Bangladesh Desh Santir Kotha

chatgpt.com/c/6753107f-497c-8006-a5a1-2afce96d9a16

The name of our country is Bangladesh.

Yes, the name of your country is Bangladesh, located in South Asia. It's known for its rich culture, history, and natural beauty, including the Sundarbans and the Padma River. Let me know if you’d like any specific information about Bangladesh!

You have to implement a playlist feature in NF. You can remove elements from the middle. You can go to the end and come back to the beginning. You can add a song and rearrange them in any order you like. What data structure would be best for the scenario? a. Array b. Queue c. Link list d. Stacked What is the purpose of the having clause in SQL? a. To update data in a table. b. To retrieve data from multiple tables. c. To perform aggregate calculations on data. d. To filter the result of a query based on aggregate calculations. Which of the following statements is false? a. For the singly linked list, the last node called the tail which points to null b. It contains a pointer that points to the previous node. c. Time complexity of insertion is o. d. It only travels in one direction. a. A graph with all vertices having equal degrees is known as a regular graph b. Complete graph c. Multi graph d. Simple graph Imagine you have an employee table and you have garbage data in that table. Now you want to clean the whole table. Which of the following operations will you use in SQL? a. Truncate b. Delete c. Destroy d. Rough Assume that we use bubble short to short an distinct element in an ascending order. When does the best case of bubble short occur? a. When elements are not shorted by any order. b. When elements are shorted in decreasing order. c. There is no best case for bubble short. It always takes big of n squared time. d. When elements are shorted in ascending order. If you subtract c from a number and divide the result by 2, you get 12. What is the number? a. Trish. b. Ataro. c. Subtrish. d. Subbish. Which of the following algorithm design technique is used in mass short? a. Debating control. b. Dynamic programming. c. Gradient method. d. Base tracking. Answer will give in short.

1. **Best data structure for the playlist:** c. **Linked list**
2. **Purpose of `HAVING` clause in SQL:** d. **To filter the result of a query based on aggregate calculations**
3. **False statement about singly linked list:** b. **It contains a pointer that points to the previous node.**
4. **Graph with all vertices having equal degrees:** a. **Regular graph**
5. **Operation to clean an entire SQL table:** a. **Truncate**
6. **Best case for bubble sort:** d. **When elements are sorted in ascending order**
7. **Number in the equation:** d. **Subbish** (The number is 26, calculated as `(12 * 2) + c`)
8. **Algorithm design technique for merge sort:** b. **Divide and conquer**

If you subtract C from a number and divide the result by 2, you get 12. What is the number? 30B18C36D24 The quiz of the following option is not true about the binary search tree. A. None of the above. B. The value of the left side should be less than the root node. C. The left and right subtrees should also be a binary search tree. D. The value of the right side should be greater than the root node. Q. Which two features of the object-oriented programming are relevant? A. Abstraction and polymorphism. B. Encapsulation and polymorphism. C. Encapsulation and abstraction. D. Inheritance and encapsulation. Q. Alip has 4 times as many as Zim and Zim has 3 times as many as C. Altogether they have 192. How many has is? A. Alip 166, Zim 130, Slide 14 B. Alip 166, Zim 130, Slide 12 A. Alip 166, Zim 130, Seed 12 C. Alip 166, Zim 130, Seed 12 D. Class Z inherits class Y and Y has inherit class X. Q. Then while creating the object of class Z, what will be the sequence of constructor getting calls? A. Constructor of Z, then Y, finally of X. B. Constructor of X, then Z, finally of Y. C. Constructor of Z, then X, finally Y. D. Constructor of X, then Y, finally Z. Q. What is the way to access a method of a class without creating an object of the class? A. None of the above. B. Using static method. C. By initiating the class. D. By calling the method directly. Q. If you have 59 cube blocks, what is the minimum number that needs to be taken away in order to construct a solid cube with none left over? A. 32. B. 34. C. 33. D. 35. Q. The Agile software development model is built based on? A. Incremental development. B. Both incremental and iterative development. C. Linear development. D. Iterative development. Q. Almost every e-commerce website has a search. A user can add item to the search, and an item can be added into the search by multiple users. Q. What kind of relationship is best suited for implementing a search solution in a software? A. Many-to-one. B. One-to-one. C. Many-to-many. D. One-to-many. Q. Observe being fast to a copy constructor. A. Must be fast by value. B. Must be fast by reference. C. Must be fast with integer type. D. Must not be mentioned in parameter list. Answer given in short.

