**Sample questions ET**

Q1. You have given two sorted array you need to merge them and create a new array in which all elements in given arrays must present in new array but in sorted.

**Input:**

First line contains two integers m and n which defines the length of two given sorted array

Second line contains the elements of first array

Third line contains the elements of second array

**Output:**

Print M+N elements in resultant array in sorted order

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 3 4  2 3 7  1 3 4 8 | 1 2 3 3 4 7 8 |

Q2. Given an array of integers. All numbers occur twice except one number which occurs once. Find the number in O(n) time & constant extra space.

**Input :**

First line contains a integer N which specify size if array

Second line contains N integer elements of array

**Output:**

Print a Integer which occurring only once in given array

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 7  1 3 2 3 4 4 2 | 1 |

Q3. Given an integer x, return true if x is a palindrome, and false otherwise.

**Input:**

First line contains an integer X

**Output:**

Print true or false

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 14341 | true |

Q.4

Implement a C++ program demonstrating polymorphism using a base class ‘Shape’ and its derived classes ‘Circle’ and ‘Rectangle’. The Shape class should have a virtual function ‘calculateArea()’ to calculate and display the area of the shape. Implement the derived classes with appropriate data members and member functions to calculate the area of a circle and rectangle respectively.

Write a program that creates objects of Circle and Rectangle classes and demonstrates polymorphism by calling the calculateArea() function on each object.

**Input:**

First line contains radius of circle

second line contain length and breadth of rectangle respectively

**Example Test case:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4.0  6.0 5.0 | Area of the circle: 50.24  Area of the rectangle: 30 |

Q5. Create a C++ class Matrix representing a 2x2 matrix. Implement operator overloading for multiplication operations for matrices. Provide a member function to display the matrix. Write appropriate test cases to demonstrate the functionality of the overloaded operators.

**Input:**

First line contains four integers which are elements of first 2x2 matrix

Second line contains four integers which are elements of second 2x2 matrix

**Example Test case:**

|  |  |
| --- | --- |
| **Input** | **Output** |
| 1 3 2 4  3 2 6 5 | Multiplication:  21 17  30 24 |

Q6. You're given an N x M matrix where each row and column is sorted in ascending order. Your task is to determine if a given number x is present in the matrix.

**Input Format:**

-The first line contains two space-separated integers, N and M, denoting the number of elements in a row and column, respectively.

-The second line of each test case contains N \* M space-separated integers representing the elements in the matrix in row-major order.

-The third line of each test case contains a single integer x, which is the element to be searched within the matrix.

**Output Format:**

Print 1 if the element is present in the matrix, else 0.

Constraints: 1 <= N,M <= 300 <= A[i] <= 100

|  |  |
| --- | --- |
| Input | Output |
| 3 3  3 30 38  44 52 54  57 60 69  62 | 0 |

Q.7 You are given N integers, your task is to sort the array in decreasing order using priority\_queue container.

Input Format:  
First Line: an integer N denoting the number of elements to sort

Second Line: N space separated integers.

Output Format:  
N integers sorted in decreasing order

Constraints:  
1 <= N <=1000

|  |  |
| --- | --- |
| Input | Output |
| 10  3 5 1 7 8 18 2 7 18 100 | 100 18 18 8 7 7 5 3 2 1 |

Q.8 You are given N integers, your task is to find kth largest element in the array

Input Format:  
First Line: an integer N denoting the number of elements in array and integer K determine Kth greatest element

Second Line: N space separated integers.

Output Format:  
Kth largest element

Constraints:  
1 <= N <=1000

|  |  |
| --- | --- |
| Input | Output |
| 10 3  3 5 1 7 8 18 2 7 18 100 | 18 |

Q 9. Given an array of **N** elements, where each element is at most K away from its target position, devise an algorithm that sorts in O(N log K) time.

Hint: Use priority queue

Input:

First line contains length of array n and k which represent the how much the array element away from its target position.

Second line contains the all n elements present in array

Output:

All elements of sorted array.

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 6 3  2 6 3 12 56 8 | 2 3 6 8 12 56 |

Q10. Find the frequency of all unique elements in vector with

Input:

First line contains the length of vector N.

Second line contains N elements of vector.

