OJEE 2026 Shift 2 M.Sc Comp.Science

Question Paper with Solutions (Memory-Based)

Conducted by Odisha Joint Entrance Examination Committee (OJEEC))

General Instructions

(i) The examination will be conducted in Computer-Based Test (CBT) mode.

(ii) OJEE MCA 2026 Question Paper consists of 120 questions.

(iii) A total of 480 marks to be attempted in 2 hours.

(iv) Each correct answer carries +4 marks, and there is a negative marking of 1 for incorrect answers.

1. Which among the following allows the user to interact computer?

(A) Command line interface

(B) Graphical user interface

(C) Both options A and B

(D) None of these

Correct Answer: (C) Both options A and B

Solution:

Step 1: Understanding the Question:

The question asks for the mechanisms or interfaces through which a human user can interact with a computer system.

Step 2: Detailed Explanation:

An interface acts as a boundary across which two independent entities meet and communicate. In computing, a user interface allows a user to control a software application or hardware

device.

A **Command Line Interface (CLI)** allows users to interact with the computer by typing text-based commands into a console or terminal.

A **Graphical User Interface (GUI)** allows users to interact with the computer using graphical elements like windows, icons, menus, and pointers.

Both CLI and GUI are standard and fundamental methods for human-computer interaction.

Step 3: Final Answer:

Since both Command line interface and Graphical user interface are correct, the right choice is "Both options A and B".

Quick Tip: Remember that CLI uses text-based commands (like MS-DOS or Linux Terminal), while GUI uses visual elements (like Windows OS or macOS) for interaction.

---

2. The number 100101_2 is equivalent to octal

(A) 54

(B) 45

(C) 37

(D) 25

Correct Answer: (B) 45

Solution:

Step 1: Understanding the Question:

We are given a number in the binary number system (base 2) and need to convert it into its equivalent value in the octal number system (base 8).

Step 2: Key Formula or Approach:

To convert a binary number to an octal number, group the binary digits into sets of three,

starting from the rightmost digit (the least significant bit).

Then, convert each 3-bit group into its equivalent decimal/octal digit.

Step 3: Detailed Explanation:

The given binary number is 100101_2.

Let's group the digits into blocks of three starting from the right:

$$\underbrace{100}_{\text{Group 2}}\ \underbrace{101}_{\text{Group 1}}$$

Now, evaluate the decimal value of each group.

For the first group on the right (101):

$$1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 4 + 0 + 1 = 5$$

For the second group on the left (100):

$$1 \times 2^2 + 0 \times 2^1 + 0 \times 2^0 = 4 + 0 + 0 = 4$$

Combining these evaluated digits from left to right, we get 45.

Therefore, $(100101)_2 = (45)_8$.

Step 4: Final Answer:

The octal equivalent is 45.

Quick Tip: Since $2^3 = 8$, every single octal digit precisely represents a unique 3-bit binary sequence. Memorizing the 3-bit sequences from 000 to 111 speeds up these conversions significantly.

---

3. Convert $(153.513)_{10}$ in octal number system is

(A) 231.408517

(B) 231.407517

(C) 231.406517

(D) 231.406617

Correct Answer: (C) 231.406517

Solution:

Step 1: Understanding the Question:

We need to convert a fractional decimal number (base 10) into an octal number (base 8).

Step 2: Key Formula or Approach:

To convert a decimal number to octal, we split the number into its integer and fractional parts.

For the integer part, we repeatedly divide by 8 and record the remainders.

For the fractional part, we repeatedly multiply by 8 and record the integer parts of the successive products.

Step 3: Detailed Explanation:

First, let's convert the integer part, which is 153.

$$153 \div 8 = 19 \text{ with a remainder of } 1$$

$$19 \div 8 = 2 \text{ with a remainder of } 3$$

$$2 \div 8 = 0 \text{ with a remainder of } 2$$

Reading the remainders from bottom to top, we get 231.

So, $(153)_{10} = (231)_8$.

