Interview Questions List

Following are the list of top interview questions technologies for freshly IT graduates:

1- OOP
2- Database
3- Data Structures
4- Analytical Problems
5- Algorithms
7- Recursions

OOP

Following are frequently asked Interview Questions for freshers as well as an experienced .net/ Java/Python Software Developers.

1) What is OOPS?

OOPS is abbreviated as Object Oriented Programming system in which programs are considered as a collection of objects. Each object is nothing but an instance of a class.

2) Write basic concepts of OOPS?

Following are the concepts of OOPS:

1. Abstraction
2. Encapsulation
3. Inheritance
4. Polymorphism

3) What is a class?

A class is simply a representation of a type of object. It is the blueprint/plan/template that describes the details of an object.

4) What is an Object?

An object is an instance of a class. It has its own state, behavior, and identity.

5) What is Encapsulation?

Encapsulation is an attribute of an object, and it contains all data which is hidden. That hidden data can be restricted to the members of that class.

Levels are Public, Protected, Private, Internal, and Protected Internal.

6) What is Polymorphism?

Polymorphism is nothing but assigning behavior or value in a subclass to something that was already declared in the main class. Simply, polymorphism takes more than one form.

7) What is Inheritance?

Inheritance is a concept where one class shares the structure and behavior defined in another class. If Inheritance applied to one class is called Single Inheritance, and if it depends on multiple classes, then it is called multiple Inheritance.

8) What are manipulators?

Manipulators are the functions which can be used in conjunction with the insertion (<<) and extraction (>>) operators on an object. Examples are endl and setw.

9) Explain the term constructor

A constructor is a method used to initialize the state of an object, and it gets invoked at the time of object creation. Rules for constructor are:

- Constructor Name should be the same as a class name.
- A constructor must have no return type.

10) Define Destructor?

A destructor is a method which is automatically called when the object is made of scope or destroyed. Destructor name is also same as class name but with the tilde symbol before the name.

11) What is an Inline function?

An inline function is a technique used by the compilers and instructs to insert complete body of the function wherever that function is used in the program source code.

12) What is a virtual function?

A virtual function is a member function of a class, and its functionality can be overridden in its derived class. This function can be implemented by using a keyword called virtual, and it can be given during function declaration.

A virtual function can be declared using a token(virtual) in C++. It can be achieved in C/Python Language by using function pointers or pointers to function.

13) What is a friend function?

A friend function is a friend of a class that is allowed to access to Public, private, or protected data in that same class. If the function is defined outside the class cannot access such information.

A friend can be declared anywhere in the class declaration, and it cannot be affected by access control keywords like private, public, or protected.

14) What is function overloading?

Function overloading is a regular function, but it can perform different tasks. It allows the creation of several methods with the same name which differ from each other by the type of input and output of the function.

Example

void add(int& a, int& b); void add(double& a, double& b); void add(struct bob& a, struct bob& b);

```
void add(int&amp;amp; a, int&amp;amp; b);

void add(double&amp;amp; a,
double&amp;amp; b);

void add(struct bob&amp;amp; a, struct
bob&amp;amp; b);
```

15) What is operator overloading?

Operator overloading is a function where different operators are applied and depends on the arguments. Operator,-,* can be used to pass through the function, and it has its own precedence to execute

16) What is an abstract class?

An abstract class is a class which cannot be instantiated. Creation of an object is not possible with an abstract class, but it can be inherited. An abstract class can contain only an Abstract method. Java allows only abstract method in abstract class while other languages allow non-abstract method as well.

17) What is a ternary operator?

The ternary operator is said to be an operator which takes three arguments. Arguments and results are of different data types, and it depends on the function. The ternary operator is also called a conditional operator.

18) What is the use of finalize method?

Finalize method helps to perform cleanup operations on the resources which are not currently used. Finalize method is protected, and it is accessible only through this class or by a derived class.

19) What are the different types of arguments?

A parameter is a variable used during the declaration of the function or subroutine, and arguments are passed to the function body, and it should match with the parameter defined. There are two types of Arguments.

- Call by Value – Value passed will get modified only inside the function, and it returns the same value whatever it is passed into the function.
- Call by Reference – Value passed will get modified in both inside and outside the functions and it returns the same or different value.

20) What is the super keyword?

The super keyword is used to invoke the overridden method, which overrides one of its superclass methods. This keyword allows to access overridden methods and also to access hidden members of the superclass.

It also forwards a call from a constructor, to a constructor in the superclass.

21) What is method overriding?

Method overriding is a feature that allows a subclass to provide the implementation of a method that overrides in the main class. It will override the implementation in the superclass by providing the same method name, same parameter, and same return type.

22) What is an interface?

An interface is a collection of an abstract method. If the class implements an interface, it thereby inherits all the abstract methods of an interface.

Java uses Interface to implement multiple inheritances.

23) What is exception handling?

An exception is an event that occurs during the execution of a program. Exceptions can be of any type – Runtime exception, Error exceptions. Those exceptions are adequately handled through exception handling mechanism like try, catch, and throw keywords.

24) What are tokens?

A compiler recognizes a token, and it cannot be broken down into component elements. Keywords, identifiers, constants, string literals, and operators are examples of tokens.

Even punctuation characters are also considered as tokens. Example: Brackets, Commas, Braces, and Parentheses.

25) What is the main difference between overloading and overriding?

Overloading is static Binding, whereas Overriding is dynamic Binding. Overloading is nothing but the same method with different arguments, and it may or may not return the equal value in the same class itself.

Overriding is the same method names with the same arguments and return types associated with the class and its child class.

26) What is the main difference between a class and an object?

An object is an instance of a class. Objects hold multiple information, but classes don't have any information. Definition of properties and functions can be done in class and can be used by the object.

A class can have sub-classes, while an object doesn't have sub-objects.

27) What is an abstraction?

Abstraction is a useful feature of OOPS, and it shows only the necessary details to the client of an object. Meaning, it shows only required details for an object, not the inner constructors, of an object. Example – When you want to switch on the television, it is not necessary to know the inner circuitry/mechanism needed to switch on the TV. Whatever is required to switch on TV will be shown by using an abstract class.

28) What are the access modifiers?

Access modifiers determine the scope of the method or variables that can be accessed from other various objects or classes. There are five types of access modifiers, and they are as follows:

- Private
- Protected
- Public
- Friend
- Protected Friend

29) What are sealed modifiers?

Sealed modifiers are the access modifiers where the methods can not inherit it. Sealed modifiers can also be applied to properties, events, and methods. This modifier cannot be used to static members.

30) How can we call the base method without creating an instance?

Yes, it is possible to call the base method without creating an instance. And that method should be “Static method.”

