Как связать поток класс ofstream с файлом file.txt</b></p>**

1. <p><b> fout.open(“file.txt”)</b></p>

2. <p><b> ofstream fout(“file.txt”)</b></p>

3. <p><b> open\_file(“file.txt”)</b></p>

4. <p><b> fout(“file.txt”)</b></p>

Ответ: 2

Комментарий: AP0009009

**<p><b> Как установить режим для записи в конец файла</b></p>**

1. <p><b> ios\_base::in </b></p>

2. <p><b> ios\_base::out </b></p>

3. <p><b> ios\_base::ate</b></p>

4. <p><b> ios\_base::app </b></p>

Ответ: 3

Комментарий: AP000910

**<p><b>Какой класс используется для установки режима открытия файла при работе с файлами потока fstream </b></p>**

1. <p><b> fstream()</b></p>

2. <p><b> is\_open()</b></p>

3. <p><b> ios\_base </b></p>

4. <p><b> ofstream fout()</b></p>

Ответ: 3

Комментарий: AP000911

**<p><b> Как установить режим открытия файла при котором удаляется содержимое файла если он существует</b></p>**

1. <p><b> ios\_base::app </b></p>

2. <p><b> ios\_base::trunc </b></p>

3. <p><b> ios\_base::ate</b></p>

4. <p><b> ios\_base::del </b></p>

Ответ: 2

Комментарий: AP000912

**<p><b>****Функция getchar() получает </b></p>**

1. <p><b>строку</b></p>

2. <p><b>один символ</b></p>

3. <p><b>слово</b></p>

Ответ: 2

Комментарий: AP00913

**<p><b>Аргументом функции putchar() может быть </b></p>**

1. <p><b>строка</b></p>

2. <p><b>слово</b></p>

3. <p><b>одиночный символ (включая знаки, представляемые управляющими последовательностями), переменная или функция, значением которой является одиночный символ</b></p>

Ответ: 3

Комментарий: AP00914

**<p><b>«Небуферизованный» ввод, означает, что </b></p>**

1. <p><b>вводимый символ оказывается временно недоступным ожидающей программе</b></p>

2. <p><b>вводимый символ сравнивается с заданным условием</b></p>

3. <p><b>вводимый символ оказывается немедленно доступным ожидающей программе</b></p>

Ответ: 3

Комментарий: AP00915

**<p><b>«Буферизованный» ввод, означает, что </b></p>**

1. <p><b>вводимые символы собираются и помещаются в некоторую область временной памяти, называемую «буфером»</b></p>

2. <p><b>вводимый символ не сравнивается с заданным условием</b></p>

3. <p><b>вводимый символ оказывается немедленно доступным ожидающей программе</b></p>

Ответ: 1

Комментарий: AP00916

**<p><b>Выберите вариант, в котором описывается преимущество буфера.</b></p>**

1. <p><b>Передачу нескольких символов в виде одного блока можно осуществить гораздо быстрее, чем передавать их последовательно по одному</b></p>

2. <p><b>Если при вводе символов допущена ошибка, вы можете воспользоваться корректирующими средствами терминала, чтобы ее исправить</b></p>

3. <p><b>Оба варианта являются верными </b></p>

Ответ: 3

Комментарий: AP00917

**<p><b>Что произойдет в результате выполнения следующего оператора putchar (' \b')?</b></p>**

1. <p><b>Переход назад на одну позицию</b></p>

2. <p><b>Переход в начало строки</b></p>

3. <p><b>Переход в начало страницы</b></p>

Ответ: 1

Комментарий: AP00918

**<p><b>Предположим, вы осуществили компиляцию программы ввод-вывод15 (UNIX) и поместили выполняемый объектный код в файл с именем getput15. Затем, чтобы запустить данную программу, вы вводите с терминала</b></p>**

1. <p><b>./getput15</b></p>

2. <p><b>getput15</b></p>

3. <p><b>/getput15</b></p>

4. <p><b>Все варианты являются верными</b></p>

5. <p><b>Ни один из вариантов не является верным</b></p>

Ответ: 2

Комментарий: AP00919

**<p><b>В ОС UNIX символ < служит обозначением операции </b></p>**

1. <p><b>сравнения</b></p>

2. <p><b>удаления</b></p>

3. <p><b>переключения</b></p>

Ответ: 3

Комментарий: AP00920

**<p><b>В ОС UNIX символ > служит обозначением операции </b></p>**

1. <p><b>сравнения</b></p>

2. <p><b>удаления</b></p>

3. <p><b>переключения</b></p>

Ответ: 3

Комментарий: AP00921

**<p><b>В ОС UNIX символ >> служит обозначением операции </b></p>**

1. <p><b>сравнения</b></p>

2. <p><b>позволяющей добавлять данные в конец существующего файла</b></p>

3. <p><b>переключения</b></p>

Ответ: 2

Комментарий: AP00922

**<p><b>В ОС UNIX символ | служит обозначением операции </b></p>**

1. <p><b>сравнения</b></p>

2. <p><b>позволяющей добавлять данные в конец существующего файла</b></p>

3. <p><b>конвейерной пересылки</b></p>

Ответ: 3

Комментарий: AP00923

**<p><b>Что объединяет данные символы (|, >, <, >>)?</b></p>**

1. <p><b>Оба утверждения являются верными</b></p>

2. <p><b>Данные символы используются в С++ </b></p>

3. <p><b>Они являются операциями в ОС UNIX</b></p>

4. <p><b>Оба утверждения не являются верными</b></p>

Ответ: 1

Комментарий: AP00924

**<p><b>** **Под функциями printf() и scanf() подразумеваются функции</b></p>**

1. <p><b> которые выполняют транспортировку данных в программу и из нее </b></p>

2. <p><b>которые выполняют транспортировку данных только в программу</b></p>

3. <p><b>которые выполняют транспортировку данных только из программы</b></p>

Ответ: 1

Комментарий: AP00925

**<p><b>Аргумент одиночный символ (включая знаки, представляемые управляющими последовательностями), переменная или функция, значением которой является одиночный символ принадлежит функции: </b></p>**

1. <p><b> getchar() </b></p>

2. <p><b> putchar()</b></p>

3. <p><b>Оба варианты верны</b></p>

Ответ: 2

Комментарий: AP00926

**<p><b> Как переключить стандартный ввод на вывод из файла ?</b></p>**

1. <p><b> program < file </b></p>

2. <p><b> program > file </b></p>

3. <p><b> program >> file</b></p>

Ответ: 2

Комментарий: AP00927

**<p><b> Как переключить стандартный вывод на ввод из файла ?</b></p>**

1. <p><b> program < file </b></p>

2. <p><b> program > file </b></p>

3. <p><b> program >> file</b></p>

Ответ: 1

Комментарий: AP00928

**<p><b> Например, в программах обработки текстов было бы желательно, чтобы каждая команда вводилась, как только вы нажимаете соответствующую клавишу. Какой ввод может оказаться приемлемым? </b></p>**

1. <p><b> Как буферизованный, так и небуферизованный ввод имеет свои достоинства </b></p>

2. <p><b> Только небуферизованный </b></p>

3. <p><b> Только буферизованный </b></p>

Ответ: 1

Комментарий: AP00929

**<p><b> Зачем нужны буферы ?</b></p>**

1. <p><b>Передачу нескольких символов в виде одного блока можно осуществить гораздо быстрее, чем передавать их последовательно по одному </b></p>

2. <p><b> Если при вводе символов допущена ошибка, то можно воспользоваться корректирующими средствами терминала, чтобы ее исправить. И когда в конце концов вы нажмете клавишу [ввод], будет произведена передача откорректированной строки </b></p>

3. <p><b>Оба утверждения верны.</b></p>

Ответ : 3

Комментарий: AP00930

**<p><b> Каким должен быть идеальный признак STOP ?</b></p>**

1. <p><b> Это должен быть такой символ, который обычно не используется в тексте и, следовательно, не приводит к ситуации, когда он случайно встретится при вводе, и работа программы будет остановлена раньше, чем мы хотели бы</b></p>

2. <p><b> Это должен быть такой символ, который обычно используется в тексте и, следовательно, приводит к ситуации, когда он случайно встретится при вводе, и работа программы будет остановлена раньше, чем мы хотели бы</b></p>

3. <p><b>Оба утверждения неверны</b></p>

Ответ: 1

Комментарий: AP00931

**<p><b> Что такое файл?</b></p>**

1. <p><b> Файлом можно назвать определенный участок памяти, в который помещена важная информация.</b></p>

2. <p><b> Файлом можно назвать участок памяти, в который помещена некоторая информация.</b></p>

3. <p><b> Оба утверждения неверны </b></p>

Ответ: 2

Комментарий: AP00932

**<p><b> Чтобы отмечать, где кончается один файл и начинается другой, полезно иметь </b></p>**

1. <p><b Специальный символ</b></p>

2. <p><b> Специальную команду</b></p>

3. <p><b> Специальное число </b></p>

Ответ: 1

Комментарий: AP00933

**<p><b>Сколько существует способов написания программ, работающих с файлами </b></p>**

1. <p><b>1</b></p>

2. <p><b> 4</b></p>

3. <p><b> 2 </b></p>

Ответ: 3

Комментарий: AP00934

## 10 тема

**<p><b>Какому if соответствует else?**

**if (x)**

**if (у) cout << "1";**

**else cout << "2"; </b></p>**

1. <p><b>Обоим</b></p>

2. <p><b>if(y)</b</p>

3. <p><b>Ни одному</b</p>

4. <p><b>if(x)</b></p>

Ответ: 2

Комментарий: AP010001

**<p><b>В языке C/C++ под вложенностью операторов if-else подразумевается …</b></p>**

1. <p><b>Дополнительное условие в операторе</b></p>

2. <p><b>Осуществление ветвления при помощи операторов</b></p>

3. <p><b>Тернарный оператор</b></p>

4. <p><b>Оператор if, имеющий ещё один if или else</b></p>

Ответ: 4

Комментарий: AP010002

**<p><b>В языке C/C++ оператор else соответствует ...</b></p>**

1. <p><b>Первому объявленному оператору if</b></p>

2. <p><b>Каждому объявленному оператору if</b></p>

3. <p><b>Ближайшему оператору if, не имеющему оператора else</b></p>

4. <p><b>Оператору if, имеющему несколько аргументов в условии</b></p>

Ответ: 3

Комментарий: AP010003

**<p><b>Каков результат выполнения программы если guess = 100?**

**#include <iostream>**

**int main()**

**{**

**int magic = 123;**

**int guess;**

**cout << "Enter Your Guess: ";**

**cin >> guess;**

**if(guess == magic) {**

**cout << “Right.\n”;**

**cout << guess << “ Is a Magic Number.\n”;**

**}**

**else {**

**cout << “Wrong.”;**

**if(guess > magic) cout << “ Too High.\n”;**

**else cout << “ Too Low.\n”;**

**}**

**return 0;**

**}</b></p>**

1. <p><b>Wrong. Too Low</b></p>

2. <p><b>Wrong</b></p>

3. <p><b>Right</b></p>

4. <p><b>Too High</b></p>

Ответ: 1

Комментарий: AP010004

**<p><b>Какому if соответствует else?**

**if(x) {**

**if(y) cout << “1”;**

**}**

**else cout << “2”;**

**</b></p>**

1. <p><b>if(y)</b></p>

2. <p><b>Обоим</b></p>

3. <p><b>Отсутствует близлежащий if</b></p>

4. <p><b>if(x)</b></p>

Ответ: 4

Комментарий: AP010005

**<p><b>Какие операторы используются для вложенности в конструкцию if-else?</b></p>**

1. <p><b>do-while</b></p>

2. <p><b>if-else</b></p>

3. <p><b>switch-case</b></p>

4. <p><b>when-otherwise</b></p>

Ответ: 2

Комментарий: AP010006

**<p><b>В чём принципиальное различие между if(condition1 && condition2) и**

**if(condition1) {**

**if(condition2) {**

**cout << “Something”;**

**}}**

**</b></p>**

<p><b>Второй пример выполняется, когда истинно первое условие и ложно второе</b></p>

<p><b>Первый пример выполняется, когда одно условие ложно</b></p>

<p><b>Второй пример выполняется, когда истинны оба условия</b></p>

<p><b>Первый пример выполняется, когда одно условие истинно</b></p>

Ответ: 3

Комментарий: AP010007

**<p><b>Хороший язык должен обеспечивать реализацию форм управления процессом выполнения программ:</b></p>**

1. <p><b>Выполнение последовательности операторов</b></p>

2. <p><b>Выполнение определенной последовательности операторов до тех пор, пока некоторое условие истинно</b></p>

3. <p><b>Использование проверки истинности условия для выбора между различными возможными способами действия</b></p>

4. <p><b>Все ответы верны</b></p>

5. <p><b>Нет верного ответа</b></p>

Ответ: 4

Комментарий: AP010008

**<p><b>Из приведенных ниже операций выберите операции отношения (сравнения) в языке С/С++:</b></p>**

1. <p><b> == </b></p>

2. <p><b> <> </b></p>

3. <p><b> != </b></p>

4. <p><b> >= </b></p>

5. <p><b> = </b></p>

6. <p><b> ++ </b></p>

7. <p><b> make </b></p>

Ответ: 1, 3, 4

Комментарий: AP010009

**<p><b>Выберите выражения, где переменная X будет соответствовать понятию “истина”: </b></p>**

1. <p><b> X = 1 </b></p>

2. <p><b> X = -310 </b></p>

3. <p><b> X = (10==20) </b></p>

4. <p><b> X = 0 </b></p>

5. <p><b> X = (20>2) </b></p>

Ответ: 1, 2, 5

Комментарий: AP010010

**<p><b>Выберите условия, в результате выполнения которых, переменная X будет присвоено значение 10:**

**int main()**

**{**

**int X, y=1;**

**//Здесь должно быть условие из ответа;**

**return 0;**

**}**

**</b></p>**

1. <p><b>

if (y)

X=10;

else

X=20;

</b></p>

2. <p><b>

if (y==0)

X=10;

else

X=20;

</b></p>

3. <p><b>

if (-30)

X=10;

else

X=20;

</b></p>

4. <p><b>

if (0)

X=10;

else

X=20;

</b></p>

5. <p><b>

if (y!=0)

X=10;

else

X=20;

</b></p>

Ответ: 1, 3, 5

Комментарий: AP010011

**<p><b>Сколько раз выведется на консоль значение переменной age?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int age = 20;**

**while (age <= 65)**

**{**

**age++;**

**if ((age % 20) == 0)**

**cout << age << endl;**

**if (age = 65)**

**cout << "До свидания!" << endl;**

**}**

**}**

**</b></p>**

1. <p><b> Один </b></p>

2. <p><b> Два </b></p>

3. <p><b> Три </b></p>

4. <p><b> Ни разу </b></p>

5. <p><b> Нет верного ответа </b></p>

Ответ: 4.

Комментарий: AP010012

**<p><b>Понятию “истина” соответствуют: </b></p>**

1. <p><b> Все неотрицательные числа </b></p>

2. <p><b> Все неотрицательные числа, кроме нуля </b></p>

3. <p><b> Все отрицательные и неотрицательные числа, кроме нуля</b></p>

4 <p><b> Только число 1 </b></p>

Ответ: 3.

Комментарий: AP010013

**<p><b>Что выведется на консоль, после завершения программы?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**int i = 0, x = 1;**

**if (i = x)**

**{**

**cout << x << endl;**

**}**

**if (i)**

**{**

**cout << i << endl;**

**}**

**}**

**</p></b>**

1. <p><b> Значение переменной x </b></p>

2. <p><b> Значение переменной i </b></p>

3. <p><b> Значение переменной x и переменной i </b></p>

4 <p><b> Ничего не выведется</b></p>

Ответ: 3.

Комментарий: AP010014

**<p><b>Как можно записать следующий оператор при помощи условного оператора «?»:**

**if (fvalue >= 0.0)**

**fvalue = fvalue;**

**else**

**fvalue = -fvalue;</p></b>**

1. <p><b>fvalue= (fvalue < 0.0) ? - fvalue : fvalue;</p></b>

2. <p><b>fvalue= (fvalue >= 0.0) ? fvalue : - fvalue;</p></b>

3. <p><b>fvalue= (fvalue < 0.0) ? fvalue : - fvalue;</p></b>

4. <p><b>fvalue= (fvalue >= 0.0) ? fvalue;</p></b>

Ответ: 1, 2

Комментарий: AP010015

**<p><b>Что выведется в консоль при запуске программы?**

**#include <iostream>**

**#include <math.h>**

**using namespace std;**

**int main ()**

**{**

**float fbalance = 1240, fpayment = 246;**

**cout << "В настоящий момент вы ";**

**cout << ((fpayment > fbalance) ? " переплатили $" : "выплатили $");**

**cout << ((fpayment > fbalance) ? abs(fbalance - fpayment) : fpayment);**

**return(0);**

**}</p></b>**

1. <p><b>В настоящий момент вы переплатили $994</p></b>

2. <p><b>994</p></b>

3. <p><b>В настоящий момент вы выплатили $246</p></b>

4. <p><b>В настоящий момент вы выплатили $1240</p></b>

Ответ: 3

Комментарий: AP010016

**<p><b>Каков синтаксис условного оператора «?»?</p></b>**

1. <p><b>Выражение\_условие ? действие\_если\_истина : действие\_если\_ложь;</p></b>

2. <p><b>Выражение\_условие ? действие\_если\_истина</p></b>

3. <p><b>Выражение\_условие ? действие\_если\_ложь : действие\_если\_истина;</p></b>

4. <p><b>Нет верных вариантов среди остальных</p></b>

Ответ: 1

Комментарий: AP010017

**<p><b>Что позволяет записать условный оператор “?”?</p></b>**

1. <p><b>Полное условие проверки</p></b>

2. <p><b>Краткое условие проверки</p></b>

3. <p><b>Цикл с определенным количеством повторений</p></b>

4. <p><b>Описание всех предупреждений в проекте</p></b>

Ответ: 2

Комментарий: AP010018

**<p><b>Вместо какого оператора может быть использован условный оператор “?”?</p></b>**

1. <p><b>if</p></b>

2. <p><b>while</p></b>

3. <p><b>do-while</p></b>

4. <p><b>if-else</p></b>

Ответ: 4

Комментарий: AP010019

<p><b>Что выведется в консоль при запуске программы?

#include <iostream>

using namespace std;

**int main ()**

**{**

**int ivalue = 1240;**

**ivalue = (ivalue <= 0) ? ivalue : - ivalue;**

**ivalue = (ivalue == 1) ? ivalue – 1000 : (int)ivalue / 2;**

**ivalue = (ivalue < 0) ? ivalue + 1000 : ivalue – 1;**

**cout << ivalue << endl;**

**return(0);**

**}</p></b>**

1. <p><b>0</p></b>

2. <p><b>994</p></b>

3. <p><b>380</p></b>

4. <p><b>1240</p></b>

Ответ: 3

Комментарий: AP010020

**<p><b>Является ли оператор if-else-if самостоятельным, либо же он является формой записи?</b></p>**

1.<p><b>Является собственным оператором</b></p>

2.<p><b>Является формой записи оператора if</b></p>

3.<p><b>Является формой записи оператора switch</b></p>

4.<p><b>Не является оператором выбора</b></p>

Ответ: 2

Комментарий: AP010021

**<p><b>Что будет выведено в консоль, при выполнении данного примера?**

**#include <stdio.h>**

**#include <stdlib.h>**

**int main()**

**{**

**int age = 99;**

**if ( age < 100 ) {**

**printf ("Вы очень молоды!\n" );**

**}**

**else if ( age == 100 ) {**

**printf( "Молодость уже позади\n" );**

**}**

**else {**

**printf( "Столько не живут\n" );**

**}**

**return 0;**

**}</b></p>**

1.<p><b>Вы очень молоды!</b></p>

2.<p><b>Молодость уже позади</b></p>

3.<p><b>Столько не живут</b></p>

4.<p><b>Пример не будет работать</b></p>

Ответ: 1

Комментарий: AP010022

**<p><b>Являются ли обязательными фигурные скобки при использовании if-else-if</b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 2

Комментарий: AP010023

**<p><b>Какая конструкция позволяет указывать неограниченное количество условий?</b></p>**

1.<p><b>if ( <условие> ) { <код> } else { <код> } else { <код> }</b></p>

2.<p><b>if ( <условие> ) { <код> } else if ( <условие > ) { <код>

} else { <код> }</b></p>

3.<p><b>if ( <условие> ) { <код> } else { <код> } if ( <условие> ) { <код> }</b></p>

Ответ: 2

Комментарий: AP010024

**<p><b>Можно ли совместно использовать операторы if-else-if и switch</b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 1

Комментарий: AP010025

**<p><b>Нужно ли в конструкции if-else-if после фигурных скобок ( {} ) ставить точку с запятой (;)?</b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 2

Комментарий: AP010026

**<p><b>Что будет выведено на консоль, после выполнения данной части программы:**

**int a = 7;**

**if(a < 4)**

**cout << "your mark is bad" << endl;**

**else if(a >= 4 && a < 8) {**

**cout << "your mark is good" << endl;**

**cout << "but you can study better" << endl;**

**}**

**else if(a >= 8 && a <= 10)**

**cout << "your mark is very good" << endl;**

**</b></p>**

1.<p><b>your mark is bad</b></p>

2.<p><b>

your mark is good

but you can study better

</b></p>

3.<p><b>your mark is good</b></p>

4.<p><b>your mark is very good</b></p>

Ответ: 2

Комментарий: AP010027

**<p><b>Что выведется в консоль при запуске программы?**

**#include <iostream>**

**int main()**

**{**

**int a = 5;**

**if (a > 10) a = a / a;**

**else if (a < 20) a++;**

**else if (-900 < a) a = a \* 2;**

**else a = 0;**

**std::cout << a;**

**}</b></p>**

1. <p><b>1</b></p>

2. <p><b>6</b></p>

3. <p><b>10</b></p>

4. <p><b>0</b></p>

Ответ: 2

Комментарий: AP010028

**<p><b>Что выведется в консоль при запуске программы?**

**#include <iostream>**

**int main()**

**{**

**int a = 5;**

**if (a < 7 && a % 2 == 0) a += a;**

**else if (a < 20) a = 0;**

**else a = a + a++;**

**std::cout << a;**

**}</b></p>**

1. <p><b>5</b></p>

2. <p><b>10</b></p>

3. <p><b>0</b></p>

4. <p><b>11</b></p>

Ответ: 3

Комментарий: AP010029

**<p><b>В каком случае фигурные скобки для действия необходимы?</b></p>**

1. <p><b>Задано более одного условия (&&, ||)</b></p>

2. <p><b>Количество условных операторов if больше трех</b></p>

3. <p><b>Действие содержит более одной команды</b></p>

4. <p><b>Количество условных операторов else больше трех</b></p>

Ответ: 3

Комментарий: AP010030

**<p><b>Что выведется в консоль при запуске программы?**

**#include <iostream>**

**int main()**

**{**

**int a = 5;**

**switch (a)**

**{**

**case 0: a = 0;**

**case 1: a = a;**

**case 5: a \*= 2;**

**case 10: a = -10;**

**}**

**if (a < 0 || a == 5) a \*= a;**

**else if (a == 10) a++;**

**else a = 0;**

**std::cout << a;**

**}</b></p>**

1. <p><b>1</b></p>

2. <p><b>6</b></p>

3. <p><b>10</b></p>

4. <p><b>0</b></p>

Ответ: 2

Комментарий: AP010031

**<p><b>Что произойдет, если не поставить break в конце case в switch?</b></p>**

1. <p><b>Начнется выполнение следующего case</b></p>

2. <p><b>Ничего не произойдет</b></p>

3. <p><b>Возникнет ошибка</b></p>

4. <p><b>Вся конструкция switch будет пропущена компилятором</b></p>

Ответ: 1

Комментарий: AP010032

**<p><b>Какое максимальное количество элементов case в switch?</b></p>**

1. <p><b>10</b></p>

2. <p><b>100</b></p>

3. <p><b>Указывается в скобках</b></p>

4. <p><b>Бесконечно много</b></p>

Ответ: 4

Комментарий: AP010033

**<p><b>Что значит default в switch?</b></p>**

1. <p><b>Первое действие</b></p>

2. <p><b>Действие по умолчанию</b></p>

3. <p><b>Последнее действие</b></p>

4. <p><b>Те же способности, что и у case</b></p>

Ответ: 2

Комментарий: AP010034

**<p><b> Что выведется в консоль при запуске программы с числом 10?**

**#include <iostream>**

**using namespace std;**

**int main()**

**{**

**setlocale(0, "");**

**double num;**

**cout << "Введите произвольное число: ";**

**cin >> num;**

**if (num < 10) {**

**cout << "Это число меньше 10." << endl;**

**}**

**else {**

**cout << "Это число больше либо равно 10." << endl;**

**}**

**return 0;**

**}</b></p>**

1. <p><b>Это число равно 10</b></p>

2. <p><b>Это число меньше 10</b></p>

3. <p><b>Это число больше либо равно 10</b></p>

4. <p><b>Ошибка</b></p>

Ответ: 3

Комментарий: AP010035

**<p><b> Возможно ли использование нескольких else относящихся к одному if ?</b></p>**

1. <p><b>Нет, не возможно</b></p>

2. <p><b>Да, с конструкцией if{...}; else{...} ;else{...}</b></p>

3. <p><b>Да, с конструкцией if{...}; else if{...}; else{...} </b></p>

4. <p><b>Да, с конструкцией if{...}; if{...}; else{...}</b></p>

Ответ: 3

Комментарий: AP010036

**<p><b> Какой условный оператор может быть использован вместо оператора if-else?</b></p>**

1. <p><b>”!”</b></p>

2. <p><b>”?”</b></p>

3. <p><b>”&”</b></p>

4. <p><b>”%”</b></p>

Ответ: 2

Комментарий: AP010037

**<p><b> Является ли оператор if else - самостоятельным оператором?</b></p>**

1. <p><b>Да, является</b></p>

2. <p><b>Нет, не является</b></p>

Ответ: 2

Комментарий: AP010038

**<p><b> Можно ли записать программу написанную с использованием “if else” с помощью оператора “switch” ?</b></p>**

1. <p><b>Да</b></p>

2. <p><b>Нет</b></p>

Ответ: 1

Комментарий: AP010039

**<p><b> Сколько всего возможно записей “else if” с оператором “if else”?</b></p>**

1. <p><b>1</b></p>

2. <p><b>5</b></p>

3. <p><b>Неограниченное количество</b></p>

4. <p><b>”else if” нельзя использовать с оператором “if else”.</b></p>

Ответ: 3

Комментарий: AP010040

## 13 тема

<p><b>Что такое битовая маска?</b></p>

1. <p><b>Определённые данные, которые используются для маскирования — выбора отдельных битов или полей из нескольких битов из двоичной строки или числа.</b></p>

