**<p><b>Что необходимо указывать при объявлении структуры дерева? <p><b>**

1. <p><b>потомки </b></p>
2. <p><b>число ветвей</b></p>
3. <p><b>тип данных потомков</b></p>
4. <p><b>указатель на предка</b></p>

Ответ: 1

Комментарий: SD001031

**<p><b>Способ преобразования блока данных в число – это</b></p>**

1. <p><b>Индексирование</b></p>
2. <p><b>Хеширование</b></p>
3. <p><b>Маркировка</b></p>
4. <p><b>Сортировка</b></p>

Ответ: 2

Комментарий: SD001032

**<p><b>Как называется узел дерева, не имеющий потомков? </b></p>**

1. <p><b>последний </b></p>
2. <p><b>терминальный </b></p>
3. <p><b>окончательный </b></p>
4. <p><b>финальный</b></p>

Ответ: 2

Комментарий: SD001033

**<p><b>У всех узлов левого поддерева произвольного узла X бинарного дерева поиска значения ключей данных ПРОПУСК значения ключа данных самого узла X. Заполните пропуск</b></p>**

1. <p><b>больше </b></p>
2. <p><b>меньше </b></p>
3. <p><b>больше или равно </b></p>
4. <p><b>меньше или равно</b></p>

Ответ: 2

Комментарий: SD001034

**<p><b>Какое максимальное количество узлов-потомков у каждого узла кучи? </b></p>**

1. <p><b>2</b></p>
2. <p><b>8</b></p>
3. <p><b>64</b></p>
4. <p><b>неограниченное</b></p>

Ответ: 4

Комментарий: SD001035

**<p><b>С помощью какого типа данных обычно реализуется куча? </b></p>**

1. <p><b>дерево</b></p>
2. <p><b>класс</b></p>
3. <p><b>динамический массив</b></p>
4. <p><b>стек</b></p>

Ответ: 1

Комментарий: SD001036

**<p><b>Из скольки объектов состоит элемент словаря? </b></p>**

1. <p><b>1</b></p>
2. <p><b>2</b></p>
3. <p><b>4</b></p>
4. <p><b>число неограниченно</b></p>

Ответ: 2

Комментарий: SD001037

**<p><b>Может ли соответствовать одно и то же значение двум разным ключам словаря? </b></p>**

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

Ответ: 1

Комментарий: SD001038

**<p><b>Дано пустое множество mySet. Что выведет данная программа?**

**mySet.insert('I'); <br>**

**mySet.insert('n'); <br>**

**mySet.insert('f'); <br>**

**mySet.insert('i'); <br>**

**mySet.insert('n'); <br>**

**copy( mySet.begin(), mySet.end(), ostream\_iterator<char>(cout, " "));<br></b></p>**

1. <p><b>I n f i n </b></p>
2. <p><b>I f i n</b></p>
3. <p><b>f i n</b></p>
4. <p><b>I f i n n</b></p>

Ответ: 2

Комментарий: SD001039

**<p><b>Может ли множество содержать дубликаты? </b></p>**

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

Ответ: 2

Комментарий: SD001040

**<p><b>Чему равно количество строк матрицы смежности графа? </b></p>**

1. <p><b>Количеству рёбер графа</b></p>
2. <p><b>Количеству вершин графа</b></p>
3. <p><b>Максимальной степени входа вершины графа</b></p>
4. <p><b>Минимальной степени входа вершины графа</b></p>

Ответ: 2

Комментарий: SD001041

**<p><b>В чём отличие очереди от дека? </b></p>**

1. <p><b>В очередь можно только добавлять элементы</b></p>
2. <p><b>В очереди с каждого конца можно либо только удалять, либо только добавлять элементы</b></p>
3. <p><b>В дек можно добавлять/удалять элементы только с одного конца</b></p>
4. <p><b>Они предназначены для разных типов данных</b></p>