1. **What is the number?**
   C. **36**
2. **False statement about binary search tree:**
   A. **None of the above**
3. **Object-oriented programming features:**
   C. **Encapsulation and abstraction**
4. **Alip, Zim, and total 192 distribution:**
   B. **Alip 144, Zim 36, C 12**
5. **Constructor call sequence in class Z:**
   D. **Constructor of X, then Y, finally Z**
6. **Accessing a class method without creating an object:**
   B. **Using static method**
7. **Minimum blocks to remove for a solid cube:**
   A. **32**
8. **Agile software development model:**
   B. **Both incremental and iterative development**
9. **Best relationship for search solution:**
   C. **Many-to-many**
10. **Fast copy constructor requirement:**
    B. **Must be passed by reference**

You need to find the shortest path between two nodes in an unweighted graph. A. Use BFS. B. Use A star source. C. Use Digest algorithm. D. BFS. Which one of the following is a functional requirement? A. Portability. B. Reliability. C. Maintainability. D. Business needs. Which of the following statements are correct for a max heap? A. The root always contains the smallest key. B. All leaves are located on the same level. C. All keys in the left subtree are always smaller than any key in the corresponding right subtree. D. Each subtree is also a max heap. Mr. Joy is implementing traffic light solution for his hometown, which will repeatedly show three different colors in sequence. Red, yellow, green, red, yellow. Which data structure is most appropriate for this type of task? A. Circular queue. B. Struct seeding list. Which word in bracket is most opposite to the word in capitals? A. Sleek. B. Sordid. C. Unimaginative. D. Disorderly. D. Disheveled. A. Disorderly. B. Sordid. C. Disheveled. D. Oblique. A pointer to the base class can hold the address of a only direct class object. B. Base class object as well as direct class object. C. None of the above. D. Only base class object. If there are more than one topological shorting of directed acyclic graph is possible, which of the following is true? A. Many Hamiltonian paths are possible. B. Given information is insufficient to comment anything. C. Exactly one Hamiltonian path is possible. D. No Hamiltonian path is possible. Which of the following is the best way to achieve both abstraction and code reusability in OOV? A. Using global variable. B. Using abstract class. C. Using inheritance. D. Using functional overloading. Which of the following prototype does not associate with phototizing model? A. Vertical prototype. B. Horizontal prototype. C. Domain prototype. D. Diagonal prototype. Which feature of OOV indicate code reusability? A. Inheritance. B. Encapsulation. C. Abstraction. D. Polymorphism. Let's say you have multiple classes where the same code in right hand multiple times. Meaning code duplication is present. Which OOV principle will help you to reduce this issue? A. Polymorphism. B. Encapsulation. C. Abstraction. D. Inheritance. What is the difference between an inner join and a left join in SQL? An inner join only returns rows where there is a mesh in both tables. While a left join returns all rows from the left table and mesh in rows from the right table. B. An inner join only returns unique way from both tables. While a left join returns all rows from both tables. C. A left join only returns rows where there is a mesh in both tables. While an inner join returns all rows from the left table and mesh in rows from the right table. D. A left join only returns unique rows from both tables. While an inner join returns all rows from both tables. Which SQL clause is used to filter the result of a query? A. Group by B. Haven. C. Order by D. Hoyer. In a family of six members A, B, C, D, E, and F, there are two married couples. A is married to B. C is married to D. A is the mother of E and F. Who is the father of E and F? A. C. B. D. C. B. D. A. Look at this series. 3, 6, 4, 7, 5, 8 What number should come next? A. 9. B. 6. C. 7. D. 10. Answer give in short.

1. **Shortest path in an unweighted graph:**
   A. **Use BFS**
2. **Functional requirement:**
   D. **Business needs**
3. **Correct statement about max heap:**
   D. **Each subtree is also a max heap**
4. **Data structure for traffic light:**
   A. **Circular queue**
5. **Most opposite to "DISHEVELED":**
   A. **Sleek**
6. **Pointer to base class can hold:**
   B. **Base class object as well as derived class object**
7. **More than one topological sorting in a DAG:**
   A. **Many Hamiltonian paths are possible**
8. **Best way for abstraction and code reusability in OOP:**
   C. **Using inheritance**
9. **Prototype not associated with prototyping model:**
   D. **Diagonal prototype**
10. **OOP feature indicating code reusability:**
    A. **Inheritance**
11. **OOP principle to reduce code duplication:**
    D. **Inheritance**
12. **Difference between inner join and left join:**
    A. **An inner join only returns rows where there is a match in both tables. A left join returns all rows from the left table and matching rows from the right table.**
13. **SQL clause to filter query results:**
    B. **HAVING**
14. **Father of E and F:**
    C. **B**

15. **Next number in the series (3, 6, 4, 7, 5, 8):**
    A. **9**

What will be the output of the following code? In mean integer i equal to 1 while i++ less than or equal to 1 semicolon while i++ less than or equal to 2 semicolon, printf i equal to person d,sh is to i, return 0. What will be the output of the following code?