2. <p><b>Функция, которая используется для маскирования — выбора отдельных битов или полей из нескольких битов из двоичной строки или числа.</b></p>

3. <p><b>Операция, которая используется для маскирования — выбора отдельных битов или полей из нескольких битов из двоичной строки или числа.</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0013001

<p><b>Какие операторы являются побитовыми?</b></p>

1. <p><b>&</b></p>

2. <p><b>~</b></p>

3. <p><b>||</b></p>

4. <p><b>&&</b></p>

Ответ: 1, 2

Комментарий: AP0013002

<p><b>Результат выполнения данной операции 5^2?</b></p>

1. <p><b>1</b></p>

2. <p><b>10</b></p>

3. <p><b>7</b></p>

4. <p><b>6</b></p>

Ответ: 3

Комментарий: AP0013003

<p><b>В дополнительном коде старший разряд является: </b></p>

1. <p><b>Всегда положительным</b></p>

2. <p><b>Всегда отрицательным</b></p>

3. <p><b>Знаковым</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 3

Комментарий: AP0013004

<p><b>Число 111000101, записанное в дополнительном коде является:</b></p>

1. <p><b>Положительным</b></p>

2. <p><b>Отрицательным</b></p>

3. <p><b>Его нельзя предстваить в дополнительном коде</b></p>

4. <p><b>Нет верного ответа </b></p>

Ответ: 2

Комментарий: AP0013005

<p><b>Результат выполнения данной операции 11<<3?</b></p>

1. <p><b>true</b></p>

2. <p><b>1221</b></p>

3. <p><b>1331</b></p>

4. <p><b>33</b></p>

Ответ: 3

Комментарий: AP0013006

<p><b></b>Побитовое отрицание инвертирует число:</p>

1. <p><b>отдельными битами</b></p>

2. <p><b>целым числом</b></p>

3. <p><b>сначала 1, потом 0</b></p>

4. <p><b>сначала 0, потом 1</b></p>

Ответ: 2

Комментарий: AP0013007

<p><b> Выберите ВТОРОЙ шаг преобразования числа в дополнительный код:</b></p>

1. <p><b>добавить единицу </b></p>

2. <p><b>инвертировать все биты числа </b></p>

3. <p><b>отнять единицу </b></p>

4. <p><b>заменить все единицы на ноль и наоборот </b></p>

Ответ: 1

Комментарий: AP0013008

<p><b>Результат выполнения программы:

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

char a = 3;

char b = 8;

char c = a & b;

printf("%d", c);

cin >> a;

}</b></p>

1. <p><b>24</b></p>

2. <p><b>11</b></p>

3. <p><b>0</b></p>

4. <p><b>11000</b></p>

Ответ: 3

Комментарий: AP0013009

<p><b>Побитовое произведение чисел 31 и 17 даст:</b></p>

1. <p><b>00010001</b></p>

2. <p><b>00001000</b></p>

3. <p><b>1000001111</b></p>

4. <p><b>00011111</b></p>

Ответ: 1

Комментарий: AP0013010

<p><b>Выберите оператор для исключающего ИЛИ:</b></p>

1. <p><b>||</b></p>

2. <p><b>|</b></p>

3. <p><b>^</b></p>

4. <p><b>~</b></p>

Ответ: 3

Комментарий: AP0013011

<p><b>Результат операции 00010011 >> 3:</b></p>

1. <p><b>10011</b></p>

2. <p><b>00100</b></p>

3. <p><b>00000010</b></p>

4. <p><b>00000110</b></p>

Ответ: 3

Комментарий: AP0013012

<p><b>Результат операции new = target ^ ( 1 << offset ):</b></p>

1. <p><b> установка 1 в отдельный разряд числа </b></p>

2. <p><b> установка 0 в отдельный разряд числа </b></p>

3. <p><b> применение исключающего ИЛИ со сдвигом влево </b></p>

4. <p><b> проверка разряда на наличие 1 или 0</b></p>

Ответ: 2

Комментарий: AP0013013

<p><b> В записи new = target & ( 1 < offset):</b></p>

1. <p><b>target это количество разрядов на которое требуется сдвинуть единицу от правого края влево</b></p>

2. <p><b>target - это число в котором мы желаем проверить некий разряд на наличие 0 или 1</b></p>

3. <p><b>target - это число для которого применяется операция &</b></p>

4. <p><b>target - это число для которого применяется операция побитового И</b></p>

Ответ: 2

Комментарий: AP0013014

<p><b> Выберите верное утверждение:</b></p>

1. <p><b> Побитовое И. Бинарный оператор. Результатом выражения a&&b является число, каждый бит которого в двоичном представлении равен результату сравнения соответствующих битов чисел a и b : значение бита равно 1, если оба сравниваемых бита равны 1. В противном случае значение бита равно 0</b></p>

2. <p><b> Побитовое исключающее ИЛИ. Унарный оператор. Результатом выражения ~a является число, которое получается побитовым инвертированием числа a </b></p>

3. <p><b> Побитовое ИЛИ. Бинарный оператор. Результатом выражения a|b является число, каждый бит которого в двоичном представлении равен результату сравнения соответствующих битов чисел a и b : значение бита равно 1, если хотя бы один из сравниваемых битов равен 1. В противном случае значение бита равно 0</b></p>

4. <p><b>Сдвиг вправо. Бинарный оператор. В двоичном представлении числа, указанном справа от оператора, выполняется сдвиг всех битов влево на число позиций, указанных слева от оператора. При этом старший бит знака остается неизменным, а выходящие за диапазон младшие биты теряются</b></p>

Ответ: 3

Комментарий: AP0013015

<p><b> Определим бинарное машинное представление для отрицательного числа -75:</b></p>

1. <p><b>01001011</b></p>

2. <p><b>10110101</b></p>

3. <p><b>11001011</b></p>

4. <p><b>10001100</b></p>

Ответ: 2

Комментарий: AP0013016

<p><b>Определенные данные которые используются для выбора отдельных битов или полей из нескольких битов из двоичной строки или числа – это:</b></p>

1. <p><b>бит</b></p>

2. <p><b>битовая маска</b></p>

3. <p><b>битовый массив</b></p>

4. <p><b>битовая операция</b></p>

Ответ: 2

Комментарий: AP0013017

<p><b> Результат выполнения программы:

#include <iostream>

using namespace std;

void main()

{

setlocale(LC\_CTYPE, "Russian");

char a = 15;

char b = 11;

char c = a | b;

printf("%d", c);

cin >> a;

}</b></p>

1. <p><b>15</b></p>

2. <p><b>1</b></p>

3. <p><b>5</b></p>

4. <p><b>11000</b></p>

Ответ: 1

Комментарий: AP0013018

<p><b>Результат операции 10111011 & 00001000:</b></p>

1. <p><b>00001011</b></p>

2. <p><b>00000011</b></p>

3. <p><b>00100011</b></p>

4. <p><b>00001000</b></p>

Ответ: 4

Комментарий: AP0013019

<p><b>Какие утверждения верны?</b></p>

1. <p><b> Для определения знака числа используют младший бит в позиционной записи. Нулевой старший бит соответствует положительному числу, а единичный старший бит соответствует отрицательному числу.</b></p>

2. <p><b>Битовые операции - это особенность языка.</b></p>

3. <p><b>Битовую маску возможно использовать для получения значения бита.</b></p>

4. <p><b>При использовании побитового отрицания знак результата всегда будет соответствовать знаку исходного числа.</b></p>

Ответ: 3

Комментарий: AP0013020

<p><b>Что такое битовый сдвиг?</b></p>

1. <p><b>Когда биты уходят.</b></p>

2. <p><b>Это изменение значения бита.</b></p>

3. <p><b>Это изменение позиций бит в машинном слове. </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 3

Комментарий: AP0013021

<p><b>Выберете неверное утверждение.</b></p>

1. <p><b>Битовые операции предусмотрены для работы с отдельными битами.</b></p>

2. <p><b>Битовые операции можно применять к переменным вещественного типа.</b></p>

3. <p><b>Битовые операции применяются в языках программирования и цифровой технике, изучаются в дискретной математике. </b></p>

4. <p><b>Все утверждения верны.</b></p>

Ответ: 2

Комментарий: AP0013022

<p><b>Если a = 0 и b = 1, то чего будет равно a & b?</b></p>

1. <p><b>10</b></p>

2. <p><b>1</b></p>

3. <p><b>0</b></p>

4. <p><b>2</b></p>

Ответ: 3

Комментарий: AP0013023

<p><b> Если a = 0 и b = 1, то чего будет равно a | b?</b></p>

1. <p><b>1</b></p>

2. <p><b>0</b></p>

3. <p><b>10</b></p>

4. <p><b>2</b></p>

Ответ: 1

Комментарий: AP0013024

<p><b> Если a = 1, то чего будет равно a << 1?</b></p>

1. <p><b>1</b></p>

2. <p><b>10</b></p>

3. <p><b>0</b></p>

4. <p><b>2</b></p>

Ответ: 4

Комментарий: AP0013025

<p><b>Для изменения всех битов числа на противоположные используется:</b></p>

1. <p><b>операция |</b></p>

2. <p><b>операция ^</b></p>

3. <p><b>операция &</b></p>

4. <p><b>операция ~</b></p>

Ответ: 4

Комментарий: AP0013026

<p><b>В двоичном числе 1010010 надо выделить второй и третий разряды по счету справа. Чему равна маска?</b></p>

1. <p><b>000</b></p>

2. <p><b>111</b></p>

3. <p><b>110</b></p>

4. <p><b>123</b></p>

Ответ: 3

Комментарий: AP0013027

<p><b> Какой результат 0110 >> 2 в двоичной системе?</b></p>

1. <p><b>13</b></p>

2. <p><b>0001</b></p>

3. <p><b>4</b></p>

4. <p><b>9</b></p>

Ответ: 1

Комментарий: AP0013028

<p><b> Какой результат 5 | 12 в десятичной системе?</b></p>

1. <p><b>13</b></p>

2. <p><b>0001</b></p>

3. <p><b>4</b></p>

4. <p><b>9</b></p>

Ответ: 2

Комментарий: AP0013029

<p><b> Какой результат 5 & 12 в десятичной системе?</b></p>

1. <p><b>13</b></p>

2. <p><b>0001</b></p>

3. <p><b>4</b></p>

4. <p><b>9</b></p>

Ответ: 3

Комментарий: AP0013030

<p><b> Какой результат 5 ^ 12 в десятичной системе?</b></p>

1. <p><b>13</b></p>

2. <p><b>0001</b></p>

3. <p><b>4</b></p>

4. <p><b>9</b></p>

Ответ: 4

Комментарий: AP0013031

<p><b>Для чего используется операция &?</b></p>

1. <p><b> Используется для дополнения к целому.</b></p>

2. <p><b> Используется для включения битов</b></p>

3. <p><b> Используется для маскирования некоторого множества битов</b></p>

4. <p><b> Используется для сдвигов</b></p>

Ответ: 3

Комментарий: AP0013032

<p><b> Для чего используется операция ! :</b></p>

1. <p><b> Используется для дополнения к целому.</b></p>

2. <p><b> Используется для включения битов</b></p>

3. <p><b> Используется для маскирования некоторого множества битов</b></p>

4. <p><b> Используется для сдвигов</b></p>

Ответ: 2

Комментарий: AP0013033

<p><b> Для чего используется операция ~ ?</b></p>

1. <p><b> Используется для дополнения к целому.</b></p>

2. <p><b> Используется для включения битов</b></p>

3. <p><b> Используется для маскирования некоторого множества битов</b></p>

4. <p><b> Используется для сдвигов</b></p>

Ответ: 1

Комментарий: AP0013034

<p><b> Для чего используются операции << и >>?</b></p>

1. <p><b> Используется для дополнения к целому.</b></p>

2. <p><b> Используется для включения битов</b></p>

3. <p><b> Используется для маскирования некоторого множества битов</b></p>

4. <p><b> Используется для сдвигов</b></p>

Ответ: 4

Комментарий: AP0013035

## 12+14 темы

**<p><b>Если flizny[] — массив, то запись &flizny[0] равносильна</b></p>**

1. <p><b>\*( flizny + 0) </b></p>

2. <p><b>flizny</b></p>

3. <p><b>flizny[0]</b></p>

Ответ: 2

Комментарий: AP00101

**<p><b>При компиляции программы на С память компьютера разбивается на области, содержащие:</b></p>**

1. <p><b>код программы</b></p>

2. <p><b>глобальные данные</b></p>

3. <p><b>локальные данные</b></p>

4. <p><b>стек</b></p>

5. <p><b>динамически распределяемую область памяти (heap)</b></p>

Ответ: 1, 2, 4, 5

Комментарий: AP00102

**<p><b>Динамическая память (heap) — это…</b></p>**

1. <p><b>просто свободная область памяти, с которой работают при помощи функций динамического выделения памяти malloc() и free(), а также операций new и delete</b></p>

2. <p><b>память, которая выделяется еще до начала работы программы, на стадии компиляции и сборки</b></p>

3. <p><b>память, которая автоматически выделяется под аргументы и локальные переменные функции</b></p>

Ответ: 1

Комментарий: AP00103

**<p><b>Что описывает следующий оператор:**

**int \*(\* (\*ifunct\_ptr) (int)) [5]</b></p>**

1. <p><b>ifunct\_ptr функция, которая получает целочисленный параметр и возвращает указатель на массив из пяти указателей типа int</b></p>

2. <p><b>ifunct\_ptr указатель на функцию, который получает целочисленный параметр и возвращает указатель на массив из пяти переменных типа int</b></p>

3. <p><b>ifunct\_ptr указатель на функцию, который получает целочисленный параметр и возвращает указатель на массив из пяти указателей типа int</b></p>

Ответ: 3

Комментарий: AP00104

**<p><b>Определите** **значение переменной ref:**

**int i=1;**

**int &ref =i;.**

**ref += 10;</b></p>**

1. <p><b>10</b></p>

2. <p><b>1</b></p>

3. <p><b>11</b></p>

4. <p><b>2</b></p>

Ответ: 3

Комментарий: AP00105

**<p><b>Есть ли ошибка в следующей записи:**

**int& riresult\_b;</b></p>**

1. <p><b>неправильно: нет начального значения</b></p>

2. <p><b>все верно</b></p>

3. <p><b>синтаксис C повзоляет описывать таким образом переменные, но не рекомендуется"</b></p>

Ответ: 1

Комментарий: AP00106

**<p><b>Запись** **int \*const P2 говорит, что P2 это - </b></p>**

1. <p><b>нет правильного ответа</b></p>

2. <p><b>константный указатель на константу</b></p>

3. <p><b>указатель на константу</b></p>

4. <p><b>константный указатель</b></p>

Ответ: 4

Комментарий: AP00107

**<p><b>** **Запись const int \*P1 говорит, что P1 это - </b></p>**

1. <p><b>нет правильного ответа</b></p>

2. <p><b>константный указатель на константу</b></p>

3. <p><b>указатель на константу</b></p>

4. <p><b>константный указатель</b></p>

Ответ: 3

Комментарий: AP00108

**<p><b>Определите значение \*ptr и \*(ptr + 2):**

**int \*ptr;**

**static int boop[4] = {12, 21, 121, 212};**

**ptr = boop;<b></p>**

1. <p><b>21 и 121</b></p>

2. <p><b>12 и 121</b></p>

3. <p><b>12 и 212</b></p>

4. <p><b>21 и 212</b></p>

Ответ: 2

Комментарий: AP00109

**<p><b>Выразите адрес grid[22] [0] различными способами</b></p>**

1. <p><b>grid[22]</b></p>

2. <p><b>\*(\*(grid + 22) + 0)</b></p>

3. <p><b>&grid[22][0]</b></p>

Ответ: 1, 3

Комментарий: AP00110

**<p><b>Что будет выведено на печать:**

**long double Num = 10;**

**long double Flag;**

**long double \*ptr = &Num;**

**Flag = \*ptr;**

**cout << Flag;</b></p>**

1. <p><b>адрес переменно Flag</b></p>

2. <p><b>адрес переменной Num</b></p>

3. <p><b>нет правильного ответа</b></p>

4. <p><b>значение переменной Num</b></p>

Ответ: 4

Комментарий: AP00111

**<p><b>Верна ли запись изменение содержимого перемменой Let с помощью указателя:**

**char Let = 'T';**

**void \*ptr;**

**ptr = &Let;**

**\*ptr = 'L';</b></p>**

1. <p><b>да</b></p>

2. <p><b>нет, ptr не приведен к типу char\*</b></p>

3. <p><b>нет, ptr не приведен к типу int\*</b></p>

4. <p><b>нет правильного ответа</b></p>

Ответ: 2

Комментарий: AP00112

**<p><b>В какик ситуациях операцию (&) применять нельзя:</b></p>**

1. <p><b>использовать вместе с константами</b></p>

2. <p><b>использовать в выражениях с арифметическими операциями</b></p>

3. <p><b>использовать вместе с регистровыми переменными</b></p>

4. <p><b>использовать вместе с переменными</b></p>

Ответ: 1, 2, 3

Комментарий: AP00113

**<p><b>В чем отличие между переменными psz и pszarray:**

**char \*psz = "File not ready";**

**char pszarray[] = "Drive not ready";</b></p>**

1. <p><b>значение psz можно изменить, а значение pszarray изменить нельзя</b></p>

2. <p><b>значение pszarray можно изменить, а значение psz изменить нельзя </b></p>

3. <p><b>нет правильного ответа</b></p>

Ответ: 1

Комментарий: AP00114

**<p><b>Какие операции можно выполнять над переменными типа указатель:</b></p>**

1. <p><b>присваивание</b></p>

2. <p><b>определение значения</b></p>

3. <p><b>получение адреса указателя</b></p>

4. <p><b>увеличение указателя</b></p>

5. <p><b>разность</b></p>

Ответ: 1, 2, 3, 4, 5

Комментарий: AP00115

**<p><b>Какой адрес будет выведен во втором случае, если в первом случае был выведен адрес 0x0068fdfc?**

**double Var;**

**double\* ptr = &Var;**

**cout << ptr << '\n';**

**ptr++;**

**cout << ptr;</b></p>**

1. <p><b>0x0068fe04</b></p>

2. <p><b>0x0068fe00</b></p>

3. <p><b>0x0068fd04</b></p>

3. <p><b>0x0068fd00</b></p>

Ответ: 1

Комментарий: AP00116

**<p><b>Какие значения будут выведены:**

**bool IsTrue = false;**

**bool\* pBool = &IsTrue;**

**cout << sizeof pBool << endl;**

**cout << sizeof(pBool) << endl;</b></p>**

1. <p><b>4 и 1</b></p>

2. <p><b>1 и 4</b></p>

3. <p><b>4 и 4</b></p>

4. <p><b>1 и 1</b></p>

Ответ: 3

Комментарий: AP00117

**<p><b>Какие операции нельзя выполнять над переменными типа указатель:</b></p>**

1. <p><b>складывать</b></p>

2. <p><b>перемножать</b></p>

3. <p><b>получать адрес указателя</b></p>

4. <p><b>вычитать</b></p>

Ответ: 1, 2

Комментарий: AP00118

**<p><b>Отметьте характеристики массива:</b></p>**

1. <p><b>отдельные объекты данных в массиве называются элементами</b></p>

2. <p><b>все элементы массива должны иметь одинаковый тип данных</b></p>

3. <p><b>все элементы располагаются в памяти компьютера последовательно, и индекс первого элемента равен нулю</b></p>

4. <p><b>имя массива является постоянным значением, представляющим собой адрес первого элемента массива </b></p>

Ответ: 1, 2, 3, 4

Комментарий: AP00119

**<p><b>Какой оператор обращается ко второму элементу массива?**

**int ivideo\_library[7];**

**int iweekend = 1;**

**int iweekday = 2;</b></p>**

1. <p><b>ivideo\_library[2]</b></p>

2. <p><b>ivideo\_library[iweekday]</b></p>

3. <p><b>ivideo\_library[iweekend + iweekday]</b></p>

4. <p><b>ivideo\_library[iweekday - iweekend]</b></p>

Ответ: 4

Комментарий: AP00120

**<p><b>Какие операции можно применять к данным типа int?</b></p>**

1. <p><b>арифметические операции</b></p>

2. <p><b>логические</b></p>

3. <p><b>операция конкатенации</b></p>

4. <p><b>операции отношения или сравнения</b></p>

Ответ: 1

Комментарий: AP00121

**<p><b>На какой символ указывает pArr?**

**char ArrayOfChar[] = {'W', 'O', 'R','L','D'};**

**char\* pArr = ArrayOfChar;**

**pArr += 3</b></p>**

1. <p><b>’W’</b></p>

2. <p><b>’R’</b></p>

3. <p><b>’D’ </b></p>

4. <p><b>’L’</b></p>

5. <p><b>’O’</b></p>

Ответ: 4

Комментарий: AP00122

**<p><b>Какие массивы можно иницализировать?</b></p>**

1. <p><b>внешние</b></p>

2. <p><b>автоматические</b></p>

3. <p><b>статические</b></p>

4. <p><b>регистровые</b></p>

Ответ: 1, 3

Комментарий: AP00123

**<p><b>Определите значение Mass[2][1][0]**

**int Mass[3] [2] [4] = {{1,2, 3, 4},{5,6,7,8}}, {{9,10,11,12},{13,14,15,16}}, {{17,18,19,20},{21,22,23,24}}</b></p>**

1. <p><b>9<br></b></p>

2. <p><b>5</b></p>

3. <p><b>17</b></p>

4. <p><b>21</b></p>

Ответ: 4

Комментарий: AP00124

**<p><b>Функция gets():</b></p>**

1. <p><b>преобразует null-символ в символ перевода строки</b></p>

2. <p><b>оставляет символы перевода строки и добавляет null-символ</b></p>

3. <p><b>преобразует символ перевода строки в null-символ</b></p>

Ответ: 3

Комментарий: AP00125

**<p><b>Функция puts():</b></p>**

1. <p><b>преобразует null-символ в символ перевода строки</b></p>

2. <p><b>оставляет символы перевода строки и добавляет null-символ</b></p>

3. <p><b>преобразует символ перевода строки в null-символ</b></p>

Ответ: 1

Комментарий: AP00126  
  
**<p><b>Выберите утверждения, которые относятся к массивам</b></p>**

1. <p><b>Конечный, последовательность набор данных одного типа, связанных общим именем</b></p>