Ответ: 2

Комментарий: SD001042

**<p><b>Почему сложных типов данных нет в стандартных библиотеках? </b></p>**

1. <p><b>они занимают слишком много места, это усложняет подключение</b></p>
2. <p><b>нет универсального решения, проще написать самому</b></p>
3. <p><b>их реализация требует подключения дополнительных библиотек с необходимыми функциями</b></p>
4. <p><b>нет единого мнения, как они должны быть реализованы</b></p>

Ответ: 2

Комментарий: SD001043

**<p><b>Тип данных, который удобно использовать при необходимости добавлять и удалять элементы в середине</b></p>**

1. <p><b>дек</b></p>
2. <p><b>одномерный массив</b></p>
3. <p><b>связанный список</b></p>
4. <p><b>очередь</b></p>

Ответ: 3

Комментарий: SD001044

**<p><b>Какой тип данных самый универсальный? </b></p>**

1. <p><b>одномерный массив</b></p>
2. <p><b>граф</b></p>
3. <p><b>динамический массив</b></p>
4. <p><b>куча</b></p>

Ответ: 3

Комментарий: SD001045

**<p><b>С помощью какого типа данных лучше всего решать численные задачи типа «Ханойская башня»? </b></p>**

1. <p><b>очередь</b></p>
2. <p><b>стек</b></p>
3. <p><b>динамический массив</b></p>
4. <p><b>дек</b></p>

Ответ: 2

Комментарий: SD001046

**<p><b>Сколько способов выделения памяти для строк существует? </b></p>**

1. <p><b>1</b></p>
2. <p><b>2</b></p>
3. <p><b>3</b></p>
4. <p><b>4</b></p>

Ответ: 2

Комментарий: SD002001

**<p><b>Возможно ли реализовать кучу при помощи одномерного массива? </b></p>**

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

Ответ: 1

Комментарий: SD001047

**<p><b>Что выведет данная программа?**

**string Numbers[] { "Один", "Два", "Три" };<br>**

**cout << "\nArray of Numbers:" << ::endl; <br>**

**for (int i = 0; i < 4; i++)<br>**

**cout << Numbers[i] << endl; <br></b></p>**

1. <p><b>Один Два Три</b></p>
2. <p><b>Один<br>

Два<br>

Три<br></b></p>

1. <p><b>Три<br>

Два<br>

Один<br></b></p>

1. <p><b>Ошибку сборки</b></p>

Ответ: 2

Комментарий: SD002002

**<p><b>Что выведет данная программа?**

**for (int i = 0; i < 7; i++)<br>**

**AS[i] = (char)('1' + i); <br>**

**cout << "\nArray AS:" << ::endl; <br>**

**for (int i = 0; i < n; i++)<br>**

**cout << "AS[" << i << "] = " << AS[i].c\_str() << endl; <br></b></p>**

1. <p><b>0

1<br>

2<br>

3<br>

4<br>

5<br>

6<br>

7<br></b></p>

1. <p><b>1<br>

2<br>

3<br>

4<br>

5<br>

6<br>

7<br>

8<br></b></p>

1. <p><b>1 2 3 4 5 6 7 8</b></p>
2. <p><b>Ошибку сборки</b></p>

Ответ: 1

Комментарий: SD002003

**<p><b>Пустой указатель – это</b></p>**

1. <p><b>NULL</b></p>
2. <p><b>nullptr</b></p>
3. <p><b>null</b></p>
4. <p><b>\_ptr</b></p>

Ответ: 2

Комментарий: SD001048

**<p><b>Как выделяется память под массив строк в данной программе?**

**AS = new string[n]; <br>**

**for (int i = 0; i < n; i++)<br>**

**AS[i] = (char)('1' + i); <br>**

**cout << "\nArray AS:" << ::endl; <br>**

**for (int i = 0; i < n; i++)<br>**

**cout << "AS[" << i << "] = " << AS[i].c\_str() << endl; <br>**

**delete[] AS; <br></b></p>**

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

Ответ: 2

Комментарий: SD002004

**<p><b>Сколько времени будет выполняться данная программа?**

**while (s != "");<br>**

**cout << "\nArray AS is as follows:\n";<br>**

**for (int i = 0; i < count; i++)<br>**

**cout << "AS[" << i << "] = " << AS[i] << endl;<br></b></p>**

1. <p><b>пока не будет нажат Enter</b></p>
2. <p><b>пока не будет введена пустая строка</b></p>
3. <p><b>бесконечно</b></p>
4. <p><b>программа не запустится</b></p>