Output:

Frequency of every unique element in vector

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 10  1 2 1 4 5 3 3 2 1 4 | 3 2 2 2 1 |

Q11. Implement a C++ function combinationSum that takes in a vector of distinct integers candidates, a target integer target, and returns a vector containing all unique combinations of candidates where the chosen numbers sum up to the target. Each number in candidates can be used an unlimited number of times.

The function should avoid duplicate combinations. The order of combinations in the output vector does not matter.

Input:

First line contains the length of vector N and target T, which sum of few elements of vector

Second line contains N elements of vector.

Output:

All combination vectors whose elements make sum up equals to target

Example test case:

|  |  |
| --- | --- |
| Input | Output |
| 10 7  4 5 3 3 2 1 4 | 4 3  4 2 1  5 2  3 3 1  3 4  2 1 4 |

**What is the main principle of OOP?**

Inheritance Encapsulation Polymorphism Abstraction

**Which concept ensures that a class should not reveal its internal implementation?**

Inheritance Encapsulation

Polymorphism Abstraction

**Which OOP concept allows one class to inherit properties and behavior from another class?**

Abstraction Polymorphism

Inheritance Encapsulation

**In C++, a class's constructor is called when:**

A class object is declared A class object is created

A class is declared A class is defined

**What does the term 'polymorphism' mean in OOP?**

Ability to access multiple classes at once

Ability to write multiple methods within a class

Ability to have multiple forms

Ability to hide implementation details

**Which OOP feature allows methods with the same name but different parameters to be defined within the same class?**

Inheritance Encapsulation

Polymorphism Abstraction

**In OOP, a derived class inherits:**

A) Data members and member functions from the base class

B) Only data members from the base class

C) Only member functions from the base class

D) Constructors from the base class

**Which keyword in C++ is used to implement data hiding?**

this friend public private

**Which OOP principle focuses on reusing code by creating new classes based on existing ones?**

Inheritance Encapsulation Polymorphism Abstraction

**Which OOP principle ensures that a class only allows access to a limited set of members to the outside world?**

Abstraction Encapsulation Polymorphism Inheritance

**In OOP, what is an abstract class?**

A class with only private members

A class that cannot have objects instantiated from it

A class with no member functions A class with only static methods

**Which OOP feature allows a class to have multiple methods with the same name but different parameters?**

Inheritance Encapsulation Polymorphism Abstraction

**What is the concept of 'method overloading' in OOP?**

Redefining a method in a derived class

Creating methods with the same name but different parameters in the same class

Accessing methods from a base class in a derived class.

Creating private methods within a class.

**Which keyword is used to prevent a method from being overridden in a derived class in C++?**

override final virtual constant

**In OOP, what does the term 'composition' refer to?**

Creating objects from classes Inheriting properties from multiple classes

Including objects of other classes within a class Using static methods in a class

**What is the process of creating a new class from an existing class called in OOP?**

A) Cloning Creating Instantiating Inheriting

**In OOP, which access specifier allows the members of a class to be accessed from outside the class?**

public private protected friend

**What is the purpose of a destructor in a C++ class?**

To allocate memory for class objects To initialize class members

To deallocate memory and perform cleanup tasks when an object is destroyed

To define the number of objects that can be created from a class

**Which OOP concept allows a class to inherit from multiple base classes?**

A) Inheritance Encapsulation Polymorphism Multiple Inheritance

**Which feature of OOP allows a class to have more than one method with the same name but different parameters?**

Inheritance Encapsulation Polymorphism Overloading

**In OOP, what is an interface?**

A class that cannot be instantiated

A blueprint for a class that specifies a set of methods to be implemented

A class with private members only A class with static methods only

**Which keyword in C++ is used to indicate that a method in a base class is intended to be overridden in a derived class?**

override virtual final extends

**What is the purpose of the 'this' pointer in C++?**

It points to the base class in inheritance

It points to the derived class in inheritance

It points to the current object It points to the parent object

**Which OOP principle allows a class to inherit properties and behaviour from more than one class?**

Inheritance Encapsulation Polymorphism Multiple Inheritance

**What is the purpose of a pure virtual function in C++?**

It cannot be overridden in a derived class

It has no implementation in the base class and must be implemented in derived classes