Doing Inheritance from that class.-Use Base Keyword from a derived class.

31) What is the difference between new and override?

The new modifier instructs the compiler to use the new implementation instead of the base class function. Whereas, Override modifier helps to override the base class function.

32) What are the various types of constructors?

There are three types of constructors:

– Default Constructor – With no parameters.

– Parametric Constructor – With Parameters. Create a new instance of a class and also passing arguments simultaneously.

– Copy Constructor – Which creates a new object as a copy of an existing object.

33) What is early and late Binding?

Early binding refers to the assignment of values to variables during design time, whereas late Binding refers to the assignment of values to variables during run time.

34) What is 'this' pointer?

THIS pointer refers to the current object of a class. THIS keyword is used as a pointer which differentiates between the current object with the global object. It refers to the current object.

35) What is the difference between structure and a class?

The default access type of a Structure is public, but class access type is private. A structure is used for grouping data, whereas a class can be used for grouping data and methods. Structures are exclusively used for data, and it doesn't require strict validation, but classes are used to encapsulate and inherent data, which requires strict validation.

36) What is the default access modifier in a class?

The default access modifier of a class is Internal and the default access modifier of a class member is Private.

37) What is a pure virtual function?

A pure virtual function is a function which can be overridden in the derived class but cannot be defined. A virtual function can be declared as Pure by using the operator =0.

Example –

Virtual void function1() // Virtual, Not pure Virtual void function2() = 0 //Pure virtual

```
Virtual void function1() // Virtual,
Not pure

Virtual void function2() = 0 //Pure
virtual
```

38) What are all the operators that cannot be overloaded?

Following are the operators that cannot be overloaded -.

1. Scope Resolution (::)
2. Member Selection (.)
3. Member selection through a pointer to function (.*)

39) What is dynamic or run time polymorphism?

Dynamic or Run time polymorphism is also known as method overriding in which call to an overridden function is resolved during run time, not at the compile time. It means having two or more methods with the same name, same signature but with different implementation.

40) Do we require a parameter for constructors?

No, we do not require a parameter for constructors.

41) What is a copy constructor?

This is a special constructor for creating a new object as a copy of an existing object. There will always be only one copy constructor that can be either defined by the user or the system.

42) What does the keyword virtual represented in the method definition?

It means we can override the method.

43) Whether static method can use nonstatic members?

False.

44) What are a base class, subclass, and superclass?

The base class is the most generalized class, and it is said to be a root class.

A Subclass is a class that inherits from one or more base classes.

The superclass is the parent class from which another class inherits.

45) What is static and dynamic Binding?

Binding is nothing but the association of a name with the class. Static Binding is a binding in which name can be associated with the class during compilation time, and it is also called as early Binding.

Dynamic Binding is a binding in which name can be associated with the class during execution time, and it is also called as Late Binding.

46) How many instances can be created for an abstract class?

Zero instances will be created for an abstract class. In other words, you cannot create an instance of an Abstract Class.

47) Which keyword can be used for overloading?

Operator keyword is used for overloading.

48) What is the default access specifier in a class definition?

Private access specifier is used in a class definition.

49) Which OOPS concept is used as a reuse mechanism?

Inheritance is the OOPS concept that can be used as a reuse mechanism.

50) Which OOPS concept exposes only the necessary information to the calling functions?

Encapsulation

DATABASE

Given below is a list of most popular Database interview questions and answers for your reference.

Q #1) What do you understand by 'Database'?

Answer: Database is an organized collection of related data where the data is stored and organized to serve some specific purpose.

For Example, A librarian maintain a database of all the information related to the books that are available in the library.

Q #2) Define DBMS.

Answer: DBMS stands for Database Management System. It is a collection of application programs which allow the user to organize, restore and retrieve information about data efficiently and as effectively as possible.

Some of the popular DBMS's are MySql, Oracle, Sybase, etc.

Q #3) Define RDBMS.

Answer: Relational Database Management System(RDBMS) is based on a relational model of data that is stored in databases in separate tables and they are related to the use of a common column. Data can be accessed easily from the relational database using Structured Query Language (SQL).

Q #4) Enlist the advantages of DBMS.

Answer: The advantages of DBMS includes:

- Data is stored in a structured way and hence redundancy is controlled.
- Validates the data entered and provide restrictions on unauthorized access to the database.
- Provides backup and recovery of the data when required.
- It provides multiple user interfaces.

Q #5) What do you understand by Data Redundancy?

Answer: Duplication of data in the database is known as data redundancy. As a result of data redundancy, duplicated data is present at multiple locations, hence it leads to wastage of the storage space and the integrity of the database is destroyed.

Q #6) What are the various types of relationships in Database? Define them.

Answer: There are 3 types of relationships in Database:

- **One-to-one:** One table has a relationship with another table having the similar kind of column. Each primary key relates to only one or no record in the related table.
- **One-to-many:** One table has a relationship with another table that has primary and foreign key relations. The primary key table contains only one record that relates to none, one or many records in the related table.
- **Many-to-many:** Each record in both the tables can relate to many numbers of records in another table.

Q #7) Explain Normalization and De-Normalization.

Answer:

Normalization is the process of removing redundant data from the database by splitting the table in a well-defined manner in order to maintain data integrity. This process saves much of the storage space.

De-normalization is the process of adding up redundant data on the table in order to speed up the complex queries and thus achieve better performance.

Q #8) What are the different types of Normalization?

Answer: Different types of Normalization are:

- **First Normal Form (1NF):** A relation is said to be in 1NF only when all the entities of the table contain unique or atomic values.
- **Second Normal Form (2NF):** A relation is said to be in 2NF only if it is in 1NF and all the non-key attribute of the table is fully dependent on the primary key.
- **Third Normal Form (3NF):** A relation is said to be in 3NF only if it is in 2NF and every non-key attribute of the table is not transitively dependent on the primary key.

Q #9) What is BCNF?

Answer: BCNF is the Boyce Code Normal form. It is the higher version of 3Nf which does not have any multiple overlapping candidate keys.

Q #10) What is SQL?

Answer: Structured Query language, SQL is an ANSI(American National Standard Institute) standard programming language that is designed specifically for storing and managing the data in the relational database management system (RDBMS) using all kinds of data operations.

Q #11) How many SQL statements are used? Define them.

Answer: SQL statements are basically divided into three categories, DDL, DML, and DCL.

They can be defined as:

- **Data Definition Language (DDL)** commands are used to define the structure that holds the data. These commands are auto-committed

i.e. changes done by the DDL commands on the database are saved permanently.