2. <p><b>Это структура данных, состоящая из элементов одного типа, связанных между собой последовательно посредством указателей.</b></p>

3. <p><b>Элементы располагаются в последовательных ячейках памяти</b></p>

4. <p><b>Имя является указателем</b></p>

5. <p><b>Могут иметь как одно, так и более одного измерений.</b></p>

Ответ: 1,3,4,5

Комментарий:AP00127

**<p><b>Укажите правильные варианты инициализации и заполнения массива</b></p>**

1. <p><b>int arr = { 5,3,2,1,8 };</b></p>

2. <p><b> int arr[5];

for (int i = 1; i <=5; i++)

std::cin >> arr[i];</b></p>

3. <p><b>int arr[5] = { 5,3,2,1,8};</b></p>

4. <p><b>int arr[] = { 5,3,2,1,8 };.</b></p>

Ответ: 3,4

Комментарий AP00128

**<p><b>Укажите размер массива char arr[] = {"размер массива"};</b></p>**

1. <p><b>14 байт</b></p>

2. <p><b>15 байт/b></p>

3. <p><b>28 байт</b></p>

4. <p><b>1 байт</b></p>

5. <p><b>Нет верного ответа</b></p>

Ответ: 2

Комментарий: AP00129

**<p><b>что будет выведено в результате выполнения кода?**

**int arr[5] = { 3,4,5,7,8 };**

**int\* ptr1 = &arr[0];**

**int\* ptr2 = arr;**

**int\* ptr3;**

**ptr3 = arr;**

**std::cout << \*ptr1<<" "<<\*ptr2++<<" "<<\*ptr3<<" ";**

**</b></p>**

1. <p><b>0078FE5C 3 3</b></p>

2. <p><b>3 4 3</b></p>

3. <p><b>3 4 4</b></p>

4. <p><b>0078FE5C 5 3</b></p>

5. <p><b>3 3 3</b></p>

Ответ: 5

Комментарий: AP00130

**<p><b>const int\* ptr = &value; ptr - это: </b></p>**

1. <p><b>Константный указатель на константу</b></p>

2. <p><b>Константный указатель</b></p>

3. <p><b>Указатель на константу</b></p>

4. <p><b>Ничего из вышеперечисленного </b></p>

Ответ:3

Комментарий: AP00131

**<p><b> что будет выведено в результате выполнения кода?**

**char arr[] = "Привет,мир!";**

**std::cout << strlen(arr) << " " << sizeof(arr) /sizeof( arr[0]);**

**</b></p>**

1. <p><b>12 12</b></p>

2. <p><b>11 11</b></p>

3. <p><b>12 11</b></p>

4.11 12

Ответ:4

Комментарий: AP00132

**<p><b> Что из нижеперечисленного относится к умным указателям? </b></p>**

1. <p><b>shared\_ptr</b></p>

2. <p><b>unique\_ptr</b></p>

3. <p><b>auto\_ptr</b></p>

4. <p><b>ничего из вышеперчисленного. </b></p>

Ответ:1,2,3

Комментарий: AP00133

**<p><b> Для чего нужны умные указатели? </b></p>**

1. <p><b>Для обеспечения освобождения выделенной памяти при выходе объекта класса из области видимости</b></p>

2. <p><b>Для перебора элементов массива без использования цикла</b></p>

3. <p><b>Для вычисления размера n-мерного массива</b></p>

4. <p><b>Для возврата нескольких значений из функции</b></p>

5. <p><b>Таких не существует</b></p>

Ответ:1

Комментарий: AP00134

**<p><b> Укажите функции с верно указанными прототипами </b></p>**

1. <p><b>char \* gets( char \* string ); </b></p>

2. <p><b>void puts(const char \*str); </b></p>

3. <p><b>char fputs(const char \*str, FILE \*stream); </b></p>

4. <p><b>int fputs(const char \*str, FILE \*stream); </b></p>

5. <p><b>char \*fgets(char \*str, int num, FILE \*stream); </b></p>

Ответ:1,4,5

Комментарий: AP00135

**<p><b> Какие функции предназначены для работы с файлами? </b></p>**

1. <p><b>gets( ); </b></p>

2. <p><b>puts( ); </b></p>

3. <p><b>fputs( ); </b></p>

4. <p><b>sprintf( );</b></p>

5. <p><b>fgets( ) ; </b></p>

Ответ:3,5

Комментарий: AP00136

**К<p><b> Выберите верные утверждения</b></p>**

1. <p><b>Ссылка служит для задания объекту дополнительного имени</b></p>

2. <p><b>Ссылку можно переопределить</b></p>

3. <p><b>Ссылка не может быть «нулевой», то есть ссылку обязательно нужно чем-то инициализировать</b></p>

4. <p><b>Ссылка ничем не отличается от указателя</b></p>

5. <p><b>Существует арифметика ссылок</b></p>

Ответ:1,3

Комментарий: AP00137

К**<p><b> К указателям на переменные какого типа могут применяться арифметические действия?</b></p>**

1. <p><b>void</b></p>

2. <p><b>float</b></p>

3. <p><b>double</b></p>

4. <p><b>int</b></p>

5 <p><b>.char</b></p>

Ответ:2,3,4,5

Комментарий: AP00138

**<p><b> Какие операции можно выполнять над указателями?</b></p>**

1. <p><b>Вычитание целого числа из указателя</b></p>

2. <p><b>Вычитание одного указателя из другого</b></p>

3. <p><b>Сравнение указателей при помощи операций отношения, например == или >=</b></p>

4 <p><b>.Сложение двух указателей</b></p>

5. <p><b>Все варианты верны</b></p>

Ответ:1,2,3,

Комментарий: AP00139

**<p><b>что выведет на экран при выполнении фрагмента кода?**

**char arr[] = { "hello\0world" };**

**std::cout << strlen(arr) << " " << sizeof(arr);</b></p>**

1. <p><b>13 10</b></p>

2. <p><b>13 11</b></p>

3. <p><b>5 12</b></p>

4. <p><b>5 11</b></p>

5. <p><b>12 12</b></p>

**Ответ:3**

Комментарий: AP00140

**<p><b>Укажите неправильное использование операции определения адреса</b></p>**

1. <p><b>pivariable = &abc;(abc - переменная ) </b></p>

2. <p><b>pivariable = &48; </b></p>

3. <p><b>pivariable = &string10[0];(string10 - массив) </b></p>

4. <p><b>pivariable = &(iresult + 15); </b></p>

Ответ:2,4

Комментарий: AP00141

**<p><b>На что указывает переменные i и j в arr[ i ][ j ] </b></p>**

1. <p><b>i – номер строки, j – номер столбца</b></p>

2. <p><b>j – номер строки,i – номер столбца </b></p>

3. <p><b>i+j – номер строки,i-j – номер столбца</b></p>

Ответ:1

Комментарий: AP00142

**<p><b>Если указатель ptr типа float содержит адрес 008FF8B0 ,то какой адрес содержит ptr+4? </b></p>**

1. <p><b>408FF8B0</b></p>

2. <p><b>008FF8B4</b></p>

3. <p><b>008FF8C0</b></p>

4. <p><b>408FF8C0</b></p>

5. <p><b>008FF8B0</b></p>

Ответ:3

Комментарий: AP00143

**<p><b>Что из перечисленного ниже не используется для динамического выделения памяти? </b></p>**

1. <p><b>delete</b></p>

2. <p><b>new</b></p>

3. <p><b>calloc( )</b></p>

4. <p><b>malloc( )</b></p>

5. <p><b>free( )</b></p>

Ответ:1,5

Комментарий: AP00144

**<p><b>Что из перечисленного ниже относится к спецификаторам класса памяти </b></p>**

1. <p><b>auto</b></p>

2. <p><b>register</b></p>

3. <p><b>static</b></p>

4. <p><b>extern</b></p>

5. <p><b>Все варианты верны</b></p>

Ответ:5

Комментарий: AP00145

**<p><b>Как вычислить число элементов массива arr ?</b></p>**

1. <p><b>sizeof(arr); </b></p>

2. <p><b>sizeof(arr[ 0 ]); </b></p>

3. <p><b>sizeof(arr/arr[0]); </b></p>

4. <p><b>sizeof(arr)/sizeof(arr[0]); </b></p>

Ответ:4

Комментарий: AP00146

**<p><b>Что делает функция strcat( )?</b></p>**

1. <p><b>Выводит на экран две переданные ей строки</b></p>

2. <p><b>Копирует вторую строку в конец первой </b></p>

3. <p><b>Копирует первую строку в конец второй </b></p>

4. <p><b>Возвращает сумму ASCII кодов переданных ей строк </b></p>

Ответ:2

Комментарий: AP00147

**<p><b>Что делает функция strncmp( )?</b></p>**

1 <p><b>.Сравнивает первые n символов петровой строки с n символами второй строки</b></p>

2. <p><b>Конкатенирует две строки</b></p>

3. <p><b>Проверяет наличие в строке переданного символа </b></p>

4. <p><b>Возвращает строку символов, которые есть в первой строке, но которых нет во второй</b></p>

Ответ:1

Комментарий: AP00148

**<p><b>Где выделяется память при использовании оператора new?</b></p>**

1. <p><b>.В стеке</b></p>

2. <p><b>.В куче</b></p>

3. <p><b>.В глобальной памяти</b></p>

4. <p><b>.Нет верного ответа </b></p>

Ответ:2

Комментарий: AP00149

**<p><b>Сколько памяти занимает указатель типа double?</b></p>**

1. <p><b>.0</b></p>

2. <p><b>.4</b></p>

3. <p><b>.8</b></p>

4. <p><b>.Нет верного ответа</b></p>

Ответ:4

Комментарий: AP00150

**<p><b>Одномерный массив arr был создан с помощью оператора new, как очистить эту память?</b></p>**

1. <p><b>.delete arr; </b></p>

2. <p><b>.delete[]arr; </b></p>

3. <p><b>.free(arr); </b></p>

4. <p><b>.Все варианты верны</b></p>

Ответ:2

Комментарий: AP00151

**<p><b>Выберите утверждения, относящиеся к динамическому массиву в С++?</b></p>**

1. <p><b>.Позволяет использовать неконстантную переменную в качестве размера</b></p>

2. <p><b>.Размер массива автоматически увеличивается при заполнении </b></p>

3. <p><b>.Память выделяется в стеке</b></p>

4. <p><b>.При завершении программы память будет очищена даже без оператора delete</b></p>

Ответ:1,4

Комментарий: AP00152

**<p><b>В какой из следующих строк выполняется обращение к седьмому элементу массива? размер массива равен 10 </b></p>**

1. <p><b>mas[7]</b></p>

2. <p><b>mas(7)</b></p>

3. <p><b>mas</b></p>

4. <p><b>mas[6]</b></p>

Ответ:4

Комментарий: AP00153

**<p><b>Какая из следующих функций добавляет одну строку в конец другой?</b></p>**

1. <p><b>stringadd ();</b></p>

2. <p><b>Append ();</b></p>

3. <p><b>strcat ();</b></p>

4. <p><b>stradd();</b></p>

Ответ:3

Комментарий: AP00154

**<p><b>Укажите зарезервированное ключевое слово для динамического выделения памяти</b></p>**

1 <p><b>.create</b></p>

2. <p><b>malloc</b></p>

3. <p><b>new</b></p>

4. <p><b>value</b></p>

Ответ:3

Комментарий: AP00155

**<p><b>В какой из следующих записей используется операция разыменования? </b></p>**

1. <p><b>adress(a); </b></p>

2. <p><b>\*a; </b></p>

3. <p><b>a; </b></p>

4. <p><b>&a; </b></p>

Ответ:2

Комментарий: AP00156

**ОО**

<p><b>Что будет выведено на экран в результате работы программы: </p></b>

<p><b>int arr[5] = {1,2,3,7,9,5};</p></b>

<p><b>for (int i = 0; i < 5; ++i)std::cout << arr[i] << std::endl; </p></b>

1. **<p><b>123795</p></b>**
2. **<p><b>1 2 3 7 9 5</p></b>**
3. **<p><b>Программа не скомпилируется</p></b>**
4. **<p><b>597321</p></b>**

Ответ:3

Комментарий: AP00157

**<p><b> Какое число будет выведено последним по завершении цикла</p></b>**

**<p><b>int arr[5] = {1,2,3,7,9};</p></b>**

**<p><b> for (int i = 0; i <= 5; ++i)std::cout << \*(arr+i) << " "; </p></b>**

**</p></b>**

1. <p><b>9</p></b>

2. <p><b>0</p></b>

3. <p><b>Компилятор выдаст исключение</p></b>

4. <p><b>Будет выведен “мусор” </p></b>

Ответ:4  
Комментарий: AP00158

**<p><b>Результат операции sizeof(Arr)/sizeof(Arr[0]), где Arr целочисленный массив</p></b>**

1. <p><b>Размер массива в байтах</p></b>

2. <p><b>Размер первого элемента массива в байтах</p></b>

3. <p><b>Количество элементов в массиве</p></b>

4. <p><b>При выполнении данной операции компилятор выдаст ошибку</p></b>

Ответ:3

Комментарий: AP00159

**<p><b>Отметьте все способы передачи массива в функцию</p></b>**

1. <p><b>По значению</p></b>

2. <p><b>По ссылке</p></b>

3. <p><b>По указателю</p></b>

4. <p><b>В языках С и С++ массивы передавать в функции нельзя</p></b>

Ответ:1, 2, 3

Комментарий: AP00160

**<p><b> Что будет выведено на экран</p></b>**

**<p><b>**int arr[5] = { 1, 2, 3, 3 ,5 }; int i = 0;**</p></b>**

**<p><b>** while (i < 4)std::cout << arr[++i];**</p></b>**

1. <p><b>12335</p></b>

2. <p><b>2335</p></b>

3. <p><b>1 2 3 3</p></b>

4. <p><b>2 3 3 5</p></b>

Ответ:2

Комментарий: AP00161

**<p><b>Отметьте неправильные применения операции взятия адреса(value,data - переменные)</p></b>**

1. <p><b>ptr=&34</p></b>

2. <p><b>ptr =&(value + 12)</p></b>

3. <p><b>ptr =&value </p></b>

4. <p><b>ptr =&(value - data)</p></b>

Ответ:1, 2, 4

Комментарий: AP00162

**<p><b> Первое число – вводимое, второе - выводимое, выберите правильные варианты</p></b>**

**<p><b>int value; std::cin >> value; </p></b>**

**<p><b>int\* ptr = &value; </p></b>**

**<p><b>int\*\* dptr = &ptr; </p></b>**

**<p><b>std::cout << sizeof(ptr) \* sizeof(dptr) + value;</p></b>**

1. <p><b>1 9</p></b>

2. <p><b>4 16</p></b>

3. <p><b>4 20</p></b>

4. <p><b>2 18</p></b>

Ответ:3, 4

Комментарий: AP00163

**<p><b>Какие из данных конструкций не допустимы к повторной инициализации</p></b>**

1. <p><b>int\* const ptr = &value;</p></b>

2. <p><b>const int\* const ptr = &value; </p></b>

3. <p><b> int\* ptr = &value;</p></b>

4. <p><b>const int\* ptr = &value;</p></b>

Ответ:1, 2

Комментарий: AP00164

**<p><b>Какие из переменных являются указателем</p></b>**

1. <p><b>double x;</p></b>

2. <p><b>char arr[4]; </p></b>

3. <p><b> int\* ptr;</p></b>

4. <p><b>double\*\* dp;</p></b>

Ответ:2,3,4

Комментарий: AP00166

**<p><b> Что выведет программа</p></b>**

**<p><b>int value = 2;</p></b>**

**<p><b> int& data = value; data = 5; </p></b>**

**<p><b>if (data == value)std::cout << value;</p></b>**

**<p><b> std::cout << data;</p></b>**

1. <p><b>55</p></b>

2. <p><b>5</p></b>

3. <p><b>2</p></b>

4. <p><b>22</p></b>

Ответ:1

Комментарий: AP00167

**<p><b>Какие пункты являются отличием ссылки от указателя</p></b>**

1. <p><b>Не подлежит разыменованию</p></b>

2. <p><b>Хранит адрес переменной</p></b>

3. <p><b>Изменение ссылки влечёт за собой изменение объекта </p></b>

4. <p><b>Адрес нельзя изменит</p></b>

Ответ:1,3

Комментарий: AP00168

**<p><b> Какие характеристики верны для bool (\*FuncPtr)(char, long)?** **</p></b>**

1. <p><b>Это функция</p></b>

2. <p><b>Это указатель на функцию</p></b>

3. <p><b>Принимает 2 параметра </p></b>

4. <p><b>Возвращает bool</p></b>

Ответ:2,3,4

Комментарий: AP00169

**<p><b> Что выведет программа</p></b>**

**<p><b> int arr[5] = { 1,2,2,1,9 };</p></b>**

**<p><b> int\* ptr = arr;</p></b>**

**<p><b>\*ptr = arr[4];</p></b>**

**<p><b> for (int i = 0; i < 5;++i)std::cout << arr[i] << " ";</p></b>**

1. <p><b>1 2 2 1 9</p></b>

2. <p><b>92219</p></b>

3. <p><b>9 2 2 1 9</p></b>

4. <p><b>1 2 2 1 1</p></b>

Ответ:3

Комментарий: AP00170

**<p><b> Объявлен указатель int\* ptr, какую смысловую нагрузку несёт int стоящий перед идентификатором? </p></b>**

1. <p><b>Указывает тип переменной</p></b>

2. <p><b>Показывает что переменная -указатель</p></b>

3. <p><b>Обозначает тип данных на которые ссылается указатель</p></b>

Ответ:3

Комментарий: AP00171

**<p><b>Что делает функция strcmp()?</b></p>**

1. <p><b>Сравнивает содержимое строк и возвращает значение 1, если они равны</b></p>

2. <p><b>Вырезает строку</b></p>

3. <p><b>Определяет длину троки</b></p>

4. <p><b>Сравнивает содержимое строк и возвращает значение 0, если они равны</b></p>

Ответ: 4.

Комментарий: AP00172

**<p><b>Что делает функция strcat()?</b></p>**

1. <p><b>Вырезает строку</b></p>

2. <p><b>Соединяет несколько строк в одну</b></p>

3. <p><b>Присоединяет копию второй строки к первой, образуя одну строку</b></p>

Ответ: 3.

Комментарий: AP00173

**<p><b> Что означает заголовочный файл malloc.h? </b></p>**

1. <p><b>Интерфейс прикладного программирования для мультимедиа</b></p>

2. <p><b>Функции работы с буферами</b></p>

3. <p><b> Функции выделения памяти</b></p>

4. <p><b>Функции работы со строками</b></p>

Ответ: 3.

Комментарий: AP00174.

**<p><b>Для чего используются функции fputs() и fgets()?</b></p>**

1. <p><b>Для форматирования ввода-вывода в текстовые файлы</b></p>

2. <p><b>Для записи и чтения строк из символьного потока</b></p>

3. <p><b> Для форматирования ввода-вывода в бинарные файлы </b></p>

4. <p><b>Для записи и чтения символов из бинарного файла</b></p>

5. <p><b> Для записи и чтения символов из текстового файла</b></p>

Ответ: 2.

Комментарий: AP00175.

**<p><b>Каким будет результат вывода переменной b в консоль? </b></p>**

<p><b>#include <iostream>

int main()

{

    int a = 18;

int \*pa = &a;

int b = \*pa + 15;

    return 0;

}</b></p>

1. <p><b>31</b></p>

2. <p><b> Некорректный вывод</b></p>

3. <p><b>33</b></p>

4. <p><b>32</b></p>

Ответ: 3.

Комментарий: AP00176

**<p><b>Каким будет результат вывода переменных b, c и d в консоль? </b></p>**

<p><b>#include <iostream>

int main()

{

int a = 10;

int\* pa = &a;

int b = ++\*pa;

int c = (\*pa)++;

int d = \*pa++;

return 0;

}</b></p>

1. <p><b> b=10, c=10, d=10</b></p>

2. <p><b> b=11, c=10, d=10</b></p>

3. <p><b> b=11, c=11, d=11</b></p>

4. <p><b> b=10, c=12, d=10</b></p>

5. <p><b> Нет правильного ответа</b></p>

Ответ: 5.

Комментарий: AP00177.

**<p><b>Что необходимо сделать при определении массива символьных строк? </b></p>**

1. <p><b>Создать указатель на массив</b></p>

2. <p><b> Заключить массив в круглые скобки</b></p>

3. <p><b>Инициализировать массив при помощи символьной константы</b></p>

4. <p><b>Нет правильного ответа</b></p>

Ответ: 3.

Комментарий: AP00178.

**<p><b>Какой символ является символом разыменования указателя? </b></p>**

1. <p><b> ^</b></p>

2. <p><b> \*</b></p>

3. <p><b>&</b></p>

4. <p><b>\*\*</b></p>

5. <p><b> $</b></p>

Ответ: 2.

Комментарий: AP00179.

**<p><b>Выход за границу массива – это …</b></p>**

1. <p><b>Ошибка компилятора</b></p>

2. <p><b>частный случай переполнения буфера</b></p>

3. <p><b> частный случай переполнения статического объема памяти</b></p>

Ответ: 2.

Комментарий: AP00180.  
  
**<p><b>Результат выполнения программы</b></p>**

<p><b>int arr = 5; </b></p>

<p><b>int arr2 = 7; </b></p>

<p><b> const int\* a = &arr; </b></p>

<p><b>a = &arr2; </b></p>

<p><b>cout << \*a; </b></p>

1. <p><b>7</b></p>

2. <p><b>Ошибка</b></p>

3. <p><b>5</b></p>

4. <p><b>57</b></p>

Ответ: 1)  
Комментарий: AP00181.