To analyze the given code snippet:

c

```c
#include <stdio.h>
int main() {
    int i = 1;
    while (i++ <= 1);
    while (i++ <= 2);
    printf("i = %d\n", i);
    return 0;
}
```

Code Breakdown:

---

1. `int i = 1;`

   `i` is initialized to 1.
2. `while (i++ <= 1);`

   The condition is checked:
    - Initially, `i = 1`, so the condition `1 <= 1` is `true`.
    - Post-increment makes `i = 2`.
    - The loop ends because of the semicolon (`;`) after the `while`.
3. `while (i++ <= 2);`

   Now, `i = 2`.
    - First iteration: `2 <= 2` is `true`, and post-increment makes `i = 3`.
    - Second iteration: `3 <= 2` is `false`, so the loop terminates.

4. `printf("i = %d\n", i);`

   At this point, `i = 3`, so the output will be:

Output:

---

css

```
i = 3
```

Only An's

Output:
`i = 3`

#include <stdio.h> int main() { int i = 1; while (i++ <= 1); while (i++ <= 2); printf("i = %d\n", i); return 0; } Ans 2,1,84?

Let's analyze the provided code step by step:

```c
#include <stdio.h>
int main() {
    int i = 1;
    while (i++ <= 1);
    while (i++ <= 2);
    printf("i = %d\n", i);
    return 0;
}
```

Step-by-Step Execution:

---

1. **Initialization:** `int i = 1;`

   `i` is initialized to 1.
2. **First `while` loop:** `while (i++ <= 1);`
   - In the first iteration:
     - `i = 1`, the condition `1 <= 1` is `true`.
     - Post-increment happens, so `i = 2`.
   - The loop terminates because the semicolon (`;`) ends the loop body.

   After this loop: `i = 2`.

3. **Second `while` loop:** `while (i++ <= 2);`
   - In the first iteration:
     - `i = 2`, the condition `2 <= 2` is `true`.
     - Post-increment happens, so `i = 3`.
   - In the next iteration:

     `i = 3`, the condition `3 <= 2` is `false`.
   - The loop terminates.

   After this loop: `i = 3`.

4. **`printf` Statement:** `printf("i = %d\n", i);`

   `i = 3` at this point.

Output:

---

css

```
i = 3
```

The answer provided (`2, 1, 84`) does not align with the actual logic of the program. The correct output is `i = 3`.

#include <stdio.h> int main() { int i = 1; while (i++ <= 1); while (i++ <= 2); printf("i = %d\n", i); return 0; Output 2,1,84?

The provided code will not produce the output **2, 1, 84**. Instead, let's carefully walk through the execution and determine the **actual output**:

Code Analysis

c

```
#include <stdio.h>
int main() {
    int i = 1;              // Initialize i to 1
    while (i++ <= 1);       // Loop runs as long as i <= 1
    while (i++ <= 2);       // Loop runs as long as i <= 2
    printf("i = %d\n", i);  // Print the value of i
    return 0;
}
```

Step-by-Step Execution

1. **Initialization:**
   `int i = 1;`

   Variable `i` is initialized to 1.

2. **First `while` Loop:**
   `while (i++ <= 1);`

   - First iteration: `i = 1`, condition `1 <= 1` is `true`. Post-increment happens, so `i` becomes 2.
   - Second iteration: `i = 2`, condition `2 <= 1` is `false`. Loop terminates.

   After this loop: `i = 2`.

3. **Second `while` Loop:**
   `while (i++ <= 2);`

   - First iteration: `i = 2`, condition `2 <= 2` is `true`. Post-increment happens, so `i` becomes 3.
   - Second iteration: `i = 3`, condition `3 <= 2` is `false`. Loop terminates.

   After this loop: `i = 3`.

4. **`printf` Statement:**
   `printf("i = %d\n", i);`

   At this point, `i = 3`, so the output will be:

Actual Output:

```css
i = 3
```

Explanation for "2, 1, 84" Misunderstanding

The output **2, 1, 84** might come from a **different program** or if multiple `printf` statements are involved, but it **does not apply** to the code provided here. This code produces a single output: `i = 3`.