- **Data Manipulation Language (DML)** commands are used to manipulate the data of the database. These commands are not auto-committed and can be rolled back.
- **Data Control Language (DCL)** commands are used to control the visibility of the data in the database like revoke access permission for using data in the database.

Q #12) Enlist some commands of DDL, DML, and DCL.

Answer: Data Definition Language (DDL) commands:

- CREATE to create a new table or database.
- ALTER for alteration.
- TRUNCATE to delete data from the table.
- DROP to drop a table.
- RENAME to rename a table.

Data Manipulation Language (DML) commands:

- INSERT to insert a new row.
- UPDATE to update an existing row.
- DELETE to delete a row.
- MERGE for merging two rows or two tables.

Data Control Language (DCL) commands:

- COMMIT to permanently save.
- ROLLBACK to undo the change.
- SAVEPOINT to save temporarily.

Q #13) Define DML Compiler.

Answer: DML compiler translates DML statements in a query language into a low-level instruction and the generated instruction can be understood by Query Evaluation Engine.

Q #14) What is DDL interpreter?

Answer: DDL Interpreter interprets the DDL statements and records the generated statements in the table containing metadata.

Q #15) Enlist the advantages of SQL.

Answer: Advantages of SQL are:

- Simple SQL queries can be used to retrieve a large amount of data from the database very quickly and efficiently.
- SQL is easy to learn and almost every DBMS supports SQL.
- It is easier to manage the database using SQL as no large amount of coding is required.

Q #16) Explain the terms 'Record', 'Field' and 'Table' in terms of database.

Answer:

Record: Record is a collection of values or fields of a specific entity. **For Example,** An employee, Salary account, etc.

Field: A field refers to an area within a record that is reserved for specific data. **For Example,** Employee ID.

Table: Table is the collection of records of specific types. **For Example,** the Employee table is a collection of records related to all the employees.

Q #17) What do you understand by Data Independence? What are its two types?

Answer: Data Independence refers to the ability to modify the schema definition in one level in such a way that it does not affect the schema definition in the next higher level.

The 2 types of Data Independence are:

- **Physical Data Independence**: It modifies the schema at the physical level without affecting the schema at the conceptual level.
- **Logical Data Independence:** It modifies the schema at the conceptual level without affecting or causing changes in the schema at the view level.

Q #18) Define the relationship between 'View' and 'Data Independence'.

Answer: View is a virtual table that does not have its data on its own rather the data is defined from one or more underlying base tables.

Views account for logical data independence as the growth and restructuring of base tables are not reflected in views.

Q #19) What are the advantages and disadvantages of views in the database?

Answer: Advantages of Views:

- As there is no physical location where the data in the view is stored, it generates output without wasting resources.
- Data access is restricted as it does not allow commands like insertion, updation, and deletion.

Disadvantages of Views:

- The view becomes irrelevant if we drop a table related to that view.
- Much memory space is occupied when the view is created for large tables.

Q #20) What do you understand by Functional dependency?

Answer: A relation is said to be in functional dependency when one attribute uniquely defines another attribute.

For Example, R is a Relation, X and Y are two attributes. T1 and T2 are two tuples. Then,

T1[X]=T2[X] and T1[Y]=T2[Y]

Means, the value of component X uniquely define the value of component Y.

Also, X->Y means Y is functionally dependent on X.

Q #21) When is functional dependency said to be the fully functional dependent?

Answer: To fulfill the criteria of fully functional dependency, the relation must meet the requirement of functional dependency.

A functional dependency ‘A’ and ‘B’ are said to be fully functional dependent when removal of any attribute say ‘X’ from ‘A’ means the dependency does not hold anymore.

Q #22) What do you understand by the E-R model?

Answer: E-R model is an Entity-Relationship model which defines the conceptual view of the database.

The E-R model basically shows the real-world entities and their association/relations. Entities here represent the set of attributes in the database.

Q #23) Define Entity, Entity type, and Entity set.

Answer:

Entity can be anything, be it a place, class or object which has an independent existence in the real world.

Entity Type represents a set of entities that have similar attributes.

Entity Set in the database represents a collection of entities having a particular entity type.

Q #24) Define a Weak Entity set.

Answer: Weak Entity set is the one whose primary key comprises its partial key as well as the primary key of its parent entity. This is the case because the entity set may not have sufficient attributes to form a primary key.

Q #25) Explain the terms 'Attribute' and 'Relations'

Answer:

Attribute is described as the properties or characteristics of an entity. **For Example**, Employee ID, Employee Name, Age, etc., can be attributes of the entity Employee.

Relation is a two-dimensional table containing a number of rows and columns where every row represents a record of the relation. Here, rows are also known as 'Tuples' and columns are known as 'Attributes'.

Q #26) What are VDL and SDL?

Answer: VDL is View Definition Language which represents user views and their mapping to the conceptual schema.

SDL is Storage Definition Language which specifies the mapping between two schemas.

Q #27) Define Cursor and its types.

Answer: Cursor is a temporary work area that stores the data, as well as the result set, occurred after manipulation of data retrieved. A cursor can hold only one row at a time.

The 2 types of Cursor are:

- **Implicit cursors** are declared automatically when DML statements like INSERT, UPDATE, DELETE is executed.
- **Explicit cursors** have to be declared when SELECT statements that are returning more than one row are executed.

Q #28) What is the Database transaction?

Answer: Sequence of operation performed which changes the consistent state of the database to another is known as the database transaction. After the completion of the transaction, either the successful completion is reflected in the system or the transaction fails and no change is reflected.

Q #29) Define Database Lock and its types.

Answer: Database lock basically signifies the transaction about the current status of the data item i.e. whether that data is being used by other transactions or not at the present point of time.

There are two types of Database lock: **Shared Lock** and **Exclusive Lock.**

Q #30) What is Data Warehousing?

Answer: The storage as well as access to data, that is being derived from the transactions and other sources, from a central location in order to perform the analysis is called Data Warehousing.

Q #31) What do you understand by Join?

Answer: Join is the process of deriving the relationship between different tables by combining columns from one or more tables having common values in each. When a table joins with itself, it is known as Self Join.

Q #32) What do you understand by Index hunting?

Answer: Index hunting is the process of boosting the collection of indexes which helps in improving the query performance as well as the speed of the database.

Q #33) How to improve query performance using Index hunting?

Answer: Index hunting help in improving query performance by:

- Using a query optimizer to coordinate queries with the workload.
- Observing the performance and effect of index and query distribution.

Q #34) Differentiate between ‘Cluster’ and ‘Non-cluster’ index.

Answer: Clustered index alters the table and re-order the way in which the records are stored in the table. Data retrieval is made faster by using the clustered index.

A Non-clustered index does alter the records that are stored in the table but creates a completely different object within the table.