**<p><b>Объявление константного указателя</b></p>**

1) <p><b>const int\* arr = &a; </b></p>

2) <p><b>int\* const arr=&a; </b></p>

3) <p><b>int const\* arr=&a; </b></p>

4) <p><b>const int\* const arr=&a; </b></p>

Ответ: 2)  
Комментарий: AP00182

**<p><b>Объявление константного указателя на константу </b></p>**

1) <p><b>const int\* arr=&a; </b></p>

2) <p><b>const int\* const arr=&a; </b></p>

3) <p><b>int\* const arr=&a; </b></p>

4) <p><b>int\*arr= const & a; </b></p>

Ответ: 2)  
Комментарий: AP00183

**<p><b>Результат выполнения программы</b></p>**

<p><b>const int arr = 5; </b></p>

<p><b> const int\* a = &arr; </b></p>

<p><b>\*a = 9; </b></p>

<p><b>cout << \*a; </b></p>

1. <p><b>7</b></p>

2. <p><b>Ошибка</b></p>

3. <p><b>5</b></p>

4. <p><b>57</b></p>

Ответ: 1)  
Комментарий: AP00184

**<p><b>strcat() – функция предназначенная для</b></p>**

1) <p><b>Разрезания строки на 2 части</b></p>

2) <p><b>Измерения длины строк</b></p>

3) <p><b>Объединения двух строк</b></p>

4) <p><b>Нет правильного ответа</b></p>

Ответ 3)  
Комментарий: AP00185

**<p><b>Конкатенация – </b></p>**

1) <p><b>Разрезание 2 строк</b></p>

2) <p><b>Измерение длины строки</b></p>

3) <p><b>Объединение двух строк</b></p>

4) <p><b>Изменение строки</b></p>

Ответ: 3)  
Комментарий: AP00186  
  
**<p><b>Результат работы программы, представленной ниже, будет: </b></p>**

int iArr[] = {12,23,3,24,5};

char cArr[] = {“world”};

cout << sizeof(iArr) + sizeof(‘a’) + sizeof(cArr);

1. <p><b>11</b></p>

2. <p><b>26</b></p>

3. <p><b>27</b></p>

4. <p><b>42</b></p>

Ответ: 3

Комментарий: AP00187  
  
**<p><b>Какие из следующих объявлений массива содержат ошибки? </b></p>**

1. <p><b> int size = 3; int array[size];</b></p>

2. <p><b> int array[3];</b></p>

3. <p><b> int n; cin >> n; int array[n];</b></p>

4. <p><b> int a = 1, b= 2, c = 3; int array[3] = {a,b,c};</b></p>

Ответ: 1, 3

Комментарий: AP00188

**<p><b>В коде “const int \*ptr;” верными утверждениями являются?</b></p>**

1. <p><b>Нельзя изменить значение ptr </b></p>

2. <p><b>Можно изменить значение ptr </b></p>

3. <p><b>Нельзя изменить значение данных по адресу ptr</b></p>

4. <p><b>Можно изменить значение данных по адресу ptr </b></p>

Ответ: 2, 3

Комментарий: AP00189

**<p><b>В коде “int \*const ptr;” верными утверждениями являются?</b></p>**

1. <p><b>Нельзя изменить значение ptr </b></p>

2. <p><b>Можно изменить значение ptr </b></p>

3. <p><b>Нельзя изменить значение данных по адресу ptr</b></p>

4. <p><b>Можно изменить значение данных по адресу ptr </b></p>

Ответ: 1, 4

Комментарий: AP00190

**<p><b>Выберите правильный ответ, выражение “const int const \*ptr:</b></p>**

1. <p><b>константный указатель</b></p>

2. <p><b>указатель на константное значение</b></p>

3. <p><b>константный указатель на константное значение</b></p>

4. <p><b>выражение содержит ошибку</b></p>

Ответ: 3

Комментарий: AP00191

**<p><b>Передача массива «int a[5]» в функцию является верной: </b></p>**

1. <p><b>func (a[5]); </b></p>

2. <p><b>func(a,5); </b></p>

3. <p><b>func(&a,5)</b></p>

4. <p><b>func(\*a,5)</b></p>

5. <p><b>Все вышеперечисленные</p></b>

Ответ: 2

Комментарий: AP00192

**<p><b>Какой вывод будет в следующем фрагменте кода?</b></p>**

#include<stdio.h>

int main()

{

char huge \*near \*far \*ptr1;

char near \*far \*huge \*ptr2;

char far \*huge \*near \*ptr3;

printf("%d, %d, %dn", sizeof(ptr1), sizeof(\*ptr2), sizeof(\*\*ptr3));

return 0;

}

1. <p><b>4,4,4</b></p>

2. <p><b>2,4,4</b></p>

3. <p><b>4,2,4 </b></p>

4. <p><b2,2,2</b></p>

Ответ: 1

Комментарии: AP00193

**<p><b>Что означает следующий код int \*f(); ? </b></p>**

1. <p><b>тип указателя функция </b></p>

2. <p><b>функция возвращающая тип int </b></p>

3. <p><b>указатель на функцию</b></p>

4. <p><b>обычный указатель на переменную</b></p>

Ответ: 2

Комментарий: AP00194

**<p><b>Что означает следующее выражение, char \* arr [10];? </b></p>**

1. <p><b>указатели на символы</b></p>

2. <p><b>указатели на функции </b></p>

3. <p><b>массив символов </b></p>

4. <p><b>указатель на массив символов</b></p>

Ответ: 1

Комментарий: AP00195

**<p><b>Каким будет вывод программы? </b></p>**

char str[20] = "Hello";

char \*const p=str;

\*p = 'b';

\*p='m';

printf("%s\n", str);

1. <p><b>mello</b></p>

2. <p><b>bello </b></p>

3. <p><b>hello </b></p>

4. <p><b>ни один из них </b></p>

Ответ: 1

Комментарий: AP00196

## 15 тема

<p><b>Обязательно ли должна быть функция main() в пректе?</b></p>

1)<p><b> да </b></p>

2)<p><b> нет </b></p>

Ответ: 1

Комментарий: AP0015001

<p><b>Каким образом из перечисленных можно вызвать функцию так, чтобы она работала?</b></p>

1)<p><b> function\_name();</b></p>

2)<p><b> function\_name() </b></p>

3)<p><b> function\_name; </b></p>

Ответ: 1

Комментарий: AP0015002

<p><b>Когда нужно прописывать прототипы функций?</b></p>

1)<p><b> если все функции прописаны до главной функции main()</b></p>

2)<p><b> если все функции прописаны после главной функции main()</b></p>

3)<p><b> в любом случае </b></p>

Ответ: 1

Комментарий: AP0015003

<p><b>Как определить какого типа будет функция?</b></p>

1)<p><b> по оператору return </b></p>

2)<p><b> по тому, с чем будет работать функция</b></p>

3)<p><b> по типу главной функции main()</b></p>

Ответ: 1

Комментарий: AP0015004

<p><b> Что такое рекурсия?</b></p>

1)<p><b> определение функции или ее части через саму себя </b></p>

2)<p><b> метод для вызова нескольких функций одновременно </b></p>

3)<p><b> метод вызова других функций , с помощью определения функции (метода) через саму себя </b></p>

Ответ: 1

Комментарий: AP0015005

<p><b>Какого типа эта функция?

<p>… password\_is\_valid (string password)</p>

<p>{ string valid\_pass = "qwerty123"; </p>

<p>if (valid\_pass == password) return true;</p>

<p>else return false; }</p>

</b></p>

1)<p><b> bool </b></p>

2)<p><b> int </b></p>

3)<p><b> void </b></p>

Ответ: 1

Комментарий: AP0015006

<p><b>Какого типа эта функция?

<p>… get\_pass () </p>

<p>{ string user\_pass; </p>

<p> cout << "Введите пароль: "; </p>

<p> getline(cin, user\_pass);</p>

<p> if (!password\_is\_valid(user\_pass))</p>

<p>{ cout << "Неверный пароль!" << endl; </p>

<p> get\_pass (); }</p>

<p> else { cout << "Доступ разрешен." << endl; } }</p>

</b></p>

1)<p><b> bool </b></p>

2)<p><b> int </b></p>

3)<p><b> void </b></p>

Ответ: 3

Комментарий: AP0015007

<p><b>Используется ли в этой функции рекурсия?

<p>int factr(int n) </p>  
<p>{int answer;</p>  
<p>if(n==1) return(1);</p>  
<p>answer = factr(n-1)\*n;</p>  
<p>return(answer);}</p>

</b></p>

1)<p><b> Да</b></p>

2)<p><b> Нет</b></p>

Ответ: 1

Комментарий: AP0015008

<p><b>Используется ли в этой функции рекурсия?

<p>int factr(int n) </p>  
<p>int t, answer; </p>  
<p>answer == 1; </p>  
<p>for(t=1; t<=n; t++) </p>  
<p>answer=answer\*(t); </p>  
<p>return(answer); </p>

</b></p>

1)<p><b> Да</b></p>

2)<p><b> Нет</b></p>

Ответ: 2

Комментарий: AP0015009

<p><b>Что можно использовать вместо рекурсии?</b></p>

1)<p><b> цикл for </b></p>

2)<p><b> цикл do…while</b></p>

3)<p><b> цикл while </b></p>

Ответ: 2

Комментарий: AP0015010

<p><b>Что такое функция?</b></p>

1)<p><b> это самостоятельная единица программы, которая спроектирована для реализации конкретной подзадачи</b></p>

2)<p><b> любой алгоритм, написанный на яп </b></p>

3)<p><b> набор команд/операторов для выполнения определенных задач </b></p>

Ответ: 1, 3

Комментарий: AP0015011

<p><b>Дайте определение сигнатруы функции</b></p>

1)<p><b> параметр функции определяющий правила использования функции </b></p>

2)<p><b> способ реализации функции</b></p>

3)<p><b> параметр функции представляющий собой тело функции.</b></p>

Ответ: 1

Комментарий: AP0015012

<p><b>Что собой представляет Семантика функции?</b></p>

1)<p><b> описание функции, включающее имя функции, перечень формальных параметров с их типами и тип возвращаемого значения</b></p>

2)<p><b> правила использования функции</b></p>

3)<p><b> тело функции</b></p>

Ответ: 3

Комментарий: AP0015013

<p><b> Каждая функция в языке Си должна быть определена, то есть должны быть указаны: </b></p>

1)<p><b> тип возвращаемого значения; имя функции; информация о формальных аргументах; тело функции </b></p>

2)<p><b> тип возвращаемого значения; имя функции; тело функции </b></p>

3)<p><b> имя функции; информация о формальных аргументах; тело функции </b></p>

Ответ: 1

Комментарий: AP0015014

<p><b>какому из перечисленных вариантов подходит продолжение к следующему определению: Системные функции …</b></p>

1)<p><b>которые созданы для реализации одного конкретного куска программы </b></p>

2)<p><b> хранятся в стандартных библиотеках, и пользователю не нужно вдаваться в подробности их реализации. Достаточно знать лишь их сигнатуру </b></p>

2)<p><b> — это функции, написанные пользователем для решения конкретной подзадачи. </b></p>

Ответ: 2

Комментарий: AP0015015

<p><b>какому из перечисленных вариантов подходит продолжение к следующему определению: Собственные функции …</b></p>

1)<p><b> хранятся в стандартных библиотеках, и пользователю не нужно вдаваться в подробности их реализации. </b></p>

2)<p><b> — это функции, написанные пользователем для решения конкретной подзадачи. </b></p>

3)<p><b> которые созданы для реализации одного конкретного куска программы </b></p>

Ответ: 2

Комментарий: AP0015016

<p><b> какому из перечисленных вариантов подходит продолжение к следующему определению: Фактический аргумент …</b></p>

1)<p><b> — это величина, которая присваивается формальному аргументу при вызове функции </b></p>

2)<p><b> это переменная в вызываемой функции</b></p>

3)<p><b> все выше перечисленное b></p>

Ответ: 1

Комментарий: AP0015017

<p><b>Что будет результатом выполнения программы?

<br>

<br>#include &lt;iostream&gt;<br>

template &lt;typename T, typename T1&gt;<br>

T max(T left, T1 right) {<br>

return (left > right) ? left : right;<br>

}<br><br>

int main() {<br>

std::cout &lt;&lt; max(5, 6.1);<br>

}

</b></p>

1)<p><b> 5 </b></p>

2)<p><b> 5.1 </b></p>

3)<p><b> 6 </b></p>

4)<p><b> 5.1 </b></p>

Ответ: 3

Комментарий: AP00015018

<p><b>В какой библиотеке определен набор макросов для работы с функциями, принимающими неопределенное количество аргументов?</p></b>

1)<p><b> stdvar.h </b></p>

2)<p><b> stdargs.h </b></p>

3)<p><b> stdlib.h </b></p>

4)<p><b> unistd.h </b></p>

Ответ: 2

Комментарий: AP00015019

<p><b>Что будет результатом выполнения программы?<br><br>

#include &lt;iostream&gt;<br>

template &lt;typename T&gt;<br>

T max(T left, T right) {<br>

return (left > right) ? left : right;<br>

}<br><br>

int main() {<br>

std::cout &lt;&lt; max(5, 6.1);<br>

}</b></p>

1)<p><b> 5 </b></p>

2)<p><b> 6.1 </b></p>

3)<p><b> программа не скомпилируется </b></p>

4)<p><b> 6 </b></p>

Ответ: 3

Комментарий: AP00015020

<p><b> Что будет результатом программы? <br><br>

#include &lt;iostream&gt;<br>

template &lt;typename T, class T1, typename T2&gt;<br>

T2 max(T left, T1 right) {<br>

return (left &lt; right) ? left : right;<br>

}<br><br>

int main() {<br>

std::cout &lt;&lt; max&lt;double, int, float&gt;(6.1, 3.6);<br>

}</b></p>

1)<p><b> программа не скомпилируется </b></p>

2)<p><b> 3 </b></p>

3)<p><b> 3.6 </b></p>

4)<p><b> 6.1 </b></p>

5)<p><b> 6 </b></p>

Ответ: 2

Комментарий: AP00015021

<p><b>Значения углов в тригонометрических функциях библиотеки math.h задаются в</b></p>

1)<p><b> радианах</b></p>

2)<p><b> градусах</b></p>

3)<p><b> минутах</b></p>

4)<p><b> секундах </b></p>

Ответ: 1

Комментарий: AP00015022

<p><b>Из имеющихся прототипов функции all() выберите неверный<br><br>

int all(char a, float b, double c, double d) {<br>

return 42;<br>

}<br>

</p></b>

1)<p><b> int all(char d, float c, double b, double a); </p></b>

2)<p><b> inline int all(char a, float b, double c, double); </p></b>

3)<p><b> int all(char, float, double, double); </p></b>

4)<p><b> int all(char a, float b, double c, d);</p></b>

Ответ: 4

Комментарий: AP00015023

<p><b>Cамым первым (под индексом 0) в массиве argv аргументов командной строки всегда будет</p></b>

1)<p><b> время запуска</p></b>

2)<p><b> имя программы</p></b>

3)<p><b> количество аргументов</p></b>

4)<p><b> там может быть все что угодно</p></b>

Ответ: 2

Комментарий: AP00015024

<p><b>Как объявить указатель на функцию foo()?<br><br>

char foo(int a, int\* b, char c) { return 'a'; }</b></p>

1)<p><b> char (имя\_указателя)(int, int\*, int); </p></b>

2)<p><b> char \*имя\_указателя (int, int\*, char); </p></b>

3)<p><b> char (имя\_указателя)(int, int\*, char); </p></b>

4)<p><b> char (\* имя\_указателя)(int, int\*, char); </p></b>

Ответ: 4

Комментарий: AP00015025

<p><b>Какая функция из math.h округляет в большую сторону?</b></p>

1)<p><b> ceil</b></p>

2)<p><b> round</b></p>

3)<p><b> mod</b></p>

4)<p><b> floor</b></p>

Ответ: 1

Комментарий: AP00015026

<p><b>Какой из перечисленных операторов существует в языке еще со времен C?</b></p>

1)<p><b> new</b></p>

2)<p><b> ::</b></p>

3)<p><b> ,</b></p>

4)<p><b> static\_cast</b></p>

Ответ: 3

Комментарий: AP00015027

<p><b>Где в прототипе функции можно размещать аргументы по умолчанию?</p></b>

1)<p><b> в начале</p></b>

2)<p><b> в середине</p></b>

3)<p><b> в конце</p></b>

4)<p><b> где угодно</p></b>

Ответ: 3

Комментарий: AP00015028

<p><b>Что из перечисленных свойств функции main() правда?</p></b>

1)<p><b> она является точкой входа программы</p></b>

2)<p><b> она не может вызывать саму себя</p></b>

3)<p><b> ее можно вызывать из других функций</p></b>

4)<p><b> она принимает аргументы из командной строки</p></b>

Ответ: 1,3,4

Комментарий: AP00015029

<p><b>Какой класс памяти обладает наивысшим приоритетом?</b></p>

1)<p><b>auto</b></p>

2)<p><b>register</b></p>

3)<p><b>extern</b></p>

4)<p><b>static</b></p>

Ответ: 1

Комментарий: AP00015030

<p><b>Какой класс памяти обеспечивает наибольшее быстродействие?</b></p>

1)<p><b>auto</b></p>

2)<p><b>register</b></p>

3)<p><b>extern</b></p>

4)<p><b>static</b></p>

Ответ: 2

Комментарий: AP00015031

<p><b>По чему нельзя перегрузить функцию?</b></p>

1)<p><b>числу аргументов</p></b>

2)<p><b>типам аргументов</p></b>

3)<p><b>возвращаемому значению</p></b>

4)<p><b>наличию эллипсиса</p></b>

Ответ: 3  
Комментарий: AP00015032

<p><b>Присвоение какого класса памяти компилятор считает необязательным и осуществляет его только  
при наличии у компьютера соответствующих ресурсов</b></p>

1)<p><b>auto</b></p>

2)<p><b>static</b></p>

3)<p><b>extern</b></p>

4)<p><b>register</b></p>  
Ответ: 4

Комментарий: AP00015033

<p><b>Что означает ключевое слово mutable у переменной?</p></b>

1)<p><b>что значение переменной может меняться извне</b></p>

2)<p><b>что она должна храниться в регистре процессора</b></p>

3)<p><b>что она может меняться в константом контексте</b></p>

4)<p><b>что она не удаляется после выхода из области видимости</b></p>

Ответ: 3

Комментарий: AP00015034

<p><b> Из чего состоит прототип функции? </b> </p>

1)<p><b> состоит из типа возврата функции, её имени и параметров</b></p>

2)<p><b> состоит её имени и параметров</b></p>

Ответ:1

Комментарий: AP00015034

<p><b> Если функция не возвращает никакого значения, то она должна иметь тип? </b> </p>

1)<p><b> Char </b></p>

2)<p><b> Void</b></p>

3)<p><b> Int </b></p>

Ответ:2

Комментарий: AP00015035

<p><b> Согласно какому стандарту все функции должны иметь прототипы? </b> </p>

1)<p><b> ANSI C </b></p>

2)<p><b> ASCII</b></p>

3)<p><b> ISO </b></p>

Ответ:1

Комментарий: AP00015036

<p><b> Как называется вспомогательный алгоритм, который напрямую или через другие вспомогательные алгоритмы вызывает сам себя? </b> </p>

1)<p><b> Линейный</b> </p>

2)<p><b> Разветвляющийся</b> </p>

3)<p><b> Циклический </b> </p>

4)<p><b> Рекурсивный </b> </p>

Ответ:4

Комментарий: AP00015037

<p><b> Функции какого заголовочного файла помогают получить и возвратить значения типа double ? </b> </p>

1)<p><b> iostream </b> </p>

2)<p><b> string.h </b> </p>

3)<p><b> math.h </b> </p>

Ответ:3

Комментарий: AP00015038

<p><b> Тип функции определяется типом её аргументов? </b> </p>

1)<p><b>Да </b> </p>

2)<p><b> Нет </b> </p>

Ответ:2

Комментарий: AP00015039

<p><b> Какое ключевое слово, помогает встраивать функции? </b> </p>

1)<p><b> using </b> </p>

2)<p><b> union </b> </p>

3)<p><b> virtual </b> </p>

4)<p><b> inline </b> </p>

Ответ:4

Комментарий: AP00015040

<p><b> Когда мы используем Многоточие (…)? </b> </p>

1)<p><b> Когда знаем количество аргументов </b> </p>

2)<p><b> Когда аргументов много </b> </p>

3)<p><b> Когда не знаем количество аргументов </b> </p>

Ответ:3

Комментарий: AP00015041

<p><b> Правила области действия констатируют, что у переменной, имеющей как локальную, так и файловую область действия , используется ее глобальное значение? </b> </p>

1)<p><b> Да</b> </p>

2)<p><b> Нет </b> </p>

Ответ:2

Комментарий: AP00015042

<p><b> Шаблон функции – это … </b> </p>

1)<p><b> определение функции, в которой типу обрабатываемых данных присвоено условное обозначение</b> </p>

2)<p><b> прототип функции, в котором вместо имен параметров указан условный тип </b> </p>

3)<p><b> определение функции, в котором указаны возможные варианты типов обрабатываемых параметров</b> </p>

4)<p><b> определение функции, в котором в прототипе указан условный тип, а в определении указаны варианты типов обрабатываемых параметров</b> </p>

Ответ:1

Комментарий: AP00015043

<p><b> Сколько аргументов можно передавать в функцию? </b> </p>

1)<p><b> до 10 </b> </p>

2)<p><b> до 5</b> </p>

3)<p><b> 1 </b> </p>

4)<p><b> Неограниченное количество </b> </p>

Ответ:4

Комментарий: AP00015043

<p><b> Основные особенности объективно-ориентированного языка (выберите наиболее точный ответ)

</b> </p>

1)<p><b> множественное наследование и полиморфизм</b> </p>

2)<p><b> абстрактные типы данных и сохранение состояний</b> </p>

3)<p><b> перегрузка операций и обработка исключений. </b> </p>

4)<p><b> инкапсуляция, наследование, полиморфизм </b> </p>

Ответ:4

Комментарий: AP00015044

<p><b> Используется ли scope resolution operator:: для уточнения области видимости?</b> </p>

1)<p><b> нет </b> </p>

2)<p><b> да</b> </p>

Ответ:2

Комментарий: AP00015045

<p><b> Что такое аргумент по умолчанию? </b> </p>

1)<p><b> аргумент функции, который программист может не указывать при вызове функции. Аргумент по умолчанию добавляется компилятором автоматически </b> </p>

2)<p><b> аргумент функции, который программист указывает при вызове функции </b> </p>

Ответ:1

Комментарий: AP00015046

<p><b> Аргументы по умолчанию объявляются в … </b> </p>

1)<p><b> в самой функции </b> </p>

2)<p><b> никакой вариант не корректен </b> </p>

3)<p><b> в прототипе функции </b> </p>

Ответ:1

Комментарий: AP00015047

<p><b> В какой библиотеке находятся функции округления </b> </p>

1)<p><b> <cmath> </b> </p>

2)<p><b> <string.h> </b> </p>

3)<p><b> <algorithm> </b> </p>

Ответ:1

Комментарий: AP00015048

<p><b> какой вариант ,написания шаблона , является корректным </b> </p>

1)<p><b> <class T> </b> </p>

2)<p><b> template typename T </b> </p>

3)<p><b> нет верных вариантов ответа </b> </p>

Ответ:3

Комментарий: AP00015049

<p><b> какого типа будет функция, если она возвращает целочисленный тип даннх?</b> </p>

1)<p><b> void </b> </p>

2)<p><b> double </b> </p>

3)<p><b> int </b> </p>

Ответ:3

Комментарий: AP00015050

## 16 тема

**<p><b>Что выведется в консоль при запуске программы?**

**#include "stdio.h"**

**#define func(a, b) a ## b**

**int main()**

**{**

**int x = 5;**

**int y = 1;**

**int xy = 10;**

**printf("%d", func(x, y));**

**return 0;**

**}</b></p>**

1. <p><b>5</b></p>

2. <p><b>1</b></p>

3. <p><b>10</b></p>

4. <p><b>Ошибка</b></p>

Ответ: 3

Комментарий: AP016001

**<p><b>Что выведется в консоль при запуске программы?**

**#include "stdio.h"**

**#define func(s) # s**

**int main()**

**{**

**printf(func(16 12 s));**

**return 0;**

**}</b></p>**

1. <p><b>s</b></p>

2. <p><b>16 12 s</b></p>

3. <p><b>16</b></p>

4. <p><b>12</b></p>

Ответ: 2

Комментарий: AP016002

**<p><b>Что выведется в консоль при запуске программы?**

**#include <iostream>**

**#define func(value) #@value**

**int main()**

**{**

**std::cout << func(d, 17);**

**return 0;**

**}</b></p>**

1. <p><b>17</b></p>

2. <p><b>Ошибка</b></p>

3. <p><b>d, 17</b></p>

4. <p><b>d</b></p>

Ответ: 2

Комментарий: AP016003

**<p><b>Значение какой переменной равно 4?**

**#include "stdio.h"**

**#define value(i) val ## i**

**int main()**

**{**

**int value(3) = 4;**

**return 0;**

**}</b></p>**

1. <p><b>val3</b></p>

2. <p><b>value3</b></p>

3. <p><b>value4</b></p>

4. <p><b>val4</b></p>

Ответ: 1

Комментарий: AP016004

**<p><b>Что выведется в консоль при запуске программы?**

**#include <iostream>**

**#define MACRO1 printf("MACRO1 invoked.")**

**#define MACRO2 printf("MACRO2 invoked.")**

**#define MAKE\_MACRO(n) MACRO ## n**

**int main(){**

**MAKE\_MACRO(1);**

**return 0;**

**}</b></p>**

1. <p><b>MACRO2 invoked.</b></p>

2. <p><b>Ошибка</b></p>

3. <p><b>MACRO1</b></p>

4. <p><b>MACRO1 invoked.</b></p>

Ответ: 4

Комментарий: AP016005

**<p><b>Чему соответствует данное сочетание символов в препроцессоре С++: #@?</b></p>**

1. <p><b>Операция конкатенации</p></b>

2. <p><b>Операция подстановки строки</b></p>

3. <p><b>Операция переопределения</b></p>

4. <p><b>Операция подстановки символа</b></p>

Ответ: 4

Комментарий: AP016006

**<p><b>Для чего используется символ "#" в препроцессоре С++?</b></p>**

1.<p><b> Для подстановки строки</p></b>

2. <p><b>Для конкатенации строк</b></p>

3. <p><b>Для подстановки символа</b></p>

4. <p><b>Для подстановки численного значения</b></p>

Ответ: 1

Комментарий: AP016007

**<p><b>Размер указателя на функцию можно изменить с помощью модификаторов?</b></p>**

1. <p><b>near</b></p>

2. <p><b>far</b></p>

3. <p><b>interrupt</b></p>

4. <p><b>huge</b></p>

Ответ: 1, 2

Комментарий: AP016008

**<p><b>Модификатор функции … указывает компилятору на то, что параметры вызываемой функции должны помещаться в стек в порядке, обратном следованию при вызове, то есть справа налево </b></p>**