What will the following program print? int array equal to 10, 20, 13, 14, 15 int strf equal to array cout strf++ strf and line return 0 a, 10, 20, 20, 10, 30, 10, 20, 20

Let's break down the provided code step by step to understand what it prints:

Code:

```cpp
#include <iostream>
using namespace std;

int main() {
    int array[] = {10, 20, 13, 14, 15}; // Initialize array with values
    int* strf = array; // Pointer strf points to the first element of the array
    cout << strf++ << " " << *strf << endl; // Output value and post-increment pointer strf
    return 0;
}
```

Step-by-Step Execution:

1. **Array Initialization:**

   ```cpp
   int array[] = {10, 20, 13, 14, 15};
   ```

   The array `array` contains the elements: `[10, 20, 13, 14, 15]`.

2. **Pointer Initialization:**

   ```cpp
   int* strf = array;
   ```

   The pointer `strf` is initialized to point to the first element of the array, i.e., `strf` points to `array[0]` (which is 10).

3. **First `cout` statement:**

   ```cpp
   cout << strf++ << " " << *strf << endl;
   ```

   Let's break this down:

   - `strf++` is a **post-increment** operation. This means that the value at `strf` (which is `array[0]` or 10) will be printed first, and then `strf` will be incremented to point to the next element in the array (`array[1]` or 20).
   - After `strf++`, `strf` now points to `array[1]` (which is 20).
   - `*strf` dereferences `strf` to access the value at `array[1]`, which is 20.

   So, the output of this statement will be:

   ```
   10 20
   ```

Final Output:

```
10 20
```

The code **does not** produce the output `10, 20, 20, 10, 30, 10, 20, 20`. It only prints the values `10` and `20`, as explained.

> #include <iostream> using namespace std; int main() { int array[] = {10, 20, 30,40,50}; // Initialize array with values int* strf = array; // Pointer strf points to the first element of the array cout << strf++ << " " << *strf << endl; // Output value and post-increment pointer strf return 0; } Only output giy

The output of the given program is:

```
10 20
```

Explanation:

1. `strf++` prints the value `strf` is pointing to before the increment. Initially, `strf` points to `array[0]`, which is `10`.
2. After the post-increment, `strf` now points to `array[1]`, which is `20`.
3. `*strf` dereferences the pointer `strf` and prints the value at `array[1]`, which is `20`.

Thus, the output is `10 20`.

What will be the output of the following recursive function? void function int n if n == 0 return cout n function n-1 cout n void function int n-1 cout n return 0 if n == 0

Let's analyze the recursive function provided, step by step:

Code:

cpp

```
#include <iostream>
using namespace std;

void function(int n) {
    if (n == 0) return;
    function(n - 1);  // Recursive call
    cout << n << " ";  // Print the current value of n
}

int main() {
    function(3);  // Call the recursive function with n = 3
    return 0;
}
```

Step-by-Step Execution:

The function is called with `n = 3`. Let's go through the recursive calls:

1. **Call `function(3)`**
    - `n != 0`, so it proceeds to call `function(2)`.
2. **Call `function(2)`**
    - `n != 0`, so it proceeds to call `function(1)`.
3. **Call `function(1)`**
    - `n != 0`, so it proceeds to call `function(0)`.
4. **Call `function(0)`**
    - `n == 0`, so it returns without doing anything.

Returning from Recursive Calls:

Now, as the recursion unwinds:

1. **Return from `function(1)`** (after calling `function(0)`):

   `cout << 1 << " ";` prints `1`.
2. **Return from `function(2)`** (after calling `function(1)`):

   `cout << 2 << " ";` prints `2`.
3. **Return from `function(3)`** (after calling `function(2)`):

   `cout << 3 << " ";` prints `3`.

Final Output:

```
1 2 3
```

The function prints the numbers in ascending order as the recursion unwinds.