Q #35) What are the disadvantages of a Query?

Answer: Disadvantages of a Query are:

- Indexes are not present.
- Stored procedures are excessively compiled.
- Difficulty in interfacing.

Q #36) What do you understand by Fragmentation?

Answer: Fragmentation is a feature that controls the logical data units, also known as fragments that are stored at different sites of a distributed database system.

Q #37) Define Join types.

Answer: Given below are the types of Join, which are explained with respect to the tables as an **Example.**

employee table:

EmpID	EmpName
1000	Rohan
1001	Shruti
1002	Nikhil
1003	Naveen

employee_info table:

EmpID	Address
1000	Delhi
1001	Mumbai
1002	Delhi
1003	Kolkata

a) Inner JOIN: Inner JOIN is also known as a simple JOIN. This SQL query returns results from both the tables having a common value in rows.

SQL Query:

```
SELECT * from employee, employee_info WHERE employee.EmpID =
employee_info.EmpID ;
```

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1000	Delhi
1002	Nikhil	1000	Delhi
1003	Naveen	1000	Delhi
1000	Rohan	1001	Mumbai
1001	Shruti	1001	Mumbai
1002	Nikhil	1001	Mumbai
1003	Naveen	1001	Mumbai
1000	Rohan	1002	Delhi
1001	Shruti	1002	Delhi
1002	Nikhil	1002	Delhi
1003	Naveen	1002	Delhi
1000	Rohan	1003	Kolkata
1001	Shruti	1003	Kolkata
1002	Nikhil	1003	Kolkata
1003	Naveen	1003	Kolkata

b) Natural JOIN: This is a type of Inner JOIN that returns results from both the tables having the same data values in the columns of both the tables to be joined.

SQL Query:

```
SELECT * from employee NATURAL JOIN
employee_info;
```

Result:

EmpID	EmpName	Address
1000	Rohan	Delhi
1001	Shruti	Mumbai
1002	Nikhil	Delhi
1003	Naveen	Kolkata

c) Cross JOIN: Cross JOIN returns the result as all the records where each row from the first table is combined with each row of the second table.

SQL Query:

```
SELECT * from employee CROSS JOIN
employee_info;
```

Result:

Let us do some modification in the above tables to understand Right JOIN, Left JOIN, and Full JOIN.

employee table:

EmpID	EmpName
1000	Rohan
1001	Shruti
1002	Nikhil
1003	Naveen
1004	Shikha
1005	Shalu

employee_info table:

EmpID	Address
1000	Delhi
1001	Mumbai
1002	Gurgaon
1003	Kolkata
1006	Noida
1007	Kerala

a) Right JOIN: Right JOIN is also known as Right Outer JOIN. This returns all the rows as a result from the right table even if the JOIN condition does not match any records in the left table.

SQL Query:

```
SELECT * from employee RIGHT OUTER JOIN employee_info on
(employee.EmpID = employee_info.EmpID);
```

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1001	Mumbai
1002	Nikhil	1002	Delhi
1003	Naveen	1003	Kolkata
Null	Null	1006	Noida
Null	Null	1007	Kerala

b) Left JOIN: Left JOIN is also known as Left Outer JOIN. This returns all the rows as a result of the left table even if the JOIN condition does not match any records in the right table. This is exactly the opposite of Right JOIN.

SQL Query:

```
SELECT * from employee LEFT OUTER JOIN employee_info on
(employee.EmpID = employee_info.EmpID);
```

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1001	Mumbai
1002	Nikhil	1002	Delhi
1003	Naveen	1003	Kolkata
1004	Shikha	null	Null
1005	Shalu	null	Null

c) Outer/Full JOIN: Full JOIN return results in combining the result of both the Left JOIN and Right JOIN.

SQL Query:

```
SELECT * from employee FULL OUTER JOIN employee_info on
(employee.EmpID = employee_info.EmpID);
```

Result:

EmpID	EmpName	EmpID	Address
1000	Rohan	1000	Delhi
1001	Shruti	1001	Mumbai
1002	Nikhil	1002	Delhi
1003	Naveen	1003	Kolkata
1004	Shikha	null	Null
1005	Shalu	null	Null
Null	Null	1006	Noida
Null	Null	1007	Kerala

Q #38) What do you understand by 'Atomicity' and 'Aggregation'?

Answer: Atomicity is the condition where either all the actions of the transaction are performed or none. This means, when there is an incomplete transaction, the database management system itself will undo the effects done by the incomplete transaction.

Aggregation is the concept of expressing the relationship with the collection of entities and their relationships.

Q #39) Define Phantom deadlock.

Answer: Phantom deadlock detection is the condition where the deadlock does not actually exist but due to a delay in propagating local information, deadlock detection algorithms identify the deadlocks.

Q #40) Define checkpoint.

Answer: Checkpoint declares a point before which all the logs are stored permanently in the storage disk and is the inconsistent state. In the case of crashes, the amount of work and time is saved as the system can restart from the checkpoint.

Q #41) What is Database partitioning?

Answer: Database partitioning is the process of partitioning tables, indexes into smaller pieces in order to manage and access the data at a finer level.

This process of partitioning reduces the cost of storing a large amount of data as well as enhances the performance and manageability.

Q #42) Explain the importance of Database partitioning.

Answer: The importance of Database partitioning are:

- Improves query performance and manageability.
- Simplifies common administration tasks.
- Acts as a key tool for building systems with extremely high availability requirements.
- Allows accessing a large part of a single partition.

Q #43) Explain the Data Dictionary.

Answer: Data dictionary is a set of information describing the content and structure of the tables and database objects. The job of the information stored in the data dictionary is to control, manipulate and access the relationship between database elements.

Q #44) Explain the Primary Key and Composite Key.

Answer: Primary Key is that column of the table whose every row data is uniquely identified. Every row in the table must have a primary key and no two rows can have the same primary key. Primary key value can never be null nor can it be modified or updated.

Composite Key is a form of the candidate key where a set of columns will uniquely identify every row in the table.

Q #45) What do you understand by the Unique key?

Answer: A Unique key is the same as the primary key whose every row data is uniquely identified with a difference of null value i.e. Unique key allows one value as a NULL value.

Q #46) What do you understand by Database Triggers?

Answer: A set of commands that automatically get executed when an event like Before Insert, After Insert, On Update, On Delete of row occurs in a table is called as Database trigger.

Q #47) Define Stored procedures.

Answer: A Stored procedure is a collection of pre-compiled SQL Queries, which when executed denotes a program taking input, process and gives the output.

Q #48) What do you understand by B-Trees?