1. <p><b>cdecl</b></p>

2. <p><b>huge</b></p>

3. <p><b>pascal</b></p>

4. <p><b>interrupt</b></p>

Ответ: 1

Комментарий: AP016009

**<p><b>Модификатор функции … указывает компилятору на то, что параметры вызываемой функции должны помещаться в стек слева направо</b></p>**

1. <p><b>cdecl</b></p>

2. <p><b>huge</b></p>

3. <p><b>pascal</b></p>

4. <p><b>interrupt</b></p>

Ответ: 3

Комментарий: AP016010

**<p><b>Ускорить вызов функции можно путем передачи параметров функции через регистры общего назначения, указав перед именем функции какой спецификатор?</b></p>**

1. <p><b>stdcall</b></p>

2. <p><b>cdecl</b></p>

3. <p><b>asm</b></p>

4. <p><b>fastcall</b></p>

Ответ: 4

Комментарий: AP016011

**<p><b>Набор структур, каждая из которых имеет некоторый элемент, или указатель, ссылающийся на другую структуру в этом списке это?</b></p>**

1. <p><b>Двусвязный список</b></p>

2. <p><b>Связный список</b></p>

3. <p><b>Односвязный список</b></p>

4. <p><b>Кольцевой связный список</b></p>

Ответ: 2

Комментарий: AP016012

**<p><b>Какое действие выполняет кусок программы?**

**firstboat=(struct stboat \*) new (struct stboat);</b></p>**

1. <p><b>Увеличивает размер структуры</b></p>

2. <p><b>Уменьшает размер структуры</b></p>

3. <p><b>Выделяет память для структуры</b></p>

4. <p><b>Удаляет указанную структуру</b></p>

Ответ: 3

Комментарий: AP016013

**<p><b>Какие параметры имеет функция реrrоr()?</p></b>**

1. <p><b>Строку символов</p></b>

2. <p><b>Строку символов и указатель на функцию</p></b>

Ответ: 1

Комментарий: AP016014

**<p><b>Сколько стандартных моделей памяти поддерживает компилятор Visual C/C++?</p></b>**

1. <p><b>3</p></b>

2. <p><b>7</p></b>

3. <p><b>6</p></b>

4. <p><b>5</p></b>

Ответ: 3

Комментарий: AP016015

**<p><b>При использовании какой модели памяти создаются программные файлы с расширением .СОМ?</p></b>**

1. <p><b>huge</p></b>

2. <p><b>tiny</p></b>

3. <p><b>large</p></b>

4. <p><b>small</p></b>

Ответ: 2

Комментарий: AP016016

**<p><b>При использовании какой модели памяти создаются программные файлы с расширением .EXE?</p></b>**

1. <p><b>huge</p></b>

2. <p><b>tiny</p></b>

3. <p><b>medium</p></b>

4. <p><b>small</p></b>

Ответ: 4

Комментарий: AP016017

**<p><b>С помощью каких спецификаторов можно определять переменные, если изначально известно, что размер приложения будет выходить за рамки отведенных 64 Кбайт?</p></b>**

1. <p><b>far и huge</p></b>

2. <p><b>huge и medium</p></b>

3. <p><b>medium и small</p></b>

4. <p><b>small и far</p></b>

Ответ: 1

Комментарий: AP016018

**<p><b>В чем главное отличие моделей памяти huge и large?</p></b>**

1. <p><b>При использовании large размер отдельного объекта данных должен превышать 64 Кб</p></b>

2. <p><b>При использовании модели huge снимаются ограничения на размер отдельных объектов данных</p></b>

3. <p><b>Никаких отличий нет</p></b>

Ответ: 2

Комментарий: AP016019

**<p><b>Для генерирования каких вызовов используется модификатор near?</p></b>**

1. <p><b>Для генерирования компилятором ближних вызовов</p></b>

2. <p><b>Для генерирования дальних вызовов</p></b>

3. <p><b>Модификатор не используется для генерирования вызовов</p></b>

Ответ: 1

Комментарий: AP016020

**<p><b>Использование оператора switch сравнимо с использованием:</p></b>**

1. <p><b>for( ; ; )</p></b>

2. <p><b>if-else-if</p></b>

3. <p><b>enum</p></b>

4. <p><b>if-else</p></b>

Ответ: 2

Комментарий: AP016021

**<p><b>Ключевым словом лейбла по умолчанию является:</p></b>**

1. <p><b>case</p></b>

2. <p><b>standard</p></b>

3. <p><b>default</p></b>

4. <p><b>const</p></b>

Ответ: 3

Комментарий: AP016022

**<p><b>Оператор switch - это:</p></b>**

1. <p><b>Условный оператор управления потоком выполнения программы, обеспечивающий множественный выбор</p></b>

2. <p><b>Условный тернарный оператор выбора (ветвления)</p></b>

3. <p><b>Оператор выхода из цикла</p></b>

4. <p><b>Оператор управления потоком выполнения программы, который заставляет центральный процессор выполнить переход из одного участка кода в другой, который идентифицируется с помощью лейбла</p></b>

Ответ: 1

Комментарий: AP016023

**<p><b>Оператор, позволяющий завершить выполнение оператора switch:</p></b>**

1. <p><b>goto</p></b>

2. <p><b>if-else</p></b>

3. <p><b>cout</p></b>

4. <p><b>break</p></b>

Ответ: 4

Комментарий: AP016024

**<p><b>Что будет выведено на консоль после выполнения данного кусочка программы:**

**switch(2)**

**{**

**case 1: std::cout << 1 << ‘\n’;**

**case 2: std::cout << 2 << ‘\n’;**

**case 3: std::cout << 3 << ‘\n’;**

**default: std::cout << 4 << ‘\n’;**

**}**

**</p></b>**

1. <p><b>2</p></b>

2. <p><b>234</p></b>

3. <p><b>

2

3

4

</p></b>

4. <p><b>

2

3

</p></b>

Ответ: 3

Комментарий: AP016025

**<p><b>Что будет выведено на консоль после выполнения данного кусочка программы:**

**switch(1+3)**

**case 1: std::cout << 1 << ‘\n’; break;**

**case 2: std::cout << 2 << ‘\n’; break;**

**case 3: std::cout << 3 << ‘\n’; break;**

**default: std::cout << 4 << ‘\n’; break;**

**</p></b>**

1. <p><b>1</p></b>

2. <p><b>3</p></b>

3. <p><b>4</p></b>

4. <p><b>Программа не запустится</p></b>

Ответ: 4

Комментарий: AP016026

**<p><b>Что будет выведено на консоль после выполнения данного кусочка программы:**

**switch(2 + 2 \* 2)**

**{**

**case 8: std::cout << 8; break;**

**case 6: std::cout << 6;**

**case 4: std::cout << 4; break;**

**default: std::cout << 2;**

**}**

**</p></b>**

1. <p><b>8</p></b>

2. <p><b>86</p></b>

3. <p><b>4</p></b>

4. <p><b>2</p></b>

5. <p><b>6</p></b>

6. <p><b>64</p></b>

Ответ: 6

Комментарий: AP016027

**<p><b>В каком из примеров демонстрируется идентичный тип переменных:**

**Пример 1:**

**int ivalue1;**

**const int ivalue2;**

**Пример 2:**

**int sivalue1;**

**unsigned int sivalue2;**

**Пример 3:**

**string peter1;**

**char Peter1;**

**</b></p>**

1.<p><b> Пример 1 </b></p>

2.<p><b> Пример 2</b></p>

3.<p><b> Пример 3</b></p>

Ответ: 2

Комментарий: AP016028

**<p><b>Совместимы ли интегральные и перечисляемые типы в языке С ?</b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 1

Комментарий: AP016029

**<p><b>Будут ли массивы совместимыми, если для них указаны разные размеры? </b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 2

Комментарий: AP016030

**<p><b>Будет ли ссылка, указывающая на спецификатор типа, который является структурой или объединением, принадлежать к тому же типу? </b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 1

Комментарий: AP016031

**<p><b>В каком из примеров демонстрируется совместимый тип переменных:**

**Пример 1:**

**struct {float fvalue1, fvalue2;} stanonymous1;**

**struct sttwofloats {float fvalue1, fvalue2;} stanonymous;**

**Пример 2:**

**struct {float fvalue1, fvalue2;} stanonymous1;**

**struct {float fvalue3, fvalue2;} stanonymous2;**

**Пример 3:**

**struct sttwofloats {float fvalue1, fvalue2;} stfloat1;**

**struct sttwofloats stfloat2;**

**</b></p>**

1.<p><b> Пример 1 </b></p>

2.<p><b> Пример 2</b></p>

3.<p><b> Пример 3</b></p>

Ответ: 3

Комментарий: AP016032

**<p><b>Выберите верное утверждение</b></p>**

1.<p><b>Два разных типа указателей считаются совместимыми, если они оба указывают на несовместимые типы</b></p>

2.<p><b>Любой тип, которому предшествует какой-либо модификатор доступа, порождает совместимые типы</b></p>

3.<p><b>В C++ перечисляемые типы с интегральными совместимы</b></p>

4.<p><b>Для перечисляемых типов не требуется, чтобы константы перечисления объявлялись в том же порядке, хотя каждая константа должна иметь то же самое перечисляемое значение</b></p>

Ответ: 4

Комментарий: AP016033

**<p><b>Совместимы ли типы у данных объявлений ?**

**int ben;**

**const int Ben;**

**</b></p>**

1.<p><b>Да</b></p>

2.<p><b>Нет</b></p>

Ответ: 2

Комментарий: AP016034

**<p><b> Выберите верное утверждение для объявления:**

**char \*pc1, \* рс2;**

**</b></p>**

1.<p><b>Переменные pc1 и рс2 являются совместимыми указателями на символы, поскольку при объявлении дополнительный пробел между символом \* и именем рс2 является ненужным (незначащим)</b></p>

2.<p><b> Переменные pc1 и рс2 являются несовместимыми указателями на символы, поскольку при объявлении дополнительный пробел между символом \* и именем рс2 является ненужным (незначащим)</b></p>

Ответ: 1

Комментарий: AP016035

**<p><b>Что проверяет оператор defined?</b></p>**

1. <p><b>Был ли перед этим определен идентификатор или макрос с указанным в этом операторе именем</b></p>

2. <p><b>Проверяет некоторое логическое выражение и в зависимости от результатов проверки выполняет то или иное действие</b></p>

3. <p><b>Проверяет, равны ли значения двух операндов или нет, если да, то условие становится истинным</b></p>

Ответ: 1

Комментарий: AP016036

**<p><b>Для чего применяется операция логического отрицания (!)? </b></p>**

1. <p><b>Для проверки того, что идентификатор или макрос не определен</b></p>

2. <p><b>Для увеличения целочисленного значения на единицу</b></p>

3. <p><b>Для освобождения памяти, занятой массивом</b></p>

4. <p><b>Для выделения динамической памяти</b></p>

Ответ: 1

Комментарий: AP016037

**<p><b>Где располагаются команды препроцессора для более эффективной компиляции? </b></p>**

1. <p><b>В библиотеке</b></p>

2. <p><b>Только в заголовочных файлах</b></p>

3. <p><b>В исходном коде</b></p>

4. <p><b>В среде разработки оконных приложений</b></p>

Ответ: 3

Комментарий: AP016038

**<p><b>Какое отличие в следующем фрагменте программы:**

**Пример 1:**

**if(DEBUG\_ON) {**

**printf("Entering Example Function");**

**printf("First argument passed has a value of %d",ifirst\_arg);**

**}**

**Пример 2:**

**#if defined(DEBUG\_ON)**

**printf("Entering Example Function");**

**printf("First argument passed has a value of %d",ifirst\_arg);**

**#endif</b></p>**

1. <p><b>Размер исполняемой программы</b></p>

2. <p><b>Компилятор не видит операторы, заключенные между командами #if...#endif</b></p>

3. <p><b>Не выполнена условная трансляция этих избыточных операторов</b></p>

Ответ: 1

Комментарий: AP016039

**<p><b>Верно ли утверждение: компиляция — сборка программы, включающая трансляцию всех модулей программы, написанных на одном или нескольких исходных языках программирования высокого уровня или языке ассемблера, в эквивалентные программные модули на низкоуровневом языке, близком машинному коду или непосредственно на машинном языке или ином двоичнокодовом низкоуровневом командном языке и последующую сборку исполняемой машинной программы?</b></p>**

1. <p><b>Нет</b></p>

2. <p><b>Да</b></p>

Ответ: 2

Комментарий: AP016040

**<p><b>Что такое условная компиляция?</b></p>**

1. <p><b> Это компиляция, в которой имеется несколько директив, которые дают возможность выборочно компилировать части исходного кода вашей программы</b></p>

2. <p><b> Это компиляция, которая осуществляется над несколькими исходными модулями в одном пункте задания</b></p>

3. <p><b> Это пошаговая независимая компиляция каждого последующего оператора</b></p>

Ответ: 1

Комментарий: AP016041

**<p><b>Что будет выведено на консоль после выполнения данной части программы?**

**#include <stdio.h>**

**#define TED 10**

**int main(void)**

**{**

**#ifdef TED**

**printf("Привет, Тэд\n");**

**#else**

**printf("Привет, кто-нибудь\n");**

**#endif**

**#ifndef RALPH**

**printf("А RALPH не определен, т.к. Ральфу не повезло.\n");**

**#endif**

**return 0;**

**}**

**</b></p>**

1. <p><b>Привет, кто-нибудь.</b></p>

2. <p><b>Привет, Тед, а также A RALPH не определен, т.к. Ральфу не повезло.</b></p>

3. <p><b>Привет, кто-нибудь, а также A RALPH не определен, т.к. Ральфу не повезло </b></p>

Ответ: 2

Комментарий: AP016042

**<p><b>Выберите правильные утверждения:</b></p>**

1. <p><b>Компилятор выполняет проверку типов в макросе</b></p>

2. <p><b>Макросы могут иметь параметры, как и функции</b></p>

3. <p><b>Макрос не может раскрываться много раз</b></p>

4. <p><b>Макрос раскрывается во время работы препроцессора</b></p>

5. <p><b>Преимущество макроса заключается в скорости выполнения</b></p>

Ответ: 2, 4, 5

Комментарий: AP016043

**<p><b>Какой знак нужно поставить в конце текущей строки для продолжения макроопределения в следующей строке?</b></p>**

1. <p><b>/</b></p>

2. <p><b>\</b></p>

3. <p><b>|</b></p>

Ответ: 2

Комментарий: AP016044

**<p><b>Что означает специальный макрос \_\_STDC\_\_?</b></p>**

1. <p><b>Номер текущей строки обрабатываемого файла</b></p>

2. <p><b>Имя текущего обрабатываемого файла</b></p>

3. <p><b>Программа была откомпилирована с использованием стандарта ANSI С со включенной проверкой на совместимость</b></p>

4. <p><b>Время начала обработки текущего файла в формате hh.mm.ss</b></p>

Ответ: 3

Комментарий: AP016045

**<p><b>Какой макрос используется для отображения времени начала обработки текущего файла?</b></p>**

1. <p><b>\_\_STDC\_\_</b></p>

2. <p><b>\_\_TIMESTAMP\_\_</b></p>

3. <p><b>\_\_DATE\_\_</b></p>

4. <p><b>\_\_TIME\_\_</b></p>

5. <p><b>\_\_FILE\_\_</b></p>

Ответ: 4

Комментарий: AP016046

**<p><b>Какое число будет выведено в консоль в данном примере?**

**#include<stdio.h>**

**#define SQUARE(x) x\*x**

**#define PR(x) printf("x = %d.\n" , x)**

**void main()**

**{**

**int x = 4;**

**PR(SQUARE(x + 2));**

**}</b></p>**

1. <p><b>36</b></p>

2. <p><b>14</b></p>

3. <p><b>26</b></p>

4. <p><b>16</b></p>

Ответ: 2

Комментарий: AP016047

**<p><b>Какое число будет выведено в консоль в данном примере?**

**#include<stdio.h>**

**#define SQUARE(x) x\*x**

**#define PR(x) printf("x = %d.\n" , x)**

**void main()**

**{**

**int x = 4;**

**PR(100/SQUARE(2));**

**}</b></p>**

1. <p><b>100</b></p>

2. <p><b>25</b></p>

3. <p><b>50</b></p>

4. <p><b>20</b></p>

Ответ: 1

Комментарий: AP016048

**<p><b>Какое число будет выведено в консоль в данном примере?**

**#include<stdio.h>**

**#define SQUARE(x) x\*x**

**#define PR(x) printf("x = %d.\n" , x)**

**void main()**

**{**

**int x = 4;**

**PR(SQUARE(++ x));**

**}</b></p>**

1. <p><b>16</b></p>

2. <p><b>20</b></p>

3. <p><b>25</b></p>

4. <p><b>30</b></p>

Ответ: 4

Комментарий: AP016049

**<p><b>Какие утверждения о директивах препроцессора языка С верны?</b></p>**

1.<p><b>Каждая директива начинается со знака кривой решётки</b></p>

2.<p><b>Действие каждой директивы распространяется до конца файла</b></p>

3.<p><b>Директивы должны быть написаны только большими латинскими буквами</b></p>

4.<p><b>При современном уровне развития языка С, большинство директив могут быть безболезненно заменены конструкциями языка</b></p>

Ответ: 1, 2

Комментарий: AP016050

**<p><b>Какие функции выполняют директивы препроцессора языка С?</b></p>**

1.<p><b>Включают в текст программы другие файлы</b></p>

2.<p><b>Вводят в исходный код символические константы</b></p>

3.<p><b>Позволяют выполнять условную компиляцию</b></p>

4.<p><b>Возбуждают ошибки времени исполнения программы</b></p>

Ответ: 1, 2, 3

Комментарий: AP016051

**<p><b>К каким действиям препроцессора приводит использование директивы #include?</b></p>**

1.<p><b>Используется для создания макросов</b></p>

2.<p><b>Препроцессор находит в файловой системе файл, имя которого задано в директиве</b></p>

3.<p><b>Препроцессор вставляет содержимое включаемого файла в то место, где была директива</b></p>

4.<p><b>Препроцессор игнорирует директивы #include, которые встречает во включаемом файле</b></p>

Ответ: 2, 3

Комментарий: AP016052

**<p><b>Как определить символьную константу DOG директивой #define?</b></p>**

1.<p><b>#define DOG '5'</b></p>

2.<p><b>#define DOG = 5</b></p>

3.<p><b>#define(DOG)</b></p>

3.<p><b>#define 5! = DOG</b></p>

Ответ: 1

Комментарий: AP016053

**<p><b> Что будет выведено на консоль после выполнения данной части программы:**

**#define P 2**

**int main()**

**{**

**system("chcp 1251");**

**system("cls");**

**#if P==1**

**printf("Выполняется ветка 1");**

**#elif P==2**

**printf("Выполняется ветка 2, P=%d", P);**

**#else**

**printf("Выполняется другая ветка, P=%d", P);**

**#endif**

**</b></p>**

1.<p><b>Выполняется ветка 1</b></p>

2.<p><b>Выполняется ветка 2, P=2</b></p>

2.<p><b>Выполняется другая ветка, P=3</b></p>

Ответ: 2

Комментарий: AP016054

**<p><b>Что делает директива #undef ?</b></p>**

1.<p><b>Подобно директиве #define создает новое имя</b></p>

2.<p><b>Удаляет имя, ранее созданное с помощью #define</b></p>

3.<p><b>Предлагает компилятору включить другой исходный файл</b></p>

Ответ: 2

Комментарий: AP016055

**<p><b>Для чего используется директива #error ?</b></p>**

1.<p><b>Позволяет управлять возможностями компилятора</b></p>

2.<p><b>Изменяет внутренний счетчик строк компилятора</b></p>

3.<p><b>Указывает компилятору, что нужно напечатать сообщение об ошибке и прекратить компиляцию</b></p>

Ответ: 3

Комментарий: AP016056

**<p><b>Является ли данное утверждение верным :при конвертировании программ на языках С++, K&R C и ANSI C могут возникнуть трудности в связи с различным вычислением в них выражений. </b></p>**

1. <p><b>Да</b></p>

2. <p><b>Нет</b></p>

Ответ: 1

Комментарий: AP016057

**<p><b>Что выведется на экран после выполнения данной части программы?**

**int x[99];**

**void f()**

**{**

**struct x { int a; };**

**std::cout << sizeof(x);**

**}</b></p>**

1. <p><b> Размер массива</b></p>

2. <p><b> Размер структуры</b></p>

3. <p><b> Размер функции</b></p>

4. <p><b> Размер класса</b></p>

Ответ: 2

Комментарий: AP016058

**<p><b>Выберите правильное утверждение: </b></p>**

1. <p><b> Два структурных типа являются различными, когда они имеют разные члены </b></p>

2. <p><b> Два структурных типа являются одинаковыми, когда они имеют одни и те же члены </b></p>

3. <p><b> Два структурных типа являются различными, даже когда они имеют одни и те же члены</b></p>

Ответ: 3

Комментарий: AP016059

**<p><b> Два списка спецификаторов типа совместимы, когда:</b></p>**

1. <p><b> Они содержат одинаковый набор спецификаторов типа с учетом одинакового размера типов</b></p>

2. <p><b> Они содержат одинаковый набор спецификаторов типа с учетом значений</b></p>

3. <p><b>Они содержат одинаковый набор спецификаторов типа с учетом синонимичности названий </b></p>

Ответ: 3

Комментарий: AP016060

## 18 тема

**<p><b>Какая библиотека требуется для использования манипуляторов форматирования? </b></p>**

1. <p><b>iomanip</b></p>

2. <p><b>iostream</b></p>

3. <p><b>fstream</b></p>

4. <p><b>iotsream</b></p>

Ответ: 1

Комментарий:AP0018001

**<p><b> Что выведет следующий код?</b></p>**

**<p><b>**cout << right << setw(8) **</p></b>**

**<p><b>**<< setprecision(2) << setfill('2') **</p></b>**

**<p><b>**<< (float)1 / 3 << endl; **</p></b>**

1. <p><b>22222.33</b></p>

2. <p><b>02222.33</b></p>

3. <p><b>22220.33</b></p>

4. <p><b>2222.333</b></p>

Ответ: 3

Комментарий: AP0018002

**<p><b>Выберите верные утверждения:</b></p>**

1. <p><b> манипулятор setw(n) определяет ширину поля вывода в n символов</b></p>

2. <p><b>манипулятор fixed не используется по умолчанию</b></p>

3. <p><b>манипулятор setprecision(n) определяет ширину поля в n символов</b></p>

4. <p><b>манипулятор setfill(n) устанавливает символ n как заполнитель</b></p>

Ответ: 1, 4

Комментарий: AP0018003

**<p><b>Что выведет следующий код?</b></p>**

**<p><b>**bool a = true; **</p></b>**

**<p><b>**cout.setf(ios::boolalpha); **</p></b>**

**<p><b>**cout << a << endl; **</p></b>**

1. <p><b>1</b></p>

2. <p><b>ничего</b></p>

3. <p><b>a</b></p>

4. <p><b>true </b></p>

5. <p><b>0</b></p>

Ответ: 4

Комментарий: AP0018004

**<p><b>Что выведет следующий код? </b></p>**

**<p><b>**cout.unsetf(ios::dec); **</p></b>**

**<p><b>**cout.setf(ios::showpos); **</p></b>**

**<p><b>**cout.setf(ios::hex); **</p></b>**

**<p><b>**cout << setw(5) << left << setfill('d') << 29 << endl; **</p></b>**

1. <p><b>29ddd</b></p>

2. <p><b>ddd29</b></p>

3. <p><b>1dddd</b></p>

4. <p><b>dddd1</b></p>

5. <p><b>ddd1d</b></p>

Ответ: 3

Комментарий: AP0018005

## 19 тема

**<p><b>Чем отличаются обращения к данным структуры через “.” и “->”?</b></p>**

1. <p><b>Ничем, они полностью идентичны</b></p>

2. <p><b>Точка предоставляет доступ к данным через указатель на структуру, а “стрелка” на прямую к данным структуры</b></p>

