<u>**Answer-1:**</u>

1st iteration:

7 2 13 2 11 4 -> 2 7 13 2 11 4

2 **7 13** 2 11 4 -> 2 7 13 2 11 4

2 7 **13 2** 11 4 -> 2 7 2 13 11 4

2 7 2 **13 11** 4 -> 2 7 2 11 13 4

2 7 2 11 **13 4** -> 2 7 2 11 4 13

2nd iteration:

2 7 2 11 4 13 -> 2 7 2 11 4 13

2 **7 2** 11 4 13 -> 2 2 7 11 4 13

2 2 **7 11** 4 13 -> 2 2 7 11 4 13

2 2 7 **11 4** 13 -> 2 2 7 4 11 13

3rd iteration:

2 2 7 4 11 13 -> 2 2 7 4 11 13

2 **2 7** 4 11 13 -> 2 2 7 4 11 13

2 2 **7 4** 11 13 -> 2 2 4 7 11 13

4th iteration:

2 2 4 7 11 13 -> 2 2 4 7 11 13

2 **2 4** 7 11 13 -> 2 2 4 7 11 13

2 2 **4 7** 11 13 -> 2 2 4 7 11 13

2 2 4 **7 11** 13 -> 2 2 4 7 11 13

2 2 4 7 **11 13** -> 2 2 4 7 11 13 (all elements are sorted)

Answer-2

Array	Vector
1. Arrays can be implemented in a static or dynamic way.	1. Vectors can only be implemented dynamically.
2. Arrays have a fixed size.	2. Vectors have a dynamic size i.e. they can resize themselves.

Answer-3

The time complexity of the given code segment is O(n^2).

The outer loop runs for n iterations, and the inner loop runs for n iterations for each iteration of the outer loop. Therefore, the total number of iterations for both loops is n * n means n^2.

The built-in function builtin_popcount(i) has a constant time complexity, meaning it takes the same amount of time to execute regardless of the size of the input.

Therefore, the time complexity of the code segment is determined by the time complexity of the loops, which is O(n^2).

Answer-4

Yes this code has a flaw. In the answer counting loop when i=0 according to the given code it will compare a[0] with a[0-1]. So a[0-1] doesn't exist. It's called undefined behavior, that's means this code can run or it will show some test cases wrong. To avoid this we have to add a condition. That is given below:

```
#include<bits/stdc++.h>
using namespace std;
int main(){
        int n;
        cin>>n;
        vector<int>a(n);
        for(int i=0;i<n;i++)
                cin>>a[i];
        sort(a.begin(),a.end());
        int ans = 0;
        for(int i=0 ; i<=n ; i++)
                //here is that condition
                if(i==0)
                        {
                         ans++;
                         continue;
                        }
                if(a[i]!=a[i-1])
                        ans++;
        cout<<ans;
        return 0;
}
```

Answer-5

The time complexity of the above code segment is O(n^2).

The outer loop runs n times and the inner loop runs for n/i times. Since the inner loop variable j increases by a multiple of i on each iteration, it will run for a maximum of n/i iterations before it exceeds n.

The inner loop will run for a maximum of n/1 + n/2 + n/3 + ... + n/n iterations, which is equal to n * (1/1 + 1/2 + 1/3 + ... + 1/n).

This series can be simplified to n * (1 + 1/2 + 1/3 + ... + 1/n), which is approximately equal to n * ln(n).

Thus, the overall time complexity of the code segment is O(n * ln(n)).

The space complexity of the code segment is O(n).
The vector d is an array of vectors of size n, and each element of the vector can store an integer, so the space complexity is O(n).

Answer-6

Name	Accessibility from own class	Accessibility from derived class	Accessibility from world

Public	YES	YES	YES
Private	YES	NO	NO
Protected	YES	YES	NO

Answer-7

In C++, the ‘new’ and ‘delete’ operators are used to dynamically allocate and deallocate memory at runtime.

delete: The delete operator is used to deallocate the memory that was previously allocated by new. It takes a pointer to the memory location as an argument.

new: The new operator is used to allocate memory for a new object. It takes a data type as an argument and returns a pointer to the memory address where the new object has been created.

So it's very important that we make sure of deallocation when we dynamically allocate memory space. Otherwise, our code will become slow, it will take more space etc.

Answer-8

Yes, I agree with Bob. In the worst case, the algorithm will take approximately 31.6 years to finish.

The time complexity of the algorithm is O(n^3), which means that the number of instructions executed by the algorithm will be proportional to n^3.
If n is equal to 10^6, then the number of instructions executed will be equal to (10^6)^3 = 1,000,000,000,000,000 = 1 quadrillion.

Since Alice's computer can run 10^9 instructions in 1 second, it will take approximately 1 quadrillion / (10^9 instructions/second) = 1,000,000,000 seconds = approximately 11574 days = approximately 31.6 years to finish.

Answer-9

Binary Search	Linear Search

1. It's necessary to sort the array before searching the element.	1. It's not necessary to sort the array before searching the element.
2. Time complexity is O(log2N)	2. Time complexity is O(N)

Answer-10

Yes, there is a flaw in the function as written: it dynamically allocates memory for a new integer object using the new operator, but it does not deallocate the memory when it is no longer needed. This will result in a memory leak, as the dynamically allocated memory will not be released back to the system.

To fix this flaw, you can modify the function as follows:

```
void func()
{
   int* p = new int;

   delete p;  // Deallocating the memory for the integer object
   return;
}
```

By calling delete p at the end of the function, the dynamically allocated memory will be deallocated and released back to the system, preventing a memory leak.