Answer: B-Tree represents the data structure in the form of a tree for external memory that reads and writes large blocks of data. It is commonly used in databases and file systems where all the insertions, deletions, sorting, etc., are done in logarithmic time.

Q #49) Name the different data models that are available for database systems.

Answer: Different data models are:

- Relational model
- Network model
- Hierarchical model

Q #50) Differentiate between 'DELETE', 'TRUNCATE' and 'DROP' commands.

Answer: After the execution of **'DELETE'** operation, COMMIT and ROLLBACK statements can be performed to retrieve the lost data.

After the execution of **'TRUNCATE'** operation, COMMIT, and ROLLBACK statements cannot be performed to retrieve the lost data.

'DROP' command is used to drop the table or key like the primary key/foreign key.

Q #51) Based on the given table, solve the following queries.

Employee table

empId	empName	Age	Address
1001	Rohan	26	Delhi
1002	Ankit	30	Gurgaon
1003	Gaurav	27	Mumbai
1004	Raja	32	Nagpur

a) Write the SELECT command to display the details of the employee with empid as 1004.

SQL Query:

```
SELECT empId, empName, Age, Address from Employee
WHERE empId = 1004;
```

Result:

empId	empName	Age	Address
1004	Raja	32	Nagpur

b) Write the SELECT command to display all the records of table Employees.

SQL Query:

```
SELECT * from
Employee;
```

Result:

empId	empName	Age	Address
1001	Rohan	26	Delhi
1002	Ankit	30	Gurgaon
1003	Gaurav	27	Mumbai
1004	Raja	32	Nagpur

c) Write the SELECT command to display all the records of the employee whose name starts with the character ‘R’.

SQL Query:

```
SELECT * from Employee WHERE
empName LIKE ‘R%’;
```

Result:

empId	empName	Age	Address
1001	Rohan	26	Delhi
1004	Raja	32	Nagpur

d) Write a SELECT command to display id, age and name of the employees with their age in both ascending and descending order.

SQL Query:

```
SELECT empId, empName, Age from Employee
ORDER BY Age;
```

Result:

empId	empName	Age
1001	Rohan	26
1003	Gaurav	27
1002	Ankit	30
1004	Raja	32

```
SELECT empId, empName, Age from Employee
ORDER BY Age Desc;
```

Result:

empId	empName	Age
1004	Raja	32
1002	Ankit	30
1003	Gaurav	27
1001	Rohan	26

e) Write the SELECT command to calculate the total amount of salary on each employee from the below Emp table.

Emp table:

empId	empName	Age	Salary
1004	Raja	32	6000
1002	Ankit	30	8000
1003	Gaurav	27	5000
1001	Rohan	26	12000
1005	Sneha	29	9000

SQL Query:

```
SELECT empName, SUM(Salary) from Emp
GROUP BY empName;
```

Result:

empName	Salary
Ankit	8000
Gaurav	5000
Raja	6000
Rohan	12000
Sneha	9000

Data Structures

1) What is data structure?

Data structure refers to the way data is organized and manipulated. It seeks to find ways to make data access more efficient. When dealing with the data structure, we not only focus on one piece of data but the different set of data and how they can relate to one another in an organized manner.

2) Differentiate between file and structure storage structure.

The key difference between both the data structure is the memory area that is being accessed. When dealing with the structure that resides the main memory of the computer system, this is referred to as storage structure. When dealing with an auxiliary structure, we refer to it as file structures.

3) When is a binary search best applied?

A binary search is an algorithm that is best applied to search a list when the elements are already in order or sorted. The list is searched starting in the middle, such that if that middle value is not the target search key, it will check to see if it will continue the search on the lower half of the list or the higher half. The split and search will then continue in the same manner.

4) What is a linked list?

A linked list is a sequence of nodes in which each node is connected to the node following it. This forms a chain-like link for data storage.

5) How do you reference all the elements in a one-dimension array?

To reference all the elements in a one -dimension array, you need to use an indexed loop, So that, the counter runs from 0 to the array size minus one. In this manner, You can reference all the elements in sequence by using the loop counter as the array subscript.

6) In what areas do data structures are applied?

Data structures are essential in almost every aspect where data is involved. In general, algorithms that involve efficient data structure is applied in the following areas: numerical analysis, operating system, A.I., compiler design, database management, graphics, and statistical analysis, to name a few.

7) What is LIFO?

LIFO is a short form of Last In First Out. It refers how data is accessed, stored and retrieved. Using this scheme, data that was stored last should be the one to be extracted first. This also means that in order to gain access to the first data, all the other data that was stored before this first data must first be retrieved and extracted.

8) What is a queue?

A queue is a data structure that can simulate a list or stream of data. In this structure, new elements are inserted at one end, and existing elements are removed from the other end.

9) What are binary trees?

A binary tree is one type of data structure that has two nodes, a left node, and a right node. In programming, binary trees are an extension of the linked list structures.

10) Which data structures are applied when dealing with a recursive function?

Recursion, is a function that calls itself based on a terminating condition, makes use of the stack. Using LIFO, a call to a recursive function saves the return address so that it knows how to return to the calling function after the call terminates.

11) What is a stack?

A stack is a data structure in which only the top element can be accessed. As data is stored in the stack, each data is pushed downward, leaving the most recently added data on top.

12) Explain Binary Search Tree

A binary search tree stores data in such a way that they can be retrieved very efficiently. The left subtree contains nodes whose keys are less than the node's key value, while the right subtree contains nodes whose keys are greater than or equal to the node's key value. Moreover, both subtrees are also binary search trees.

13) What are multidimensional arrays?

Multidimensional arrays make use of multiple indexes to store data. It is useful when storing data that cannot be represented using single dimensional indexing, such as data representation in a board game, tables with data stored in more than one column.

14) Are linked lists considered linear or non-linear data structures?

It depends on where you intend to apply linked lists. If you based it on storage, a linked list is considered non-linear. On the other hand, if you based it on access strategies, then a linked list is considered linear.

15) How does dynamic memory allocation help in managing data?

Apart from being able to store simple structured data types, dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.

16) What is FIFO?

FIFO stands for First-in, First-out, and is used to represent how data is accessed in a queue. Data has been inserted into the queue list the longest is the one that is removed first.

17) What is an ordered list?

An ordered list is a list in which each node's position in the list is determined by the value of its key component, so that the key values form an increasing sequence, as the list is traversed.

18) What is merge sort?

Merge sort, is a divide-and-conquer approach for sorting the data. In a sequence of data, adjacent ones are merged and sorted to create bigger sorted lists. These sorted lists are then merged again to form an even bigger sorted list, which continues until you have one single sorted list.