3. <p><b>“Стрелка ” предоставляет доступ к данным через указатель на структуру, а точка на прямую к данным структуры</b></p>

Ответ: 3

Комментарий: AP0019001

**<p><b>Что не так с вводам данных в структуру в коде ниже?**

**<p><b>**void enterCarInfo(Car\* item) {**</p></b>**

**<p><b>**sdt::cin >> item->doors; **</p></b>**

**<p><b>**std::cin >> (\*item).wheels; **</p></b>**

**<p><b>**std::cin >> item.windows; **</p></b>**

**<p><b>**}**</p></b>**

1. <p><b> Код правильный</b></p>

2. <p><b>Ошибка при вводе doors</b></p>

3. <p><b>Ошибка при вводе wheels</b></p>

4. <p><b>Ошибка при вводе windows </b></p>

Ответ: 4

Комментарий: AP0019002

**<p><b>С помощью какого(-их) способа(-ов) можно передать структуру в функцию?</b></p>**

1. <p><b>передача структуры по значению</b></p>

2. <p><b>передача указателя на структуру</b></p>

3. <p><b>передача ссылки на структуру</b></p>

4. <p><b>все варианты</b></p>

Ответ: 1, 2

Комментарий: AP0019003

**<p><b>Сколько байт занимает данный шаблон структуры?**

**<p><b>**struct Car**</p></b>**

**<p><b>**{**</p></b>**

**<p><b>**unsigned short doors:4; **</p></b>**

**<p><b>**unsigned short wheels:4; **</p></b>**

**<p><b>**char name[64]; **</p></b>**

**<p><b>**};**</p></b>**

1. <p><b>72 байта</b></p>

2. <p><b>66 байт</b></p>

3. <p><b>65 байт</b></p>

4. <p><b>0 байт</b></p>

Ответ: 4

Комментарий: AP0019004

**<p><b>Сколько байт занимает данная структура?**

**<p><b>**struct Car**</p></b>**

**<p><b>**{**</p></b>**

**<p><b>**unsigned short doors:3; **</p></b>**

**<p><b>**unsigned short wheels:5; **</p></b>**

**<p><b>**char name[59]; **</p></b>**

**<p><b>**} Audi; **</p></b>**

1. <p><b>67 байт</b></p>

2. <p><b>68 байт</b></p>

3. <p><b>0 байт</b></p>

4. <p><b>60 байт</b></p>

Ответ: 4

Комментарий: AP0019005

## 20 тема

<p><b>Структура данных представляет собой</b></p>

1. <p><b> набор правил и ограничений, определяющих связи между отдельными элементами и группами данных</b></p>
2. <p><b> набор правил и ограничений, определяющих связи между отдельными элементами данных</b></p>
3. <p><b> набор правил и ограничений, определяющих связи между отдельными группами данных</b></p>
4. <p><b> некоторую иерархию данных</b></p>

Ответ:1

Комментарий: AP00200001

<p><b>Линейный список, в котором доступен только последний элемент, называется</b></p>

1. <p><b>стеком</b></p>
2. <p><b>очередью</b></p>
3. <p><b>деком</b></p>
4. <p><b>массивом</b></p>

Ответ:1

Комментарий: AP002000

<p><b>В чём особенности стека ?</b></p>

1. <p><b>открыт с обеих сторон на вставку и удаление;</b></p>
2. <p><b>доступен любой элемент;</b></p>
3. <p><b>открыт с одной стороны на вставку и удаление.</b></p>

Ответ: 3

Комментарий: AP0020003

<p><b>Каково правило выборки элемента из стека ?</b></p>

1. <p><b>первый элемент;</b></p>
2. <p><b>последний элемент;</b></p>
3. <p><b>любой элемент.</b></p>

Ответ: 2

Комментарий: AP0020004

<p><b>Стандартным способом устранения рекурсии при поиске в глубину является использование:</b></p>

1. <p><b>массива;</b></p>
2. <p><b>очереди;</b></p>
3. <p><b>стека;</b></p>
4. <p><b>циклического списка.</b></p>

Ответ: 3

Комментарий: AP0020005

<p><b>Структура данных работа с элементами которой организована по принципу FIFO (первый пришел - первый ушел) это –</b></p>

1. <p><b>Стек</b></p>
2. <p><b>Дек</b></p>
3. <p><b>Очередь</b></p>
4. <p><b>Список</b></p>

Ответ: 3

Комментарий: AP0020006

<p><b> В чём особенности очереди ?</b></p>

1. <p><b>открыта с обеих сторон ;</b></p>
2. <p><b>открыта с одной стороны на вставку и удаление;</b></p>
3. <p><b>доступен любой элемент.</b></p>

Ответ: 1

Комментарий: AP0020007

<p><b>Какую дисциплину обслуживания принято называть FIFO ?</b></p>

1. <p><b>стек;</b></p>
2. <p><b>очередь;</b></p>
3. <p><b>дек.</b></p>

Ответ: 2

Комментарий: AP0020008

<p><b>С помощью какой структуры данных наиболее рационально реализовать очередь ?</b></p>

1. <p><b>стек;</b></p>
2. <p><b>список </b></p>
3. <p><b>дек.</b></p>

Ответ: 2

Комментарий: AP0020009

<p><b>При поиске в ширину используется:</b></p>

1. <p><b>массив;</b></p>
2. <p><b> очередь;</b></p>
3. <p><b>стек;</b></p>
4. <p><b> циклический список.</b></p>

Ответ: 2

Комментарий: AP0020010

<p><b> Линейный последовательный список, в котором включение исключение элементов возможно с обоих концов, называется</b></p>

1. <p><b> стеком</b></p>
2. <p><b>очередью</b></p>
3. <p><b> деком</b></p>
4. <p><b>кольцевой очередью</b></p>

Ответ:3

Комментарий: AP0020011

<p><b>В памяти ЭВМ бинарное дерево удобно представлять в виде:</b></p>

1. <p><b>связанных линейных списков;</b></p>
2. <p><b>массивов;</b></p>
3. <p><b>связанных нелинейных списков </b></p>

Ответ:2

Комментарий: AP0020012

<p><b>Элемент t, на который нет ссылок называется</b></p>

1. <p><b>корнем </b></p>
2. <p><b>промежуточным;</b></p>
3. <p><b>терминальным (лист).</b></p>

Ответ:1

Комментарий: AP0020013

<p><b>Дерево называется полным бинарным, если степень исходов вершин равна:</b></p>

1. <p><b>2 или 0</b></p>
2. <p><b>2;</b></p>
3. <p><b>М или 0;</b></p>
4. <p><b>M.</b></p>

Ответ:1

Комментарий: AP0020014

<p><b>Существуют следующие методы сортировки. Найдите ошибку.</b></p>

1. <p><b>строгие;</b></p>
2. <p><b>улучшенные;</b></p>
3. <p><b>динамические </b></p>

Ответ:3

Комментарий: AP0020015

<p><b>Что из перечисленных ниже понятий является одним из типов сортировки ?</b></p>

1. <p><b>внутренняя сортировка </b></p>
2. <p><b>сортировка по убыванию;</b></p>
3. <p><b>сортировка данных;</b></p>
4. <p><b>сортировка по возрастанию.</b></p>

Ответ:1

Комментарий: AP0020016

<p><b>Как отсортировать массив быстрее, пользуясь пузырьковым методом?</b></p>

1. <p><b>одинаково </b></p>
2. <p><b>по возрастанию элементов;</b></p>
3. <p><b>по убыванию элементов.</b></p>

Ответ:1

Комментарий: AP0020017

<p><b> Массив сортируется “пузырьковым” методом. За сколько проходов по массиву самый “лёгкий” элемент в массиве окажется вверху ?</b></p>

1. <p><b>за 1 проход </b></p>
2. <p><b>за n-1 проходов;</b></p>
3. <p><b>за n проходов, где n – число элементов массива.</b></p>

Ответ:1

Комментарий: AP0020018

<p><b>При обходе дерева слева направо получаем последовательность…</b></p>

1. <p><b>отсортированную по убыванию;</b></p>
2. <p><b>не отсортированную </b></p>
3. <p><b>отсортированную по возрастанию.</b></p>

Ответ:2

Комментарий: AP0020019

<p><b> В чём суть бинарного поиска ?</b></p>

1. <p><b>нахождение элемента массива x путём деления массива пополам каждый раз, пока элемент не найден </b></p>
2. <p><b>нахождение элемента x путём обхода массива;</b></p>
3. <p><b>нахождение элемента массива х путем деления массива.</b></p>

Ответ:1

Комментарий: AP0020020

<p><b>Как расположены элементы в массиве бинарного поиска ?</b></p>

1. <p><b>по возрастанию </b></p>
2. <p><b>хаотично;</b></p>
3. <p><b>по убыванию.</b></p>

Ответ:1

Комментарий: AP0020021

<p><b> В чём суть линейного поиска ?</b></p>

1. <p><b>производится последовательный просмотр от начала до конца и обратно через 2 элемента;</b></p>
2. <p><b> производится последовательный просмотр элементов от середины таблицы;</b></p>
3. <p><b>производится последовательный просмотр каждого элемента </b></p>

Ответ:3

Комментарий: AP0020022

<p><b>Элемент дерева, который не ссылается на другие, называется</b></p>

1. <p><b> корнем</b></p>
2. <p><b>листом</b></p>
3. <p><b> узлом</b></p>
4. <p><b>промежуточным </b></p>

Ответ:2

Комментарий: AP0020023

<p><b>Элемент дерева, на который не ссылаются другие, называется</b></p>

1. <p><b>корнем</b></p>
2. <p><b>листом</b></p>
3. <p><b> узлом</b></p>
4. <p><b> промежуточным</b></p>

Ответ:1

Комментарий: AP0020024

<p><b>Элемент дерева, который имеет предка и потомков, называется</b></p>

1. <p><b>корнем</b></p>
2. <p><b>листом</b></p>
3. <p><b>узлом</b></p>
4. <p><b>промежуточным</b></p>

Ответ:4

Комментарий: AP0020025

<p><b> Высотой дерева называется

1. <p><b>максимальное количество узлов</b></p>
2. <p><b>максимальное количество связей</b></p>
3. <p><b>максимальное количество листьев</b></p>
4. <p><b> максимальная длина пути от корня до листа</b></p>

Ответ:4

Комментарий: AP0020026

<p><b> Степенью дерева называется</b></p>

1. <p><b>максимальная степень всех узлов</b></p>
2. <p><b>максимальное количество уровней его узлов</b></p>
3. <p><b>максимальное количество узлов</b></p>
4. <p><b> максимальное количество связей</b></p>
5. <p><b>максимальное количество листьев</b></p>

Ответ:1

Комментарий: AP0020027

<p><b>Как определяется длина пути дерева</b></p>

1. <p><b>как сумма длин путей всех его узлов</b></p>
2. <p><b>как количество ребер от узла до вершины</b></p>
3. <p><b>как количество ребер от листа до вершины</b></p>
4. <p><b>как максимальное количество ребер</b></p>
5. <p><b> как максимальное количество листьев</b></p>
6. <p><b>как длина самого длинного пути от ближайшего узла до какого-либо листа</b></p>

Ответ:6

Комментарий: AP0020028

<p><b>Бинарное дерево можно представить</b></p>

1. <p><b> с помощью указателей</b></p>
2. <p><b> с помощью массивов</b></p>
3. <p><b> с помощью индексов</b></p>
4. <p><b> правильного ответа нет</b></p>

Ответ:1,2

Комментарий: AP0020029

<p><b> Как называются предки узла, имеющие уровень на единицу меньше уровня самого узла</b></p>

1. <p><b>детьми</b></p>
2. <p><b>родителями</b></p>
3. <p><b>братьями </b></p>

Ответ:2

Комментарий: AP0020030

<p><b>Что такое граф?</b></p>

1. <p><b>Графом называется система объектов произвольной природы (вершин) и связок (рёбер), соединяющих некоторые пары этих объектов.</b></p>
2. <p><b>Упорядоченная система объектов, имеющая иерархическую <p><b>структуру.</b></p>
3. <p><b>Граф - [королевское](https://ru.wikipedia.org/wiki/%D0%9A%D0%BE%D1%80%D0%BE%D0%BB%D1%8C) [должностное лицо](https://ru.wikipedia.org/wiki/%D0%94%D0%BE%D0%BB%D0%B6%D0%BD%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D0%B5_%D0%BB%D0%B8%D1%86%D0%BE) в [Раннем Средневековье](https://ru.wikipedia.org/wiki/%D0%A0%D0%B0%D0%BD%D0%BD%D0%B5%D0%B5_%D0%A1%D1%80%D0%B5%D0%B4%D0%BD%D0%B5%D0%B2%D0%B5%D0%BA%D0%BE%D0%B2%D1%8C%D0%B5) в [Западной Европе](https://ru.wikipedia.org/wiki/%D0%97%D0%B0%D0%BF%D0%B0%D0%B4%D0%BD%D0%B0%D1%8F_%D0%95%D0%B2%D1%80%D0%BE%D0%BF%D0%B0).</b></p>
4. <p><b>Модель взаимосвязей между группами объектов и их представление в памяти программы.</b></p>

Ответ:1

Комментарий: AP0020031

<p><b>Бинарная куча представляет собой</b></p>

1. <p><b> бинарное дерево, для которого выполняется
2. основное свойство кучи: приоритет каждой вершины больше приоритетов её потомков</b></p>
3. <p><b> набор правил и ограничений, определяющих связи между отдельными элементами данных</b></p>
4. <p><b> некоторую иерархию данных</b></p>
5. <p><b> Все утверждения верны</b></p>

Ответ:1

Комментарий: AP0020032

<p><b>Бинарное дерево - это структура, имеющая следующие свойства</b></p>

1. <p><b> существует единственный элемент (узел, вершина), на который не ссылается никакой другой и который называется *корнем*</b></p>
2. <p><b> начиная с корня и следуя по определенной цепочке указателей, можно осуществить доступ к любому элементу структуры</b></p>
3. <p><b> на каждый элемент, кроме корня, имеется

единственная ссылка</b></p>

1. <p><b> Все утверждения верны</b></p>

Ответ:5

Комментарий:AP0020033

<p><b>Что такое Дек</b></p>

1. <p><b> двусторонняя очередь</b></p>
2. <p><b> набор правил и ограничений, определяющих связи между отдельными элементами данных</b></p>
3. <p><b> иерархия данных</b></p>
4. <p><b> Все утверждения верны</b></p>

Ответ:1

Комментарий:AP0020034

<p><b>В чём особенность односвязных списков</b></p>

1. <p><b>каждый элемент имеет указатель на следующий</b></p>
2. <p><b>первый и последний элементы ссылаются друг на друга</b></p>
3. <p><b>каждый элемент имеет указатель на следующий и на предыдущий</b></p>
4. <p><b>каждый элемент имеет указатель на первый и последний элемент</b></p>

Ответ:1

Комментарий:AP0020035

<p><b>Каким свойством обладает очередь(как динамическая структура данных)</b></p>

1. <p><b> добавление новых элементов допустимо с одного конца</b></p>
2. <p><b>каждый элемент имеет указатель на следующий и на предыдущий</b></p>
3. <p><b>каждый элемент имеет указатель на первый и последний элемент</b></p>
4. <p><b>Все утверждения верны</b></p>

Ответ:1

Комментарий:AP0020036

<p><b>Каким свойством обладает двусвязный циклический список</b></p>

1. <p><b>первый и последний элементы ссылаются друг на друга</b></p>
2. <p><b>каждый элемент имеет указатель на следующий и на предыдущий</b></p>
3. <p><b>каждый элемент имеет указатель на первый и последний элемент</b></p>
4. <p><b>Все утверждения верны</b></p>

Ответ:1

Комментарий:AP0020037

<p><b>Какое свойство подходит к двусвязным списком</b></p>

1. <p><b> существует единственный элемент (узел, вершина), на который не ссылается никакой другой и который называется *корнем*</b></p>

2. <p><b> начиная с корня и следуя по определенной цепочке указателей, можно осуществить доступ к любому элементу структуры</b></p>

3. <p><b> каждый элемент имеет указатели на следующий и на предыдущий элементы</b></p>

4. <p><b> Все утверждения верны</b></p>

Ответ:3

Комментарий: AP0020038

<p><b>Виды списков(как динамические структуры данных)</b></p>

1. <p><b>односвязные </b></p>
2. <p><b>корневые</b></p>
3. <p><b>прототипные</b></p>
4. <p><b>первичные</b></p>

Ответ:1

Комментарий: AP0020039

<p><b>На основе чего может быть реализован стек</b></p>

1. <p><b>на основе массива</b></p>
2. <p><b>на основе списка</b></p>
3. <p><b>на основе бинарной кучи </b></p>
4. <p><b>Все варианты верны</b></p>

Ответ:1,2

Комментарий: AP0020040

<p><b>struct Node

{

int key;

int count; //счетчик дубликатов

Node \*left, \*right;

};

представляет

</b></p>

1. <p><b>Бинарное дерево </b></p>
2. <p><b>Односвязный список</b></p>
3. <p><b>Структуру кодирование информации</b></p>
4. <p><b>Матрицу смежности</b></p>

Ответ: 1

Комментарий: AP0020041

<p><b>Основные способы хранения информации о структуре графов в памяти:</b></p>

1. <p><b>Матрицы смежности</b></p>
2. <p><b>Список смежности</b></p>
3. <p><b>Матрицы инцидентности</b></p>
4. <p><b>Список ребер </b></p>

Ответ: 1, 3, 4

Комментарий: AP0020042

## 21 тема

<p><b> Выберите виды рекурсии:</b></p>

<p><b>1) Ветвящаяся </b></p>

<p><b>2) Косвенная </b></p>

<p><b>3) Динамическая</b></p>

<p><b>4) Сложная </b></p>

Ответ: 1,2,4

Комментарий: AP0021001

<p><b> Основное преимущество реализации алгоритма при помощи рекурсии - ...</b></p>

<p><b>1) Более краткая и читаемая запись кода</b></p>

<p><b>2) Экономия памяти стека</b></p>

<p><b>3) Повышение производительности</b></p>

<p><b>4) Возможность задать более гибкое условие окончания рекурсии</b></p>

Ответ: 1

Комментарий: AP0021002

<p><b> Глубиной рекурсией называется...</b></p>

<p><b>1) Количество одновременно выполняемых процедур </b></p>

<p><b>2) Всё, что находится внутри рекурсивной функции</b></p>

<p><b>3) Способ определение объекта через себя</b></p>

<p><b>4) Время выполнения рекурсивного алгоритма</b></p>

Ответ: 1

Комментарий: AP0021003

<p><b> Геометрические фигуры, обладающие свойством самоподобия, то есть состоящие из частей, подобных всей фигуре - это...</b></p>

<p><b>1) Деревья</b></p>

<p><b>2) Деки</b></p>

<p><b>3) Фракталы</b></p>

<p><b>4) Графы</b></p>

Ответ: 3

Комментарий: AP0021004

<p><b> Все, что находится внутри рекурсивной функции</b></p>

<p><b>1) Глубина рекурсии</b></p>

<p><b>2) Базис рекурсии</b></p>

<p><b>3) Подпрограмма</b></p>

<p><b>4) Глубина рекурсии</b></p>

Ответ: 3

Комментарий: AP0021005

<p><b> Рекурсия называется ветвящейся, если...</b></p>

<p><b>1) Исполнение подпрограммы приводит только к одному вызову этой же самой подпрограммы</b></p>

<p><b>2) Имеется циклическая последовательность вызовов нескольких алгоритмов</b></p>

<p><b>3) Происходит непосредственный вызов алгоритма (функции, процедуры, метода) из текста самого метода</b></p>

<p><b>4) Каждый экземпляр подпрограммы может вызвать себя несколько раз</b></p>

Ответ: 4

Комментарий: AP0021006

<p><b> Рекурсия называется косвенной, если...</b></p>

<p><b>1) Исполнение подпрограммы приводит только к одному вызову этой же самой подпрограммы</b></p>

<p><b>2) Имеется циклическая последовательность вызовов нескольких алгоритмов</b></p>

<p><b>3) Происходит непосредственный вызов алгоритма (функции, процедуры, метода) из текста самого метода</b></p>

<p><b>4) Каждый экземпляр подпрограммы может вызвать себя несколько раз</b></p>

Ответ: 2

Комментарий: AP0021007

<p><b> Рекурсия называется прямой, если...</b></p>

<p><b>1) Исполнение подпрограммы приводит только к одному вызову этой же самой подпрограммы</b></p>

<p><b>2) Имеется циклическая последовательность вызовов нескольких алгоритмов</b></p>

<p><b>3) Происходит непосредственный вызов алгоритма (функции, процедуры, метода) из текста самого метода</b></p>

<p><b>4) Каждый экземпляр подпрограммы может вызвать себя несколько раз</b></p>

Ответ: 3

Комментарий: AP0021008

<p><b> Рекурсия называется линейной, если...</b></p>

<p><b>1) Исполнение подпрограммы приводит только к одному вызову этой же самой подпрограммы</b></p>

<p><b>2) Имеется циклическая последовательность вызовов нескольких алгоритмов</b></p>

<p><b>3) Происходит непосредственный вызов алгоритма (функции, процедуры, метода) из текста самого метода</b></p>

<p><b>4) Каждый экземпляр подпрограммы может вызвать себя несколько раз</b></p>

Ответ: 1

Комментарий: AP0021009

<p><b> Минусы рекурсии: </b></p>

<p><b>1) Занимает много оперативной памяти </b></p>

<p><b>2) Более краткая и читаемая запись кода</b></p>

<p><b>3) Минусов нет</b></p>

<p><b>4) Имеет свойство аварийно завершить программу</b></p>

Ответ: 1

Комментарий: AP0021010

<p><b> Какой способ сортировки массивов основан на рекурсии?</b></p>

<p><b>1) Быстрая сортировка </b></p>

<p><b>2) Сортировка “пузырьком”</b></p>

<p><b>3) Сортировка вставкой</b></p>

<p><b>4) Сортировка выбором</b></p>

Ответ: 1

Комментарий: AP0021011

<p><b> Как называется рекурсия, при которой функция А вызывает функцию В, а та в свою очередь вызывает А?</b></p>

<p><b>1) Бесконечная рекурсия</b></p>

<p><b>2) Прямая рекурсия</b></p>

<p><b>3) Сложная рекурсия </b></p>

<p><b>4) Ветвящаяся рекурсия</b></p>

Ответ: 3

Комментарий: AP0021012

<p><b> Шаг рекурсии - это...</b></p>

<p><b>1) Способ общего определения объекта или действия через себя, с использованием ранее заданных частных определений</b></p>

<p><b>2) Это предложение, определяющее некую начальную ситуацию или ситуацию в момент прекращения </b></p>

<p><b>3) Количество одновременно выполняемых процедур</b></p>

<p><b>4) Это правило, в теле которого обязательно содержится, в качестве подцели, вызов определяемого предиката</b></p>

Ответ: 4

Комментарий: AP0021013

<p><b>Вопрос : Базис рекурсии - это...</b></p>

<p><b>1) Количество одновременно выполняемых процедур</b></p>

<p><b>2) Это правило, в теле которого обязательно содержится, в качестве подцели, вызов определяемого предиката</b></p>

<p><b>3) Это предложение, определяющее некую начальную ситуацию или ситуацию в момент прекращения</b></p>

<p><b>4) Способ общего определения объекта или действия через себя, с использованием ранее заданных частных определений</b></p>

Ответ: 3

Комментарий: AP0021014

<p><b> Итерация - это...</b></p>

<p><b>1) Инициализация данных</b></p>

<p><b>2) Один шаг в циклическом процессе</b></p>

<p><b>3) Получение значения переменной по её адресу</b></p>

<p><b>4) Выделение памяти под новый объект</b></p>

Ответ: 2

Комментарий: AP0021015

<p><b> Рекурсивный спуск - это ...</b></p>

<p><b>1) Уменьшение значения степени выражения</b></p>

<p><b>2) Увеличение значения степени выражения</b></p>

<p><b>3) Завершение работы рекурсивных подпрограмм, вплоть до самой первой, инициировавшей рекурсивные вызовы</b></p>

<p><b>4) Порождение все новых копий подпрограммы до выхода на граничное условие</b></p>

Ответ: 4

Комментарий: AP0021016

<p><b> Рекурсивный подъем - это …</b></p>

<p><b>1) Увеличение значения степени выражения</b></p>

<p><b>2) Завершение работы рекурсивных подпрограмм, вплоть до самой первой, инициировавшей рекурсивные вызовы</b></p>