Ответ: 2

Комментарий: SD002005

**<p><b>Как в памяти хранится связный список? </b></p>**

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

Ответ: 2

Комментарий: SD001049

**<p><b>Укажите неверное утверждение для полного бинарного дерева</b></p>**

1. <p><b>у каждой вершины не более двух потомков</b></p>
2. <p><b>заполнение вершин идёт сверху вниз</b></p>
3. <p><b>в пределах одного уровня заполнение идёт справа налево</b></p>
4. <p><b>является кучей</b></p>

Ответ: 3

Комментарий: SD001050

**<p><b>Какой индекс будет иметь левый потомок вершины i двоичной кучи, хранимой в виде одномерного массива? </b></p>**

1. <p><b>i + 1</b></p>
2. <p><b>i + 2</b></p>
3. <p><b>2 \* i + 1</b></p>
4. <p><b>2 \* i + 2</b></p>

Ответ: 3

Комментарий: SD001051

**<p><b>Где может храниться самый большой элемент двоичной кучи? </b></p>**

1. <p><b>на левом краю нижнего ряда</b></p>
2. <p><b>на правом краю нижнего ряда</b></p>
3. <p><b>на верхнем ряду</b></p>
4. <p><b>в любом месте</b></p>

Ответ: 3

Комментарий: SD001052

**<p><b>При добавлении в какую часть динамического массива он показывает наилучшую производительность? </b></p>**

1. <p><b>начало</b></p>
2. <p><b>середина</b></p>
3. <p><b>конец</b></p>
4. <p><b>не имеет значения</b></p>

Ответ: 3

Комментарий: SD001053

**<p><b>Что делает данная программа?**

**while (ptr\_p) {**

**cout << p->val << " ";**

**p = p->next; } </b></p>**

1. <p><b>Выводит количество элементов списка</b></p>
2. <p><b>Проверяет, нет ли в списке пустых значений</b></p>
3. <p><b>Выводит на экран весь список</b></p>
4. <p><b>Очищает список</b></p>

Ответ: 3

Комментарий: SD001054

**<p><b>Структура FIFO – это…</b></p>**

1. <p><b>очередь</b></p>
2. <p><b>стек</b></p>
3. <p><b>дек</b></p>
4. <p><b>динамический массив</b></p>

Ответ: 1

Комментарий: SD001055

**<p><b>Как называется граф, каждой вершине которого поставлено в соответствие конкретное числовое значение</b></p>**

1. <p><b>Числовой</b></p>
2. <p><b>Определённый</b></p>
3. <p><b>Взвешенный</b></p>
4. <p><b>Нормальный</b></p>

Ответ: 3

Комментарий: SD001056

**<p><b>Как называется граф, каждой вершине которого поставлено в соответствие конкретное числовое значение</b></p>**

1. <p><b>Матрица смежности</b></p>
2. <p><b>Матрица инцидентности</b></p>
3. <p><b>Дерево</b></p>
4. <p><b>Список ребер</b></p>

Ответ: 1, 2, 4

Комментарий: SD001057

**<p><b>Какое минимальное количество путей между каждой парой вершин существует в связном графе?</b></p>**

1. <p><b>0</b></p>
2. <p><b>1</b></p>
3. <p><b>2</b></p>
4. <p><b>3</b></p>

Ответ: 2

Комментарий: SD001058

**<p><b>Можно ли применять алгоритм поиска в ширину для неориентированного графа?</b></p>**

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

Ответ: 1

Комментарий: SD001059

**<p><b>Какой тип данных эффективнее всего использовать для реализации поиска в ширину в графе?</b></p>**

1. <p><b>Стек</b></p>
2. <p><b>Дерево</b></p>
3. <p><b>Динамический массив</b></p>
4. <p><b>Очередь</b></p>

Ответ: 4

Комментарий: SD001060