19) Differentiate NULL and VOID

Null is a value, whereas Void is a data type identifier. A variable that is given a Null value indicates an empty value. The void is used to identify pointers as having no initial size.

20) What is the primary advantage of a linked list?

A linked list is an ideal data structure because it can be modified easily. This means that editing a linked list works regardless of how many elements are in the list.

21) What is the difference between a PUSH and a POP?

Pushing and popping applies to the way data is stored and retrieved in a stack. A push denotes data being added to it, meaning data is being "pushed" into the stack. On the other hand, a pop denotes data retrieval, and in particular, refers to the topmost data being accessed.

22) What is a linear search?

A linear search refers to the way a target key is being searched in a sequential data structure. In this method, each element in the list is checked and compared against the target key. The process is repeated until found or if the end of the file has been reached.

23) How does variable declaration affect memory allocation?

The amount of memory to be allocated or reserved would depend on the data type of the variable being declared. For example, if a variable is declared to be of integer type, then 32 bits of memory storage will be reserved for that variable.

24) What is the advantage of the heap over a stack?

The heap is more flexible than the stack. That's because memory space for the heap can be dynamically allocated and de-allocated as needed. However, the memory of the heap can at times be slower when compared to that stack.

25) What is a postfix expression?

A postfix expression is an expression in which each operator follows its operands. The advantage of this form is that there is no need to group sub-expressions in parentheses or to consider operator precedence.

26) What is Data abstraction?

Data abstraction is a powerful tool for breaking down complex data problems into manageable chunks. This is applied by initially specifying the data objects involved and the operations to be performed on these data objects without being overly concerned with how the data objects will be represented and stored in memory.

27) How do you insert a new item in a binary search tree?

Assuming that the data to be inserted is a unique value (that is, not an existing entry in the tree), check first if the tree is empty. If it's empty, just insert the new item in the root node. If it's not empty, refer to the new item's key. If it's smaller than the root's key, insert it into the root's left subtree, otherwise, insert it into the root's right subtree.

28) How does a selection sort work for an array?

The selection sort is a fairly intuitive sorting algorithm, though not necessarily efficient. In this process, the smallest element is first located

and switched with the element at subscript zero, thereby placing the smallest element in the first position.

The smallest element remaining in the subarray is then located next to subscripts 1 through n-1 and switched with the element at subscript 1, thereby placing the second smallest element in the second position. The steps are repeated in the same manner till the last element.

29) How do signed and unsigned numbers affect memory?

In the case of signed numbers, the first bit is used to indicate whether positive or negative, which leaves you with one bit short. With unsigned numbers, you have all bits available for that number. The effect is best seen in the number range (an unsigned 8-bit number has a range 0-255, while the 8-bit signed number has a range -128 to +127.

30) What is the minimum number of nodes that a binary tree can have?

A binary tree can have a minimum of zero nodes, which occurs when the nodes have NULL values. Furthermore, a binary tree can also have 1 or 2 nodes.

31) What are dynamic data structures?

Dynamic data structures are structures that expand and contract as a program runs. It provides a flexible means of manipulating data because it can adjust according to the size of the data.

32) In what data structures are pointers applied?

Pointers that are used in linked list have various applications in the data structure. Data structures that make use of this concept include the Stack, Queue, Linked List and Binary Tree.

33) Do all declaration statements result in a fixed reservation in memory?

Most declarations do, with the exemption of pointers. Pointer declaration does not allocate memory for data, but for the address of the pointer variable. Actual memory allocation for the data comes during run-time.

34) What are ARRAYs?

When dealing with arrays, data is stored and retrieved using an index that refers to the element number in the data sequence. This means that data can be accessed in any order. In programming, an array is declared as a variable having a number of indexed elements.

35) What is the minimum number of queues needed when implementing a priority queue?

The minimum number of queues needed in this case is two. One queue is intended for sorting priorities while the other queue is used for actual storage of data.

36) Which sorting algorithm is considered the fastest?

There are many types of sorting algorithms: quick sort, bubble sort, balloon sort, radix sort, merge sort, etc. Not one can be considered the fastest because each algorithm is designed for a particular data structure and data set. It would depend on the data set that you would want to sort.

37) Differentiate STACK from ARRAY.

Stack follows a LIFO pattern. It means that data access follows a sequence wherein the last data to be stored when the first one to be extracted. Arrays, on the other hand, does not follow a particular order and instead can be accessed by referring to the indexed element within the array.

38) Give a basic algorithm for searching a binary search tree.

1. if the tree is empty, then the target is not in the tree, end search
    2. if the tree is not empty, the target is in the tree
    3. check if the target is in the root item
    4. if a target is not in the root item, check if a target is smaller than

the root's value
5. if a target is smaller than the root's value, search the left subtree
6. else, search the right subtree

39) What is a dequeue?

A dequeue is a double-ended queue. This is a structure wherein elements can be inserted or removed from either end.

40) What is a bubble sort and how do you perform it?

A bubble sort is one sorting technique that can be applied to data structures such as an array. It works by comparing adjacent elements and exchanges their values if they are out of order. This method lets the smaller values "bubble" to the top of the list, while the larger value sinks to the bottom.

41) What are the parts of a linked list?

A linked list typically has two parts: the head and the tail. Between the head and tail lie the actual nodes. All these nodes are linked sequentially.

42) How does selection sort work?

Selection sort works by picking the smallest number from the list and placing it at the front. This process is repeated for the second position towards the end of the list. It is the simplest sort algorithm.

43) What is a graph?

A graph is one type of data structure that contains a set of ordered pairs. These ordered pairs are also referred to as edges or arcs and are used to connect nodes where data can be stored and retrieved.

44) Differentiate linear from a nonlinear data structure.

The linear data structure is a structure wherein data elements are adjacent to each other. Examples of linear data structure include arrays, linked lists, stacks, and queues. On the other hand, a non-linear data structure is a

structure wherein each data element can connect to more than two adjacent data elements. Examples of nonlinear data structure include trees and graphs.

45) What is an AVL tree?

An AVL tree is a type of binary search tree that is always in a state of partially balanced. The balance is measured as a difference between the heights of the subtrees from the root. This self-balancing tree was known to be the first data structure to be designed as such.

46) What are doubly linked lists?

Doubly linked lists are a special type of linked list wherein traversal across the data elements can be done in both directions. This is made possible by having two links in every node, one that links to the next node and another one that connects to the previous node.

47) What is Huffman's algorithm?

Huffman's algorithm is used for creating extended binary trees that have minimum weighted path lengths from the given weights. It makes use of a table that contains the frequency of occurrence for each data element.

48) What is Fibonacci search?