<p><b>3) Уменьшение значения степени выражения</b></p>

<p><b>4) Порождение все новых копий подпрограммы до выхода на граничное условие</b></p>

Ответ: 2

Комментарий: AP0021017

<p><b>Вопрос : Граничное условие - это ...</b></p>

<p><b>1) Условие, с которого начинается инициализация рекурсивных обращений</b></p>

<p><b>2) Условие, на котором прекращается инициализация рекурсивных обращений</b></p>

<p><b>3) Условие, на котором прекращается работа программы</b></p>

<p><b>4) Условие, с которого начинается работа программы</b></p>

Ответ: 2

Комментарий: AP0021018

<p><b> Совокупность множества вершин и множества ребер называется ...</b></p>

<p><b>1) Дерево</b></p>

<p><b>2) Граф</b></p>

<p><b>3) Дек</b></p>

<p><b>4) Фрактал</b></p>

Ответ: 2

Комментарий: AP0021019

<p><b> При каждом новом рекурсивном обращении ...</b></p>

<p><b>1) Происходит очистка памяти</b></p>

<p><b>2) Создается новая копия подпрограммы в памяти</b></p>

<p><b>3) Переменные аннулируются</b></p>

<p><b>4) Происходит обновление переменных</b></p>

Ответ: 2

Комментарий: AP0021020

<p><b> Графы могут быть:</b></p>

<p><b>1) Зависимые</b></p>

<p><b>2) Ориентированными</b></p>

<p><b>3) Независимые</b></p>

<p><b>4) Неориентированными</b></p>

Ответ: 2, 4

Комментарий: AP0021021

<p><b> Путем в графе называется:</b></p>

<p><b>1) Расстояние между ребрами</b></p>

<p><b>2) Все ответы верны</b></p>

<p><b>3) Последовательность вершин, каждая из которых соединена со следующей ребром</b></p>

<p><b>4) Расстояние между вершинами</b></p>

Ответ: 3

Комментарий: AP0021022

<p><b> F(0)=1;F(1)=1;F(n)=F(n-1)\*F(n-2)+5,при n>1.Чему равно значение функции F(3)?</b></p>

<p><b>1) 71</b></p>

<p><b>2) 11</b></p>

<p><b>3) 66</b></p>

<p><b>4) 6</b></p>

Ответ: 2

Комментарий: AP0021023

<p><b>int F(int n){if(n>2) return F(n-1)+F(n-2) else return 1;} Чему равно значение функции F(3)?</b></p>

<p><b>1) 7</b></p>

<p><b>2) 71</b></p>

<p><b>3) 2</b></p>

<p><b>4) 1</b></p>

Ответ: 3

Комментарий: AP0021024

<p><b> int F(int n){if(n>2) return F(n-1)+F(n-2) else return n;}Чему равно значение функции F(3)?</b></p>

<p><b>1) 7</b></p>

<p><b>2) 5</b></p>

<p><b>3) 1</b></p>

<p><b>4) 3</b></p>

Ответ: 4

Комментарий: AP0021025

<p><b> void F(int n) {std::cout << n;if (n >= 3) {F(n / 2);F(n - 1);} } Посчитайте сумму всех выведенных чисел при выполнении вызова F(4). </b></p>

<p><b>1) 5</b></p>

<p><b>2) 12</b></p>

<p><b>3) 9</b></p>

<p><b>4) 13</b></p>

Ответ: 2

Комментарий: AP0021026

<p><b> void A(int n){ printf(“@”); if(n>2) A(n-2); if(n>0) A(n-1);}Сколько символов будет напечатано на экране при выполнении вызова F(3).</b></p>

<p><b>1) 6</b></p>

<p><b>2) 12</b></p>

<p><b>3) 5</b></p>

<p><b>4) 20</b></p>

Ответ: 1

Комментарий:AP0021027

<p><b> Если имеется циклическая последовательность вызовов нескольких алгоритмов, то рекурсия называется...</b></p>

<p><b>1) Линейной</b></p>

<p><b>2) Прямой</b></p>

<p><b>3) Ветвящейся</b></p>

<p><b>4) Косвенной</b></p>

Ответ: 4

Комментарий: AP0021028

<p><b> Рекурсия - это определение объекта через…</b></p>

<p><b>1) Цикл</b></p>

<p><b>2) Условные операторы</b></p>

<p><b>3) Обращение к самому себе</b></p>

<p><b>4) Другие объекты</b></p>

Ответ: 3

Комментарий: AP0021029

<p><b> Рекурсивная триада - это этапы моделирования задачи, на которых определяется… </b></p>

<p><b>1) Место использования рекурсии</b></p>

<p><b>2) Набор параметров</b></p>

<p><b>3) Соотношение между параметрами</b></p>

<p><b>4) Три обязательных параметра </b></p>

Ответ: 2, 3

Комментарий: AP0021030

<p><b> Если исполнение подпрограммы приводит только к одному вызову этой же самой подпрограммы, то рекурсия называется… </b></p>

<p><b>1) Линейной</b></p>

<p><b>2) Прямой</b></p>

<p><b>3) Ветвящейся</b></p>

<p><b>4) Косвенной</b></p>

Ответ: 1

Комментарий: AP0021031

<p><b> Чему будет равна переменная res в результате выполнения программы?

int func(int N)

{

if (N == 0) return 1;

else return ( N \* func(N - 1) );

}

void main()

{

int res = func(5);

}</b></p>

<p><b>1) 24</b></p>

<p><b>2) 1</b></p>

<p><b>3) 120</b></p>

<p><b>4) 5</b></p>

Ответ: 3

Комментарий: AP0021032

<p><b> Элемент бинарного дерева не имеющий потомков - это...</b></p>

<p><b>1) Корень</b></p>

<p><b>2) Ветвь</b></p>

<p><b>3) Лист</b></p>

<p><b>4) Поддерево</b></p>

Ответ: 3

Комментарий: AP0021033

<p><b> Высота дерева - это… </b></p>

<p><b>1) Кол-во элементов дерева </b></p>

<p><b>2) Кол-во ветвей дерева </b></p>

<p><b>3) Максимальное значение ключа </b></p>

<p><b>4) Максимальный уровень листа дерева</b></p>

Ответ: 4

Комментарий: AP0021034

<p><b>Какому обходу бинарного дерева соответствует следующий алгоритм посещения: Корень - Левое поддерево - Правое поддерево </b></p>

<p><b>1) Обратный обход</b></p>

<p><b>2) Прямой обход</b></p>

<p><b>3) Симметричный обход</b></p>

<p><b>4) Обход в глубину</b></p>

Ответ: 2

Комментарий: AP0021035

<p><b> Совокупность **узлов (вершин)** и соединяющих их направленных **ребер (дуг)**, причем в каждый узел (за исключением одного - **корня**) ведет ровно одна дуга, это:</b></p>

<p><b>1) Дерево</b></p>

<p><b>2) Список</b></p>

<p><b>3) Стек</b></p>

<p><b>4) Куча</b></p>

Ответ: 1

Комментарий: AP0021036

<p><b> Для матрицы смежности правдивы утверждения:</b></p>

<p><b>1) Каждый элемент с индексами (i,j) является логическим значением</b></p>

<p><b>2) Каждый элемент с индексами (i,j) равен “весу” ребра из вершины i в вершину j</b></p>

<p><b>3) Каждый элемент с индексами (i,j) показывает, есть ли дуга из вершины i в вершину j</b></p>

<p><b>4) Указывает, есть ли связь между вершинами</b></p>

Ответ: 1, 3, 4

Комментарий: AP0021037

<p><b> Когда деревья являются подобными?</b></p>

<p><b>1) Когда вершины содержат одинаковую информацию</b></p>

<p><b>2) Когда деревья имеют одинаковую структуру</b></p>

<p><b>3) Когда деревья имеют одинаковое количество вершин </b></p>

<p><b>4) Когда деревья имеют одинаковую глубину </b></p>

Ответ: 2

Комментарий: AP0021038

<p><b> Сыном узла x называется узел дерева, ...</b></p>

<p><b>1) в который существует непосредственная дуга из узла x</b></p>

<p><b>2) из которого существует непосредственная дуга в узел x</b></p>

<p><b>3) в который существует путь (по стрелкам) из узла x</b></p>

<p><b>4) из которого существует путь в узел x</b></p>

Ответ: 1

Комментарий: AP0021039

<p><b> Какой вид графа разрешает наличие между двумя вершинами нескольких ребер?</b></p>

<p><b>1) Монограф </b></p>

<p><b>2) Мультиграф</b></p>

<p><b>3) Циклический </b></p>

<p><b>4) Макрограф</b></p>

Ответ: 2

Комментарий: AP0021040

<p><b> Выберите способы представления графов.</b></p>

<p><b>1) Двойной цикл</b></p>

<p><b>2) Список смежности</b></p>

<p><b>3) Матрица смежности</b></p>

<p><b>4) Матрица инцидентности</b></p>

Ответ: 2, 3

Комментарий: AP0021041

<p><b> Что такое вес бинарного дерева?</b></p>

<p><b>1) Число листьев</b></p>

<p><b>2) Число узлов</b></p>

<p><b>3) Число узлов, ссылающихся на нулевой элемент</b></p>

<p><b>4) Число дуг дерева</b></p>

Ответ: 1

Комментарий: AP0021042

<p><b> Что такое дуга бинарного дерева?</b></p>

<p><b>1) Ориентированная связь между двумя вершинами дерева</b></p>

<p><b>2) Не ориентированная связь между двумя вершинами дерева</b></p>

<p><b>3) Путь от корня дерева к любому его листу</b></p>

<p><b>4) Значение, которое характеризуется количеством поддеревьев бинарного дерева</b></p>

Ответ: 1

Комментарий: AP0021043

<p><b> Что такое ребро бинарного дерева?</b></p>

<p><b>1) Неориентированная связь между двумя вершинами дерева.</b></p>

<p><b>2) Ориентированная связь между двумя вершинами дерева</b></p>

<p><b>3) Путь от корня дерева к любому его листу</b></p>

<p><b>4) Значение, которое характеризуется количеством поддеревьев бинарного дерева</b></p>

Ответ: 1

Комментарий: AP0021044

<p><b> Что такое ветвь бинарного дерева?</b></p>

<p><b>1) Путь от корня дерева к любому его листу</b></p>

<p><b>2) Длина внутреннего пути в левом поддереве</b></p>

<p><b>3) Длина внутреннего пути в правом поддереве</b></p>

<p><b>4) Количество узлов в дереве</b></p>

Ответ: 1

Комментарий: AP0021045

<p><b> Деревья Хаффмена - это..</b></p>

<p><b>1) Один из способов кодирования текста</b></p>

<p><b>2) Способ кодирования массивов</b></p>

<p><b>3) Деревья игр</b></p>

<p><b>4) Нет правильного ответа</b></p>

Ответ: 1

Комментарий: AP0021046

<p><b> Данный код

struct Node

{

int key;

int count; //счетчик дубликатов

Node \*left, \*right;

};

представляет...</b></p>

<p><b>1) Бинарное дерево</b></p>

<p><b>2) Односвязный список</b></p>

<p><b>3) Структуру кодирование информации</b></p>

<p><b>4) Матрицу смежности</b></p>

Ответ: 1

Комментарий: AP0021047

<p><b> Реализация бинарной кучи в С/С++ использует следующую особенность языка</b></p>

<p><b>1) Динамическая память</b></p>

<p><b>2) Статическая память</b></p>

<p><b>3) Автоматическая память</b></p>

<p><b>4) Регистровая память</b></p>

Ответ: 1

Комментарий: AP0021048

<p><b> Что такое поддерево?</b></p>

<p><b>1) Часть древообразной структуры данных со степенью не более двух.</b></p>

<p><b>2) Часть структура данных, представляющее собой дерево, в котором каждая вершина имеет не более двух потомков.</b></p>

<p><b>3) Часть древообразной структуры данных, которая может быть представлена в виде отдельного дерева.</b></p>

<p><b>4) Часть дерева, имеющая только одну вершину.</b></p>

Ответ: 3

Комментарий: AP0021049

<p><b> Нисходящий обход узлов дерева-формулы дает запись арифметического выражения...</b></p>

<p><b>1) В постфиксной форме</b></p>

<p><b>2) В префиксной форме</b></p>

<p><b>3) Дает инфиксную запись</b></p>

<p><b>4) Позволяет учитывать приоритет выражений, согласно исходной форме</b></p>

Ответ: 2

Комментарий: AP0021050

<p><b> Выберите утверждения, верные для идеально сбалансированного дерева.</b></p>

<p><b>1) Каждой его вершины количество вершин в левом и правом поддереве различаются не более чем на 1</b></p>

<p><b>2) Каждый уровень содержит не более узлов, где корень - нулевой узел</b></p>

<p><b>3) Для каждой его вершины высота ее двух поддеревьев различается не более, чем на 1</b></p>

<p><b>4) Для поиска любого элемента перебирается не более log2n вершин, где n - число вершин в дереве</b></p>

Ответ: 1, 3, 4

Комментарий: AP0021051

<p><b> Что должно находиться на главной диагонали матрицы смежности графа?</b></p>

<p><b>1) Индексы элементов графа</b></p>

<p><b>2) Единицы</b></p>

<p><b>3) Нули</b></p>

<p><b>4) Вес самого длинного ребра</b></p>

Ответ: 3

Комментарий: AP0021052

<p><b> Сбалансированным бинарным деревом называется...</b></p>

<p><b>1) Дерево, для каждой вершины которого высота двух его поддеревьев различается не более чем на единицу</b></p>

<p><b>2) Дерево, элементы которого распределены таким образом, что слева от корня находятся элементы с меньшим индексом, а справа - с большим</b></p>

<p><b>3) Дерево, в котором у каждого поддерева имеется пара потомков</b></p>

<p><b>4) Дерево, индекс корня которого равен среднему арифметическому всех остальных индексов</b></p>

Ответ: 1

Комментарий: AP0021053

<p><b> Какому обходу бинарного дерева соответствует следующий алгоритм посещения: Левое поддерево - Корень - Правое поддерево?</b></p>

<p><b>1) Прямой обход</b></p>

<p><b>2) Обход в концевом порядке</b></p>

<p><b>3) Обратный обход</b></p>

<p><b>4) Симметричный обход</b></p>

Ответ: 3

Комментарий: AP0021054

<p><b> Для обхода бинарного дерева в концевом порядке характерен следующий алгоритм посещения: </b></p>

<p><b>1) Левое поддерево - Правое поддерево - Корень</b></p>

<p><b>2) Корень - Левое поддерево - Правое поддерево</b></p>

<p><b>3) Правое поддерево - Корень - Левое поддерево</b></p>

<p><b>4) Левое поддерево - Корень - Правое поддерево</b></p>

Ответ: 1

Комментарий: AP0021055

<p><b> Построение фракталов естественно производить с помощью:</b></p>

<p><b>1) Цикла</b></p>

<p><b>2) Рекурсивных процедур</b></p>

<p><b>3) Бинарного дерева</b></p>

<p><b>4) Хэш - таблиц</b></p>

Ответ: 2

Комментарий: AP0021056

<p><b> Основные способы хранения информации о структуре графов в памяти:</b></p>

<p><b>1) Матрицы смежности</b></p>

<p><b>2) Матрицы инцидентности</b></p>

<p><b>3) Список смежности</b></p>

<p><b>4) Список ребер</b></p>

Ответ: 1, 2, 4

Комментарий: AP0021057

<p><b> Древовидные структуры можно изобразить, используя... </b></p>

<p><b>1) Граф</b></p>

<p><b>2) Вложенные множества</b></p>

<p><b>3) Вложенные скобки</b></p>

<p><b>4) Уступчатый список</b></p>

Ответ: 1, 2, 3, 4

Комментарий: AP0021058

<p><b> В случае удаления существующего узла отсортированного дерева поиска...</b></p>

<p><b>1) Дерево останется отсортированным</b></p>

<p><b>2) Дерево перестанет быть отсортированным</b></p>

<p><b>3) Неизвестно наверняка, останется ли дерево отсортированным</b></p>

Ответ: 1

Комментарий: AP0021059

<p><b> Какой из пунктов не обязателен для бинарного отсортированного дерева поиска?</b></p>

<p><b>1) Каждый правый узел дерева должен быть больше родителя</b></p>

<p><b>2) Каждый левый узел дерева должен быть меньше родителя</b></p>

<p><b>3) Каждый узел должен хранить указатель на двух потомков и своего родителя</b></p>

<p><b>4) Все пункты обязательны</b></p>

Ответ: 3

Комментарий: AP0021060

## 22 тема

<p><b>Что такое хэш таблица?</b></p>

1. <p><b>Структура данных, которая позволяет хранить пары (ключ, значение) и выполнять три операции: добавления новой пары, поиска и удаления пары по ключу</b></p>

2. <p><b>Структура, имеющая следующие свойства:

– существует единственный элемент (узел, вершина), на который не ссылается никакой другой и который называется корнем;

– начиная с корня и следуя по определенной цепочке указателей, можно осуществить доступ к любому элементу струк-туры;

– на каждый элемент, кроме корня, имеется единственная ссылка.</b></p>

3. <p><b>Представляет собой бинарное дерево, для которого выполняется основное свойство кучи: приоритет каждой вершины больше приоритетов её потомков</b></p>

4. <p><b>Одномерная структура данных, для которых загрузка или извлечение элементов осуществляется с помощью указателей начала извлечения (head) и конца (tail) очереди в соответствии с правилом FIFO</b></p>

Ответ: 1

Комментарий: AP0022001

<p><b>Что такое коллизия?</b></p>

1. <p><b> Ситуация, когда для различных ключей получается одно и то же значение хеш-функции</b></p>

2. <p><b> Переменная, содержащая адрес объекта</b></p>

3. <p><b> Поименованная совокупность данных разных типов, размещаемых в одной и той же обла-сти памяти, размер которой достаточен для хранения наибольшего элемента</b></p>

4. <p><b> Пользовательский тип данных, совокупность логически связанных данных различного типа, объединенных под одним идентификатором</b></p>

Ответ: 1

Комментарий: AP0022002

<p><b> Что такое хэширование?</b></p>

1. <p><b> Процесс преобраз больш кол-ва данных с помощью спец функций в уник запись фиксир длины</b></p>

2. <p><b> Конструкция,при которой функция вызывает сама себя.</b></p>

3. <p><b> Процесс упорядочения набора элементов в возрастающем или убывающем порядке</b></p>

4. <p><b> Линейная последовательность переменных, каждая из которых связана указателями со своими соседями</b></p>

Ответ: 1

Комментарий: AP0022003

<p><b> Какое основное отличие хэш таблиц от других динамических множеств?</b></p>

1. <p><b> Вычисление адреса элемента по значению ключа с помощью хеш-функции</b></p>

2. <p><b> Включение и исключение элементов которого осуществляется с помощью указателя стека в соответствии с правилом "по-следним введен, первым выведен" (last-in, first-out : LIFO).</b></p>

3. <p><b> Загрузка или извлечение элементов осуществляется с помощью указателей начала извлечения (head) и конца (tail) очере-ди в соответствии с правилом FIFO</b></p>

4. <p><b> Каждый элемент имеет указатели на следующий и на предыдущий элементы.</b></p>

Ответ: 1

Комментарий: AP0022004

<p><b> Что такое хэш-таблица с цепочками?</b></p>

1. <p><b> Каждая ячейка массива является указателем на связный список (цепочку) пар ключ-значение, соответствующих одному и тому же хеш-значению ключа</b></p>

2. <p><b> Хеш-таблица представляет собой некоторый массив, элементы которого есть пары</b></p>

3. <p><b> Каждый элемент имеет указатель на следующий</b></p>

4. <p><b> Каждый элемент имеет указатель на следующий и на предыдущий</b></p>

Ответ: 1

Комментарий: AP0022005

<p><b>Что такое хэш-таблица с открытой адресацией?</b></p>

1. <p><b> Каждая ячейка массива является указателем на связный список (цепочку) пар ключ-значение, соответствующих одному и тому же хеш-значению ключа</b></p>

2. <p><b> Хеш-таблица представляет собой некоторый массив, элементы которого есть пары</b></p>

3. <p><b> Каждый элемент имеет указатель на следующий</b></p>

4. <p><b> Каждый элемент имеет указатель на следующий и на предыдущий</b></p>

Ответ: 2

Комментарий: AP0022006

<p><b> Что такое хэш-функция?</b></p>

1. <p><b> Это функция c такой организацией работы, при которой она вызывает сама себя</b></p>

2. <p><b> Функция, осуществляющая преобразование массива входных данных произвольной длины в (выходную) битовую строку установленной длины, выполняемое определённым алгоритмом</b></p>

3. <p><b> Объявление функции, не содержащее тела функции</b></p>

4. <p><b> Функция для чтения из файла</b></p>

Ответ: 2

Комментарий: AP0022007

<p><b> Чем характерно закрытое хэширование?</b></p>

1. <p><b> Основная идея бвзовой структуры при открытом (внешнем) потенциальное множество разбивается на конечное кол-во классов, для каждого класса (0, В-1) строится хеш-функция.</b></p>

2. <p><b> Каждый элемент имеет указатель на следующий</b></p>

3. <p><b> Каждый элемент имеет указатель на следующий и на предыдущий</b></p>

4. <p><b> Попытка размещения данных в свободном вдресном пространстве или уже в удаленных записях при абсолютном дополнении хеш таблицы добавление новых данных исключенно</b></p>

Ответ: 4

Комментарий: AP0022008

<p><b> Чем характерно открытое хэширование?</b></p>

1. <p><b> Попытка размещения данных в свободном вдресном пространстве или уже в удаленных записях при абсолютном дополнении хеш таблицы добавление новых данных исключенно</b></p>

2. <p><b> Основная идея бвзовой структуры при открытом (внешнем) потенциальное множество разбивается на конечное кол-во классов, для каждого класса (0, В-1) строится хеш-функция.</b></p>

3. <p><b> Каждый элемент имеет указатель на следующий</b></p>

4. <p><b> Каждый элемент имеет указатель на следующий и на предыдущий</b></p>

Ответ: 2

Комментарий: AP0022009

<p><b> Выберете метод разрешения коллизии:</b></p>

1. <p><b> Метод деления</b></p>

2. <p><b> Метод умножения</b></p>

3. <p><b> Метод цепочек</b></p>

4. <p><b> Золотое сечение</b></p>

Ответ: 3

Комментарий: AP0022010

<p><b> Основное преимущество битовой карты:</b></p>

1. <p><b> Включение и исключение элементов которого осуществляется с помощью указателя стека в соответствии с правилом "по-следним введен, первым выведен" (last-in, first-out : LIFO).</b></p>

2. <p><b> Загрузка или извлечение элементов осуществляется с помощью указателей начала извлечения (head) и конца (tail) очере-ди в соответствии с правилом FIFO</b></p>

3. <p><b> Очень простая и эффективная организация обработки сложных запросов, которые могут объединять значения ключей различными логическими предикатами.</b></p>

4. <p><b> Каждый элемент имеет указатели на следующий и на предыдущий элементы.</b></p>

Ответ: 3

Комментарий: AP0022011

<p><b> Что такое линейное опробование:</b></p>

1. <p><b> Шаг перебора элементов не линейно зависит от номера попытки найти свободный элемент </b></p>

2. <p><b> Основана на нелинейной адресации, достигаемой за счет суммирования значений основной и дополнительной хеш-функций</b></p>

3. <p><b> Последовательный перебор элементов таблицы с некоторым фиксированным шагом</b></p>

4. <p><b> Нет верного ответа. </b></p>

Ответ: 3

Комментарий: AP0022012

<p><b> Что такое квадратичное опробование:</b></p>

1. <p><b> Шаг перебора элементов не линейно зависит от номера попытки найти свободный элемент </b></p>

2. <p><b> Основана на нелинейной адресации, достигаемой за счет суммирования значений основной и дополнительной хеш-функций</b></p>

3. <p><b> Последовательный перебор элементов таблицы с некоторым фиксированным шагом</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022013

<p><b> Что такое двойное хеширование:</b></p>

1. <p><b> Шаг перебора элементов не линейно зависит от номера попытки найти свободный элемент </b></p>

2. <p><b> Основана на нелинейной адресации, достигаемой за счет суммирования значений основной и дополнительной хеш-функций</b></p>

3. <p><b> Последовательный перебор элементов таблицы с некоторым фиксированным шагом</b></p>