> #include <iostream> using namespace std; void function(int n) { if (n == 0) return; cout << n << " "; function(n - 1); // Recursive call cout << n << " "; // Print the current of n } int main() { function(3); // Call the recursive function with n = 3 return 0; }

Let's analyze the provided recursive function step by step:

Code:

cpp

```
#include <iostream>
using namespace std;

void function(int n) {
    if (n == 0) return;
    cout << n << " ";  // Print the current value of n before the 
recursive call
    function(n - 1);    // Recursive call with n-1
    cout << n << " ";   // Print the current value of n after the 
recursive call
}

int main() {
    function(3);  // Call the recursive function with n = 3
    return 0;
}
```

Step-by-Step Execution:

---

1. **Call `function(3)`:**
    - Print `3` (because `n` is 3).
    - Call `function(2)`.
2. **Call `function(2)`:**
    - Print `2` (because `n` is 2).
    - Call `function(1)`.
3. **Call `function(1)`:**
    - Print `1` (because `n` is 1).
    - Call `function(0)`.
4. **Call `function(0)`:**

    Since `n == 0`, the function returns immediately without printing anything.

Unwinding the Recursive Calls:

---

1. **Return from `function(1)`:**

    After the call to `function(0)` returns, print `1` again (because of the second `cout` statement in the function).

2. **Return from `function(2)`:**

    After the call to `function(1)` returns, print `2` again (because of the second `cout` statement in the function).

3. **Return from `function(3)`:**

    After the call to `function(2)` returns, print `3` again (because of the second `cout` statement in the function).

Final Output:

```
3 2 1 1 2 3
```

Explanation:

- The function prints the value of `n` before the recursive call, then recursively calls itself with `n-1`, and after the recursive call finishes, it prints the value of `n` again.
- This results in the numbers being printed in a "descending" order during the recursion and in an "ascending" order during the return phase.

What is the time complexity of the following program? Integer count sum on 0, total sum on n. While total greater than 0, count plus or equal to total. Total divided by equal to 2. A log n, B n log n, C n square, D n by 2.

The given program description can be summarized as follows:

```cpp
int count = 0, total = n;
while (total > 0) {
    count += total;
    total /= 2;
}
```

Step-by-Step Analysis:

1. **Initial Setup**:
    - `count` is initialized to 0.
    - `total` is initialized to `n`.
2. **While Loop Condition**:
    - The loop runs as long as `total > 0`.
    - In each iteration of the loop:
        - `count` is incremented by the current value of `total`.
        - `total` is halved (i.e., `total /= 2`).
3. **Iterations of the While Loop**:
    - Initially, `total = n`.
    - In the next iteration, `total = n / 2`.
    - Then `total = n / 4`, and so on, halving the value of `total` at each step.
4. **How Many Times Does the Loop Execute?**:
    - The loop continues halving `total` until `total` becomes 0.
    - This means the number of iterations is approximately `log2(n)` (because you keep dividing by 2).

Time Complexity:

- The loop runs approximately `log2(n)` times because `total` is halved in each iteration.
- Therefore, the time complexity is **O(log n)**.

Correct Answer: A) log n

Which access modifier can be used with a base class member to inherit that in the direct class, a public B and C, C-protected, D-private? What is the primary objective of the planning phase in the software development life cycle? a. To create the system architecture b. To define the scope, schedule, and resource for the project c. To conduct testing activities d. To gather requirements What does a relational database management system organize data to manage data in more than one file at once? a. Tuples b. Both A and B c. Tables d. Relations c. Fellows us a. Contain information on a node b. Both B and C c. Find the smallest fellow in an array d. Find the largest fellow in an array What is the best suitable answer? a. Fetal is to flower b. Cat is to dog c. Fan is to paper d. Engine is to car e. Ball is to game What is the purpose of ACID properties in database management system? a. To optimize storage efficiency b. To enforce referential integrity c. To improve query performance d. To guarantee the reliability of data What is the normalization of a database? a. The process of organizing data in a table b. The process of removing redundant data from a table c. The process of optimizing data in a table d. The process of adding redundant data to a table Who holds the responsibility for writing a new test? a. Customer b. Project manager c. Users d. Developer Arrange the word below in the alphabetic order Answer give in short

1. **Access modifier for base class member to inherit in derived class:**

   Answer: C) Protected
2. **Primary objective of the planning phase in SDLC:**

   Answer: B) To define the scope, schedule, and resources for the project
3. **What does a relational database management system organize data to manage in multiple files at once?**

   Answer: C) Tables
4. **Which is the correct pair analogy?**

   Answer: E) Ball is to game
5. **Purpose of ACID properties in database management system:**

   Answer: D) To guarantee the reliability of data
6. **Normalization of a database:**

   Answer: B) The process of removing redundant data from a table
7. **Who holds the responsibility for writing a new test?**

   Answer: D) Developer

8. **Arrange the words in alphabetical order:**

   (This would depend on the provided words, which are not listed in the question.)

Let me know if you need further clarification!

ChatGPT can make mistakes. Check important info.