Fibonacci search is a search algorithm that applies to a sorted array. It makes use of a divide-and-conquer approach that can significantly reduce the time needed in order to reach the target element.

49) Briefly explain recursive algorithm.

Recursive algorithm targets a problem by dividing it into smaller, manageable sub-problems. The output of one recursion after processing one sub-problem becomes the input to the next recursive process.

50) How do you search for a target key in a linked list?

To find the target key in a linked list, you have to apply sequential search. Each node is traversed and compared with the target key, and if it is different, then it follows the link to the next node. This traversal continues until either the target key is found or if the last node is reached.

ANALYTICAL QUESTIONS

1) Describe a time when you were given a problem without a lot of information. How did you handle this situation?

This question assesses your problem-solving skills, along with your research and logical thinking abilities. When answering this question, highlight your process and how you find information and use it to help you get to the next steps. Consider using the STAR technique, which stands for Situation, Task, Action and Result.

Example: *"In my current role, I am often tasked with developing better processes in our workplace. My first step is to look at our database and see if there are other processes we've used previously. One particular process included entering information into both a cloud-based database and a spreadsheet. I found that we weren't always updating the info in both places. I also saw that the database could export to a spreadsheet. I brought this up to my manager, and now all the information only gets entered into the database so that when we use the information for projects, it's the most accurate."*

2) How do you use advantages and disadvantages to make a decision?

This question gauges your ability to assess both positive and negative situations to improve your processes at work. When you answer this question, describe what your process is and how you evaluate the good and bad in your work.

Example: *"I try to make this process as simple as I can. In general, if the advantages outweigh the disadvantages, then we will usually move forward with that decision. However, I do try to take this to another level. For*

example, there have been a few instances where the pros outweighed the cons by one advantage. So I took a deeper dive into what this decision could cost us if we moved forward with it. I also looked at other costs outside of financials and found that this decision would end up costing us more in the long-term and could end up a detriment to our company. So we chose to forgo the decision and ultimately found a better solution to our current issues."

3) Describe your process in troubleshooting an issue.

This is a simple question that an employer will ask to assess your process development skills. An employer will want to hear about ways you might deviate from your usual process and why.

Example: *"I always like to understand the background of the issue quickly. From there, I go through the steps and see if there was a misstep somewhere that caused the issue. After that, I'll try different ways to fix the problem while testing those out. Hopefully, at this point, I've found a solution. If not, I will backtrack to see if there is another step I missed or if I need to escalate it to managers."*

4) What types of metrics do you currently track, and how do you use these to make a decision?

Your interviewer might ask this to see what you are currently using to make decisions and if your decision-making process is similar to theirs. They may also be looking to see how you would adapt to new processes when needed.

Example: *"I currently use analytical software that tracks website views, click-throughs and unique visitors. These metrics are great for my team when we are developing campaigns to recruit new customers. This also helps us track if a current campaign is not doing quite as well, along with how often we are getting new consumers to our website. We use these metrics in many ways to determine what our next campaign will be along with if we need to pull current campaigns sooner."*

5) Give an example of a time when you had to take a risk to achieve your desired goal. What was your approach, and what was the outcome?

An employer wants to know more about how you come up with and reach goals. When you answer this question, give details about how you take informed risks to achieve those goals. It will also help to have an example that showcases these abilities.

Example: *"I recently had a situation like this. My team needed to fix an issue with our company's website. Our contact page wouldn't load, and we explored all the reasons why this could happen. Normally we don't deviate from our usual processes, but I saw something in our code that just seemed off. I spoke up about it and said that all we needed to do was add a semicolon to a couple of lines. At this point, we just needed to fix the website so that we didn't lose any customers. So we tried it, and it worked. Our team is now backtracking so that we know more about why this happened. Normally we would have done some tests first, but we had to act fast, and it paid off."*

Algorithms

1) Explain what is an algorithm in computing?

An algorithm is a well-defined computational procedure that take some value as input and generate some value as output. In simple words, it's a sequence of computational steps that converts input into the output.

2) Explain what is Quick Sort algorithm?

Quick Sort algorithm has the ability to sort list or queries quickly. It is based on the principle of partition exchange sort or Divide and conquer. This type of algorithm occupies less space, and it segregates the list into three main parts

- Elements less than the Pivot element
- Pivot element
- Elements greater than the Pivot element

3) Explain what is time complexity of Algorithm?

Time complexity of an algorithm indicates the total time needed by the program to run to completion. It is usually expressed by using the **big O notation.**

4) Mention what are the types of Notation used for Time Complexity?

The types of Notations used for Time Complexity includes

- **Big Oh:** It indicates "fewer than or the same as" <expression>iterations
- **Big Omega**: It indicates "more than or same as" <expression>iterations
- **Big Theta:** It indicates "the same as"<expression>iterations
- **Little Oh:** It indicates "fewer than" <expression>iterations
- **Little Omega:** It indicates "more than" <expression>iterations

5) Explain how binary search works?

In binary search, we compare the key with the item in the middle position of the array. If the key is less than the item searched then it must lie in the lower half of the array, if the key is greater than the item searched than it should be in upper half of the array.

6) Explain whether it is possible to use binary search for linked lists?

Since random access is not acceptable in linked list, it is impossible to reach the middle element of O(1) time. Thus, binary search is not possible for linked list.

7) Explain what is heap sort?

Heap-sort can be defined as a comparison based sorting algorithm. It divides its input into the unsorted and sorted region, until it shrinks the unsorted region by eliminating the smallest element and moving that to the sorted region.

8) Explain what is Skip list?

Skip list the method for data structuring, where it allows the algorithm to search, delete and insert elements in a symbol table or dictionary. In a skip list, each element is represented by a node. The search function returns the content of the value related to key. The insert operation associates a specified key with a new value, while the delete function deletes the specified key.

9) Explain what is Space complexity of insertion sort algorithm?

Insertion sort is an in-place sorting algorithm which means that it requires no extra or little. storage. For insertion sort, it requires only single list elements to be stored out-side the initial data, making the space-complexity 0(1).

10) Explain what a "Hash Algorithm" is and what are they used for?

"Hash Algorithm" is a hash function that takes a string of any length and decreases it to a unique fixed length string. It is used for password validity, message & data integrity and for many other cryptographic systems.

11) Explain how to find whether the linked list has a loop?

To know whether the linked list has a loop, we will take two pointer approach. If we maintain two pointers, and we increase one pointer after processing two nodes and other after processing every node, we are likely to encounter a situation where both the pointer will be pointing to the same node. This will only occur if linked list has a loop.

12) Explain how encryption algorithm works?