It can be called directly without creating objects

It can only be accessed from the base class

**In OOP, what does the term 'dynamic binding' refer to?**

Resolving function calls at compile-time Resolving function calls at runtime

Binding methods to classes Binding objects to classes

**Which OOP principle emphasizes the ability of a class to take on multiple forms?**

A) Encapsulation Polymorphism Abstraction Inheritance

**What does the 'protected' access specifier in a C++ class signify?**

Members are accessible only within the class

Members are accessible within the class and derived classes

Members are accessible outside the class Members are accessible globally

**Which OOP concept involves using a single name to define different functions?**

Overloading Inheritance Encapsulation Polymorphism

**Recursion in programming refers to:**

A loop that repeats a block of code A function calling itself

Breaking down a problem into smaller subproblems

Declaring multiple functions with the same name

**The function that calls itself in a recursive process is known as the:**

Parent func Subsidiary func Recursive function Auxiliary func

**Which of the following is required for a recursive function?**

A loop structure A base case

An iterative statement A function prototype

**What is the base case in recursion?**

The case when the function has multiple return statements

The case when the function is called for the first time

The case that terminates the recursive process

The case where the function has the smallest input

**Which of the following problems is well-suited for solving with recursion?**

Calculating factorial Implementing a sorting algorithm

Parsing a string Matrix multiplication

**What is the process in which a function calls itself directly or indirectly is called?**

Self-calling Self-execution Recursive calling Function iteration

**Recursion can lead to:**

A) Efficient memory management Stack overflow

C) Improved performance Reduced program complexity

**The function that terminates the recursion process is referred to as the:**

Anchor func Starting func Base func Terminal funct

**What is an array in C++?**

A collection of different data types

A collection of variables of the same data type

A built-in function A class used for data manipulation

**The index of the first element in a C++ array is:**

1 0 -1 Depends on the size of the array

**How do you declare a one-dimensional array in C++?**

A) array myArray[]; int myArray[];

C) myArray[] = new array; int myArray[SIZE];

**What is a stack in C++?**

A data structure that follows FIFO (First-In-First-Out) order

A data structure that follows LIFO (Last-In-First-Out) order

C) A sorting algorithm

D) A data structure that arranges elements in ascending order

**Which operation adds an element to the top of a stack in C++?**

push( ) add( ) insert( ) append( )

**In a stack, removing an element from the top is done using which operation?**

pop( ) remove( ) delete( ) extract( )

**What does STL stand for in C++?**

A) System Template Library Standard Template Library

C) Structured Template Library Static Template Library

**Which header file is used to include vectors in C++ STL?**

A) <list> <vector> <map> <stack>

**Which STL container is used to implement a Last-In-First-Out (LIFO) data structure?**

set map stack queue

**In STL, which algorithm is used to find the maximum element in a container?**

find\_max( ) max\_element( ) maximum( ) find\_maximum( )

**What does the 'push\_back()' function do in a vector container of the STL?**

Adds an element to the front of the vector

Adds an element to the back of the vector

Removes an element from the front of the vector

Removes an element from the back of the vector

**In STL, which algorithm is used to sort elements in a container?**

order( ) sort( ) arrange( ) organize( )

**What does the 'pop()' function do in a stack container of the STL?**

Adds an element to the stack Removes the top element from the stack

Removes the bottom element from the stack

Retrieves the top element from the stack without removing it

**Which keyword is used to throw an exception explicitly in C++?**

catch throw try throws

**Which block is mandatory to accompany a 'try' block in C++ exception handling?**

catch throw finally exception

**What is the purpose of the 'catch' block in C++ exception handling?**

A) To throw exceptions

B) To define the block of code where an exception might occur

C) To handle exceptions thrown by the 'try' block

D) To finalize code execution

**How does the 'throw' statement work in C++?**

It terminates the program execution It generates an error message

It transfers control to the nearest 'catch' block It initializes an exception object

**Which standard header file provides the base class 'exception' for all standard C++ exceptions?**

<stdexcept> <iostream> <exception> <stdlib.h>

**Which of the following operators cannot be overloaded in C++?**

+ = \* ::

**How is the addition operator '+' overloaded for a custom class 'A' in C++?**

A operator+(A obj) operator+(A obj1, A obj2)

C) A operator+(A obj1, A obj2) A + operator(A obj)