4. <p><b> Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022014

<p><b> Существенный недостаток инвентированных индексов:</b></p>

1. <p><b> Очень простая и эффективная организация обработки сложных запросов, которые могут объединять значения ключей различными логическими предикатами.</b></p>

2. <p><b> Большие затраты времени на составление вспомогательной структуры данных и ее обновление</b></p>

3. <p><b> Загрузка или извлечение элементов осуществляется с помощью указателей начала извлечения (head) и конца (tail) очере-ди в соответствии с правилом FIFO</b></p>

4. <p><b> Каждый элемент имеет указатели на следующий и на предыдущий элементы.</b></p>

Ответ: 2

Комментарий: AP0022015

<p><b> Основное применения хэширования:</b></p>

1. <p><b> Криптография</b></p>

2. <p><b> Вычисление факториала</b></p>

3. <p><b> Написание кода</b></p>

4. <p><b> Реализация бинарных деревьев</b></p>

Ответ: 1

Комментарий: AP0022016

<p><b> Условие хорошего хэш-алгоритма:</b></p>

1. <p><b> Одни и те же данные должны давать всегда один и тот же хэш.</b></p>

2. <p><b> Разные данные должны давать разный хэш.</b></p>

3. <p><b> Оба варианта верны</b></p>

4. <p><b> Оба неверны</b></p>

Ответ: 3

Комментарий: AP0022017

<p><b> Как ведет себя хорошая хэш-функция:</b></p>

1. <p><b> Весь доступный диапазон хэшей используется по максимуму. То есть, если на хэш отведено 32 байта, то разные данные дают максимально разнообразный хэш, который может являться совершенно любой комбинацией битов. То есть, диапазон хэшей не "простаивает".</b></p>

2. <p><b> Даже небольшое изменение входных данных (даже изменение 1 бита входных данных) должно давать другой хэш. Не должно быть такого, что небольшие изменения дают тот же самый хэш. Тот же самый хэш должен возникать в результате какого-то совершенно другого набора данных, чтобы вероятность случайного присутствия двух таких данных (дающих одинаковый хэш) была минимальной.</b></p>

3. <p><b> Оба варианта верны </b></p>

4. <p><b> Оба неверны </b></p>

Ответ: 3

Комментарий: AP0022018

<p><b> Ситуация, когда для различных ключей получается одно и то же значение хеш-функции это: </b></p>

1. <p><b> Коллизия </b></p>

2. <p><b> Рекурсия </b></p>

3. <p><b> Хеширование </b></p>

4. <p><b> Цикл </b></p>

Ответ: 1

Комментарий: AP0022019

<p><b> Структура данных, которая позволяет хранить пары (ключ, значение) и выполнять три операции: добавления новой пары, поиска и удаления пары по ключу это: </b></p>

1. <p><b> Хеш-таблица </b></p>

2. <p><b> Бинарное дерево </b></p>

3. <p><b> Стек </b></p>

4. <p><b> Очередь </b></p>

Ответ: 1

Комментарий: AP0022020

<p><b> Процесс преобразования большого количества данных с помощью специальных функций в уникальную запись фиксированной длины это: </b></p>

1. <p><b> Хеширование </b></p>

2. <p><b> Рекурсия </b></p>

3. <p><b> Коллизия </b></p>

4. <p><b> Цикл </b></p>

Ответ: 1

Комментарий: AP0022021

<p><b> Каждая ячейка массива является указателем на связный список

(цепочку) пар ключ-значение, соответствующих одному и тому же хеш-значению ключа это: </b></p>

1. <p><b> Хэш-таблица с цепочками </b></p>

2. <p><b> Хэш-таблица с открытой адресацией </b></p>

3. <p><b> Односвзянный список </b></p>

4. <p><b> Двухсвзянный список </b></p>

Ответ: 1

Комментарий: AP0022022

<p><b> Хеш-таблица представляет собой некоторый массив, элементы которого есть пары это:</b></p>

1. <p><b> Хэш-таблица с цепочками</b></p>

2. <p><b> Хэш-таблица с открытой адресацией</b></p>

3. <p><b> Односвзянный список</b></p>

4. <p><b> Двухсвзянный список</b></p>

Ответ: 2

Комментарий: AP0022023

<p><b> Функция, осуществляющая преобразование массива входных данных произвольной длины в (выходную) битовую строку установленной длины, выполняемое определённым алгоритмом это:</b></p>

1. <p><b> Хэш-функция</b></p>

2. <p><b> Рекурсия</b></p>

3. <p><b> Прототип функции</b></p>

4. <p><b> Шаблонная функция</b></p>

Ответ: 1

Комментарий: AP0022024

<p><b> Попытка размещения данных в свободном вдресном пространстве или уже в удаленных записях при абсолютном дополнении хеш таблицы добавление новых данных исключенно это характерно для:</b></p>

1. <p><b> Закрытое хеширование</b></p>

2. <p><b> Открытое хеширование</b></p>

3. <p><b> Односвязнный список</b></p>

4. <p><b> Двухсвзяный список</b></p>

Ответ: 1

Комментарий: AP0022025

<p><b> Основная идея бвзовой структуры при открытом (внешнем) потенциальное множество разбивается на конечное кол-во классов, для каждого класса (0, В-1) строится хеш-функция это характерно для:</b></p>

1. <p><b> Закрытое хеширование</b></p>

2. <p><b> Открытое хеширование</b></p>

3. <p><b> Односвязнный список</b></p>

4. <p><b> Двухсвзяный список</b></p>

Ответ: 2

Комментарий: AP0022026

<p><b>Метод цепочек это:</b></p>

1. <p><b>Метод разрешения коллизии</b></p>

2. <p><b>Метод для хэш-функции</b></p>

3. <p><b>Алгортим для реализации бинарвых деревьев</b></p>

4. <p><b>Алгортим для реализации бинарвых куч</b></p>

Ответ: 1

Комментарий: AP0022027

<p><b>Хеш-таблица это?</b></p>

1. <p><b>Динамическое множество, использующее спеациальные методы адресации</b></p>

2. <p><b>Список</b></p>

3. <p><b>Сетевая структура данных</b></p>

4. <p><b>Матрица</b></p>

Ответ: 1

Комментарий: AP0022028

<p><b>Коллизия – это ситуация, которая возникает когда</b></p>

1. <p><b>Одинаковым значениям ключей хеш-функция генерирует одинаковые адреса</b></p>

2. <p><b>Разным значениям ключей хеш-функция генерирует одинаковые адреса</b></p>

3. <p><b>Разным значениям ключей хеш-функция генерирует разные адреса</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022029

<p><b>Функция H(key) = [hashTableSize(key\*A mod 1)] используется при: </b></p>

1. <p><b>Мультипликативном методе хеширования</b></p>

2. <p><b>Аддитивном методе хеширования</b></p>

3. <p><b>Модульном хешировании</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022030

<p><b>Функция h(key, i) = (h1(key) + i\*h2(key))mod hashTableSize используется при</b></p>

1. <p><b>Двойном хешировании</b></p>

2. <p><b>Универсальном хешированиии</b></p>

3. <p><b>Аддитивном хешировании</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022031

<p><b>Коэффициент заполнения хеш-таблицы – это: </b></p>

1. <p><b>число зранимых элементов n, деленное на размер массива hashTableSiz</b></p>

2. <p><b>Число возможных значений хеш-функции</b></p>

3. <p><b>Количество ухлов в каждом списке таблицы</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022032

<p><b> Преобразование входного массива данных произвольной длины в выходную битовую строку фиксированной длины носит название:</b></p>

1. <p><b> Интерпретация </b></p>

2. <p><b> Хеширование </b></p>

3. <p><b> Модуляция </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022033

<p><b>Результат работы функции свёртки носит название: </b></p>

1. <p><b>Модуль </b></p>

2. <p><b>Контейнер </b></p>

3. <p><b>Хеш</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 3

Комментарий: AP0022034

<p><b>К характеристикам алгоритмов хеширования следует отнести:</b></p>

1. <p><b>Вычислительная сложность </b></p>

2. <p><b>Разрядность</b></p>

3. <p><b>Криптостойкость </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1, 2, 3

Комментарий: AP0022035

<p><b>К простейшим примерам хеш-функций следует отнести:</b></p>

1. <p><b>Контрольную сумму </b></p>

2. <p><b>Массив идентификаторов </b></p>

3. <p><b>Ориентированный граф </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022036

<p><b>Множество массивов данных, дающих одинаковые хеш-коды, носят название:</b></p>

1. <p><b>Cегменты</b></p>

2. <p><b>Коллизии </b></p>

3. <p><b>Итераторы</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022037

<p><b>Обычная разрядность контрольных сумм составляет:</b></p>

1. <p><b>32 бита </b></p>

2. <p><b>64 бита </b></p>

3. <p><b>128 бит </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022038

<p><b>Простейшим способом усложнения поиска коллизий является: </b></p>

1. <p><b>Изменение типизации данных </b></p>

2. <p><b>Увеличение разрядности хеша </b></p>

3. <p><b>Изменение адресации памяти </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022039

<p><b>В каких структурах данных используются хеш-функции?</b></p>

1. <p><b>хеш-таблицы </b></p>

2. <p><b>декартовы деревья </b></p>

3. <p><b>массивы коллизий </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1, 2

Комментарий: AP0022040

<p><b>Хеширование применяется: </b></p>

1. <p><b>Для сверки данных</b></p>

2. <p><b>Для проверки на наличие ошибок </b></p>

3. <p><b>Для проверки парольной фразы </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1, 2, 3

Комментарий: AP0022041

<p><b>К вариантам адресации в хеш-таблицах следует отнести: </b></p>

1. <p><b>Прямую </b></p>

2. <p><b>Открытую</b></p>

3. <p><b>Терминальную</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1, 2

Комментарий: AP0022042

<p><b>Ситуация в хеш-таблице, когда для различных ключей получается одно и то же хэш-значение, называется: </b></p>

1. <p><b>рекурсией </b></p>

2. <p><b>коллизией </b></p>

3. <p><b>сегрегацией </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022043

<p><b>Число хранимых элементов хеш-таблицы делённое на число возможных значений хэш-функции называется: </b></p>

1. <p><b>Коэффициентом аддитивности хеш-таблицы </b></p>

2. <p><b>Коэффициентом заполнения хеш-таблицы </b></p>

3. <p><b>Коэффициентом маркировки хеш-таблицы</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022044

<p><b>Среднее время выполнения операций в хеш-таблице зависит: </b></p>

1. <p><b>От типа данных</b></p>

2. <p><b>От коэффициента заполнения таблицы </b></p>

3. <p><b>От маркировки идентификаторов</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022045

<p><b>Мерой криптостойкости хеш-функции является: </b></p>

1. <p><b>Вычислительная сложность нахождения коллизии </b></p>

2. <p><b>Степень свободы хеш-функции </b></p>

3. <p><b>Типизация данных хеш-функции</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022046

<p><b>Какая хеш-функция по определению не имеет коллизии?</b></p>

1. <p><b>Когнитивная </b></p>

2. <p><b>Инъективная </b></p>

3. <p><b>Модификативная </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022047

<p><b>Вычислительная невозможность нахождения исходного блока данных по известному значению хеш-функции от этого блока носит название: </b></p>

1. <p><b>Априорность </b></p>

2. <p><b>Необратимость </b></p>

3. <p><b>Вариативность</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022048

<p><b>Для устранения коллизий хеш-функций используют: </b></p>

1. <p><b>Метод цепочек</b></p>

2. <p><b>Метод взаимосвязей </b></p>

3. <p><b>Метод корреляции</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1

Комментарий: AP0022049

<p><b>Какие из приведенных ниже методов используются для устранения коллизий хеш-функций?</b></p>

1. <p><b>Маркированная адресация</b></p>

2. <p><b>Открытая адресация </b></p>

3. <p><b>Модульная адресация </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 2

Комментарий: AP0022050

<p><b>Из приведенных ниже записей выделите методы устранения коллизий хеш-функций:</b></p>

1. <p><b>Метод цепочек </b></p>

2. <p><b>Строгая типизация </b></p>

3. <p><b>Открытая адресация </b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 1, 3

Комментарий: AP0022051

<p><b>С чего начинается выполнение операция в хеш-таблице?</b></p>

1. <p><b>С добавления данных</b></p>

2. <p><b>С вычисления хеш-функции от ключа</b></p>

3. <p><b>С заполнения таблицы</b></p>

4. <p><b>С объявления данных</b></p>

Ответ: 2

Комментарий: AP0022052

<p><b>Каким свойствам должны соответствовать хеш-таблицы?</b></p>

1. <p><b>Выполнение операции в хеш-тиблице начинается с вычисления хеш-функции от ключа.</b></p>

2. <p><b>Механизм разрешения коллизий является важной составляющей любой хеш-таблицы</b></p>

3. <p><b>Количество хранимых элементов массива, деленное на число возможных значений хеш-функции, называется коэффицентом заполнения хеш-таблицы и является важным параметром, от которого зависит среднее время выполнения операций.</b></p>

4. <p><b>Все ответы верны.</b></p>

Ответ: 4

Комментарий: AP0022053

<p><b>Хеш-таблица это:</b></p>

1. <p><b>Тип данных</b></p>

2. <p><b>Функция</b></p>

3. <p><b>Обычный массив с необычной адресацией</b></p>

4. <p><b>Нет верного ответа.</b></p>

Ответ: 3

Комментарий: AP0022054

<p><b>Для чего можно использовать хеш-таблицы?</b></p>

1. <p><b>Для баз данных телефонных номеров. </b></p>

2. <p><b>Для хранения паролей пользователей. </b></p>

3. <p><b>Для поиска информации о водители лишь по его номеру в водительском удостоверении. </b></p>

4. <p><b>Все ответы верны.</b></p>

Ответ: 4

Комментарий: AP0022055

## 23 тема

<p><b> Для сортировки вставками характерно следующее: </b></p>

1. <p><b> перебор элементов в неотсортированной части массива и дальнейшее перемещение каждого элемента в отсортированную часть массива на нужное место</b></p>
2. <p><b> происходит сравнение соседних элементов и обмен местами, если следующий элемент меньше предыдущего</b></p>
3. <p><b> перебор элементов в неотсортированной части массива и обмен местами максимального и последнего элементов</b></p>

Ответ: 1

Комментарий: AP0023001

<p><b> Сортировка выбором происходит следующим образом:</b></p>

1. <p><b> на первом этапе в массиве выбирается некоторый опорный элемент. На втором этапе все элементы, которые больше опорного перемещаются вправо, а те, которые меньше – влево</b></p>
2. <p><b> перебор элементов в неотсортированной части массива и обмен местами максимального и последнего элементов. Сортировка продолжается, пока неотсортированная часть массива не уменьшится до одного элемента</b></p>
3. <p><b> на первом этапе массив разбивается на две равные части. Затем происходит их рекурсивная сортировка и слияние</b></p>

Ответ: 2

Комментарий: AP0023002

<p><b> Выберите верные утверждения для сортировки пузырьком </b></p>

1. <p><b> Сложность алгоритма O(n\*n)</b></p>

2. <p><b> Сложность алгоритма O(n\*log(n))</ </b></p>

3. <p><b> Cоседние элементы некоторой последовательности чисел сравниваются между собой </b></p>

4. <p><b> Весь массив в процессе сортировки делится на две части: упорядоченную и неупорядоченную </b></p>

Ответ: 1, 3

Комментарий: AP0023003

<p><b> Выберите верные утверждения для сортировки Хоара</b></p>

1. <p><b> Сложность алгоритма O(n\*n)</b></p>

2. <p><b> Сложность алгоритма O(n\*log(n))</ </b></p>

3. <p><b> Cоседние элементы некоторой последовательности чисел сравниваются между собой </b></p>

4. <p><b> Массив делится на две части относительно некоторого значения, называемого *медианой*. </b></p>

Ответ: 2, 4

Комментарий: AP0023004

<p><b> Быстрая сортировка Хоара не является рекурсивным алгоритмом</b></p>

1. <p><b> Да </b></p>

2. <p><b> Нет </b></p>

Ответ: 2

Комментарий: AP0023005

<p><b> Что описывает Big-O</b></p>

1. <p><b> Направление функции </b></p>

2. <p><b> Насколько быстро работает алгоритм </b></p>

3. <p><b> Происхождение алгоритма </b></p>

4. <p><b> Имя алгоритма </b></p>

Ответ: 2

Комментарий: AP0023006

<p><b> Выберите верные утверждения для сортировки методом простой вставки</b></p>

1. <p><b> Сложность алгоритма O(n\*n)</b></p>

2. <p><b> Сложность алгоритма O(n\*log(n))</ </b></p>

3. <p><b> Весь массив в процессе сортировки делится на две части: упорядоченную и неупорядоченную. </b></p>

4. <p><b> Массив делится на две части относительно некоторого значения, называемого *медианой*. </b></p>

Ответ: 1, 3

Комментарий: AP0023007

**<p><b>Какой метод сортировки требует выделения дополнительной памяти?</b></p>**

1. <p><b>сортировка выбором</b></p>

2. <p><b>сортировка слиянием</b></p>

3. <p><b>сортировка пузырьком</b></p>

4. <p><b>сортировка вставками</b></p>

Ответ: 2

Комментарий: AP0023008

**<p><b> Сортировка будет идти по массиву слева направо. Если текущий элемент больше следующего, меняем их местами. Делаем так, пока массив не будет отсортирован. </b></p>**

1. <p><b>Сортировка пузырьком/ Bubble sort </b></p>

2. <p><b>Сортировка вставками/ Insertion sort</b></p>

3. <p><b>Сортировка выбором / Selection sort </b></p>

Ответ: 1

Комментарий: AP0023009

**<p><b> Создается массив, в котором после завершения алгоритма будет лежать ответ. Будем поочередно вставлять элементы из исходного массива так, чтобы элементы в массиве-ответе всегда были отсортированы. </b></p>**

1. <p><b>Сортировка пузырьком/ Bubble sort </b></p>

2. <p><b>Сортировка вставками/ Insertion sort</b></p>

3. <p><b>Сортировка выбором / Selection sort</b></p>

Ответ: 2

Комментарий: AP0023010

**<p><b> На очередной итерации будет находиться минимум в массиве после текущего элемента и менять его с ним, если надо. Таким образом, после i-ой итерации первые i элементов будут стоять на своих местах. </b></p>**

1. <p><b>Сортировка пузырьком/ Bubble sort </b></p>

2. <p><b>Сортировка вставками/ Insertion sort</b></p>

3. <p><b>Сортировка выбором / Selection sort</b></p>

Ответ: 3

Комментарий: AP0023011

**<p><b>Массив делится пополам, рекурсивно отсортироуваются части, после чего выполняется процедура слияния: поддерживается два указателя, один на текущий элемент первой части, второй – на текущий элемент второй части. Из этих двух элементов выбирается минимальный, вставляется в ответ и сдвигаемтсяуказатель, соответствующий минимуму. </b></p>**

1. <p><b>Сортировка слиянием / Merge sort</b></p>

2. <p><b>Сортировка подсчетом / Counting sort</b></p>

3. <p><b>Быстрая сортировка / Quicksort</b></p>

Ответ: 1

Комментарий: AP0023012

**<p><b> Создадается массив размера r – l, где l – минимальный, а r – максимальный элемент массива. После этого проходимся по массиву и подсчитаем количество вхождений каждого элемента. Теперь можно пройти по массиву значений и выписать каждое число столько раз, сколько нужно. </b></p>**

1. <p><b>Сортировка слиянием / Merge sort</b></p>

2. <p><b>Сортировка подсчетом / Counting sort</b></p>

3. <p><b>Быстрая сортировка / Quicksort</b></p>

Ответ: 2

Комментарий: AP0023013

**<p><b> Выбрается элемент из массива(опорный). Потом происходит разбиение: перераспределение элементов в массиве таким образом, что элементы меньше опорного помещаются перед ним, а больше или равные после. Рекурсивно применить первые два шага к двум подмассивам слева и справа от опорного элемента. </b></p>**

1. <p><b>Сортировка слиянием / Merge sort</b></p>

2. <p><b>Сортировка подсчетом / Counting sort</b></p>

3. <p><b>Быстрая сортировка / Quicksort</b></p>

Ответ: 3

Комментарий: AP0023014

**<p><b>Алгоритм сортировки, в котором используется диапазон чисел сортируемого массива (списка) для подсчёта совпадающих элементов </b></p>**

1. <p><b>Поразрядная сортировка / Radix sort</b></p>

2. <p><b>Пирамидальная сортировка / Heapsort</b></p>

3. <p><b>Сортировка подсчетом / Counting sort</b></p>

Ответ: 3

Комментарий: AP0023015

**<p><b>Исходно предназначен для сортировки целых чисел, записанных цифрами.** **Но так как в памяти компьютеров любая информация записывается целыми числами, алгоритм пригоден для сортировки любых объектов, запись которых можно поделить на «разряды», содержащие сравнимые значения.</b></p>**

1. <p><b>Поразрядная сортировка / Radix sort</b></p>

2. <p><b>Пирамидальная сортировка / Heapsort</b></p>

3. <p><b>Сортировка подсчетом / Counting sort</b></p>

Ответ: 1

Комментарий: AP0023016

**<p><b> Сортировка использует бинарное сортирующее дерево. Каждый лист имеет глубину либо d, либо d-1, d — максимальная глубина дерева.**

**Значение в любой вершине не меньше (другой вариант — не больше) значения её потомков. </b></p>**

1. <p><b>Поразрядная сортировка / Radix sort</b></p>

2. <p><b>Пирамидальная сортировка / Heapsort</b></p>

3. <p><b>Сортировка подсчетом / Counting sort</b></p>

Ответ: 2

Комментарий: AP0023017

**<p><b> Это те операторы, которые применяются к двум операндам (слева и справа)</b></p>**

1. <p><b>Арифметические операторы присваивания</b></p>

2. <p><b>Унарные арифметические операторы</b></p>

3. <p><b>Бинарные арифметические операторы</b></p>

Ответ: 3

Комментарий: AP0023018

**<p><b> Это те операторы, которые применяются только к одному операнду</b></p>**

1. <p><b>Арифметические операторы присваивания</b></p>

2. <p><b>Унарные арифметические операторы</b></p>

3. <p><b>Бинарные арифметические операторы</b></p>

Ответ: 2

Комментарий: AP0023019

<p><b>Определение сложности алгоритма сортировки является</b></p>

1. <p><b>нотация</b></p>

2. <p><b>итерация</b></p>

3. <p><b>спецификация алгоритма</b></p>

Ответ: 1

Комментарий: AP023020

<p><b>Параметры основной оценки алгоритма сортировки</b></p>

1. <p><b>время</b></p>

2. <p><b>память</b></p>

3. <p><b>длина кода</b></p>

4. <p><b>устойчивость</b></p>

Ответ: 1, 2, 4

Комментарий: AP023021

<p><b>Почему алгоритмы быстрых сортировок не дают большого выигрыша при малых размерах массивов?</b></p>

1. <p><b>предварительная подготовка к сортировке</b></p>

2. <p><b>сложность выполнения</b></p>

3. <p><b>меньшее количество операций</b></p>

4. <p><b>больший размер кода</b></p>

Ответ: 1

Комментарий: AP023022

<p><b>Чтобы алгоритм бинарного поиска работал правильно, нужно, чтобы массив (список) был</b></p>

1. <p><b>отсортированным </b></p>

2. <p><b>несортированным </b></p>

3. <p><b>в куче </b></p>

Ответ: 1

Комментарий: AP023023

<p><b>Какой алгоритм из нижеперечисленных будет самым производительным, если дан уже отсортированный массив?</b></p>

1. <p><b>сортировка слиянием </b></p>

2. <p><b>сортировка вставками </b></p>

3. <p><b>быстрая сортировка </b></p>

4. <p><b>пирамидальная сортировка </b></p>

Ответ: 2

Комментарий: AP023024

<p><b>На какой сортировке основана сортировка Шелла?</b></p>

1. <p><b>вставками</b></p>

2. <p><b>перестановками </b></p>

3. <p><b>отбором</b></p>

4. <p><b>Хоара</b></p>

Ответ: 1

Комментарий: AP023025

<p><b>Какая сортировка является самой неэффективной из представленных?</b></p>

1. <p><b>отбором</b></p>

2. <p><b>Шелла</b></p>

3. <p><b>пузырьковая</b></p>

4. <p><b>вставками</b></p>

Ответ: 2

Комментарий: AP023026