Encryption is the process of converting plaintext into a secret code format referred as "Ciphertext". To convert the text, algorithm uses a string of bits referred as "keys" for calculations. The larger the key, the greater the number of potential patterns for creating cipher text. Most encryption algorithm use codes fixed blocks of input that have length about 64 to 128 bits, while some uses stream method.

13) List out some of the commonly used cryptographic algorithms?

Some of the commonly used cryptographic algorithms are

- 3-way
- Blowfish
- CAST
- CMEA
- GOST
- DES and Triple DES
- IDEA
- LOKI and so on

14) Explain what is the difference between best case scenario and worst case scenario of an algorithm?

- **Best case scenario:** Best case scenario for an algorithm is explained as the arrangement of data for which the algorithm performs best. For example, we take a binary search, for which the best case scenario would be if the target value is at the very center of the data you are searching. The best case time complexity would be 0 (1)
- **Worst case scenario:** It is referred for the worst set of input for a given algorithm. For example quicksort, which can perform worst if you select the largest or smallest element of a sublist for the pivot value. It will cause quicksort to degenerate to O (n2).

15) Explain what is Radix Sort algorithm?

Radix sort puts the element in order by comparing the digits of the numbers. It is one of the linear sorting algorithms for integers.

16) Explain what is a recursive algorithm?

Recursive algorithm is a method of solving a complicated problem by breaking a problem down into smaller and smaller sub-problems until you get the problem small enough that it can be solved easily. Usually, it involves a function **calling itself**.

17) Mention what are the three laws of recursion algorithm?

All recursive algorithm must follow three laws

- It should have a base case
- A recursive algorithm must call itself
- A recursive algorithm must change its state and move towards the base case

18) Explain what is bubble sort algorithm?

Bubble sort algorithm is also referred as sinking sort. In this type of sorting, the list to be sorted out compares the pair of adjacent items. If they are organized in the wrong order, it will swap the values and arrange them in the correct order.

Recursions

Question 1 **Write a recursive function that takes a number and returns the sum of all the numbers from zero to that number.**

I will call this function '**cumulative'.** If I provide 10 as an input it should return the sum of all the numbers from zero to 10. That is 55.

Here is the python solution:

```
def cumulative(num):
  if num in [0, 1]:
     return num
  else:
     return num + cumulative(num-1)
```

Question 2 **Write a recursive function that takes a number as an input and returns the factorial of that number.**

I am sure we all learned what factorial is. I will name the function 'factorial'.

Here is the python solution:

```
def factorial(n):
  assert n >=0 and int(n) == n, 'The number must be a positive integer only!'
  if n in [0,1]:
     return 1
  else:
     return n * factorial(n-1)
```

Question 3 **Write a recursive function that takes a number 'n' and returns the nth number of the Fibonacci number.**

As a reminder Fibonacci series is the sequence of positive integers that start with 0 and 1 and the rest of the numbers are just the sum of the previous two numbers: 0, 1, 1, 2, 3, 5, 8, 11…

Here is the python solution:

```
def fibonacci(n):
  if n in [0, 1]:
     return n
  else:
     return fibonacci(n-1) + fibonacci(n-2)
```

Question 4 **Write a recursive function that takes a list of numbers as an input and returns the product of all the numbers in the list.**

If you are not a python user, a list in python is like an array in Java or JavaScript, or PHP.

Here is the python solution:

```
def productOfArray(arr):
  if len(arr) == 0:
     return 0
  if len(arr) == 1:
     return arr[0]
  else:
```

```
    return arr[len(arr)-1] * productOfArray(arr[:len(arr)-1])
```

Question 5 **Write a function that takes a string and returns if the string is a palindrome.**

As a reminder, if a string is equal to its reverse, it is called a palindrome. Such as Madam, civic, kayak. If you reverse any of these words they stay the same.

Here is the recursive solution in python:

```
def isPalindrom(strng):
  if len(strng) == 0:
    return True
  if strng[0] != strng[len(strng)-1]:
    return False
  return isPalindrome(strng[1:-1])
```

This function returns True if the string is a palindrome and false otherwise.

Question 6 **Write a recursive function that takes a string and reverse the string.**

If the input is ‘amazing’, it should return ‘gnizama’.

Here is the Python solution:

```
def reverse(st):
  if len(st) in [0, 1]:
    return st
  else:
    return st[len(st)-1] + reverse(st[:len(st)-1])
```

Question 7 **Write a recursive function that takes an array that may contain more arrays in it and returns an array with all values flattened.**

Suppose this is the input array:

[[1], [2, 3], [4], [3, [2, 4]]]

The output should be:

[1, 2, 3, 4, 3, 2, 4]

Here is the python solution:

```
def flatten(arr):
  res = []
  for i in arr:
    if type(i) is list:
      res.extend(flatten(i))
    else:
      res.append(i)
  return res
```

Question 8 **Write a recursive function that takes an array of words and returns an array that contains all the words capitalized.**

If this is the input array:

['foo', 'bar', 'world', 'hello']

The output array should be:

['FOO', 'BAR', 'WORLD', 'HELLO']

Here is the python solution:

```
def capitalizeWords(arr):
  if len(arr) == 0:
    return []
  else:
    return [arr[0].upper()]+capitalizeWords(arr[1:])
```

Question 9 **Write a recursive function that takes an array and a callback function and returns True if any value of that array returns True from that callback function otherwise returns False.**

In this solution, I used the function ‘isEven’ as a callback function that returns True if a number is even number and returns False otherwise.

Here is the callback function:

```
def isEven(num):
  if num%2==0:
     return True
  else:
     return False
```

Our main recursive function should return True if even one element of the input array returns True from the ‘isEven’ function and False otherwise. Here is an array:

[1, 2, 3, 5]

The recursive function should return True here because this array has one element that is an even number.

Here is the python solution:

```
def anyEven(arr, cb):
  if len(arr) == 0:
     return False
  if cb(arr[0]):
     return True
  return anyEven(arr[1:], cb)
```

Question 10 **Write a recursive function that will return the sum of all the positive numbers in a dictionary which may contain more dictionaries nested in it.**

Here is an example:

```
obj = {
 "a": 2,
 "b": {"x": 2, "y": {"foo": 3, "z": {"bar": 2}}},
```

```
 "c": {"p": {"h": 2, "r": 5}, "q": 'ball', "r": 5},
 "d": 1,
 "e": {"nn": {"lil": 2}, "mm": 'car'}
```

This should return 10. Because this dictionary contains five 2s and no other even numbers.

Here is the python solution:

```
def evenSum(obj, sum=0):
   for k in obj.values():
      if type(k) == int and k%2 ==0:
         sum += k
      elif isinstance(k, dict):
         sum += evenSum(k, sum=0)
   return sum
```