Next, let's convert the fractional part, which is 0.513.

$$0.513 \times 8 = 4.104 \implies \text{Integer part is } 4$$

$$0.104 \times 8 = 0.832 \implies \text{Integer part is } 0$$

$$0.832 \times 8 = 6.656 \implies \text{Integer part is } 6$$

$$0.656 \times 8 = 5.248 \implies \text{Integer part is } 5$$

$$0.248 \times 8 = 1.984 \implies \text{Integer part is } 1$$

$$0.984 \times 8 = 7.872 \implies \text{Integer part is } 7$$

Reading the extracted integer parts from top to bottom, we get .406517.

So, $(0.513)_{10} \approx (0.406517)_8$.

Combining the integer and fractional parts, we get $(231.406517)_8$.

Step 4: Final Answer:

The complete octal representation is 231.406517.

Quick Tip: Notice that Option A contains the digit '8', which is strictly invalid in the octal number system (base 8 only uses digits 0-7), allowing you to eliminate it immediately.

---

4. The main advantage of hexadecimal number is the ease of conversion from hexadecimal to..... and vice versa.

(A) Decimal

(B) binary

(C) ASCII

(D) BCD

Correct Answer: (B) binary

Solution:

Step 1: Understanding the Question:

The question tests the fundamental reason why the hexadecimal number system is so widely utilized in computer science alongside binary.

Step 2: Detailed Explanation:

Hexadecimal is a base-16 number system, while binary is a base-2 number system.

Since $16 = 2^4$, each individual hexadecimal digit perfectly corresponds to a distinct 4-bit binary sequence, also known as a nibble.

This direct bit-level mapping means that converting between hexadecimal and binary is a

simple, direct substitution process without complex mathematical operations.
Converting to decimal, on the other hand, requires polynomial evaluation and repeated division, which is computationally heavier.
Therefore, the main advantage of hexadecimal representation is the ease of conversion to and from binary, making it a compact way for humans to read long binary machine codes.

Step 3: Final Answer:
The correct choice is binary.

Quick Tip: Hexadecimal is effectively just a shorthand representation for binary; one hex digit compresses exactly four binary bits, drastically reducing the length of strings representing memory addresses and data.

---

5. EBCDIC 8 bit encoding scheme were mainly used by

(A) DELL
(B) HP
(C) IBM
(D) Linux

Correct Answer: (C) IBM

Solution:

Step 1: Understanding the Question:
The question asks us to identify the major technology company associated with the creation and widespread use of the EBCDIC encoding scheme.

Step 2: Detailed Explanation:
EBCDIC stands for Extended Binary Coded Decimal Interchange Code.
It is an 8-bit character encoding scheme that was developed exclusively by IBM in the 1960s.
This specific encoding scheme was predominantly used on IBM mainframe operating systems

and IBM midrange computer architectures.
In contrast, most other computer systems, manufacturers, and platforms eventually adopted the ASCII (American Standard Code for Information Interchange) encoding standard.

Step 3: Final Answer:
The EBCDIC encoding scheme was mainly used by IBM.

Quick Tip: While ASCII is universally used in almost all modern computers and the internet, EBCDIC is a legacy encoding strongly tied to IBM Mainframe environments.

---

6. If a function is defined as static, it means

(A) The value returned by the function does not change
(B) all the variable declared inside the function automatically will be assigned initial value of zero
(C) It should be called only within the same source code / program file.
(D) None of the other choices as it is wrong to add static prefix to a function

Correct Answer: (C) It should be called only within the same source code / program file.

Solution:

Step 1: Understanding the Question:
We need to identify the specific behavior or restriction that applies to a function when it is defined with the **static** keyword in languages like C or C++.

Step 2: Detailed Explanation:
In programming languages like C and C++, the **static** keyword has multiple specific uses depending on its context (variables vs. functions).
When the **static** keyword is applied to a global function definition, it changes the function's linkage type from external linkage to internal linkage.
This restricts the scope and visibility of the function exclusively to the translation unit or the

specific source code file in which it is declared.
Consequently, it cannot be accessed or called from other separate program files, which helps prevent naming conflicts in large, multi-file projects.

Step 3: Final Answer:
It should be called only within the same source code / program file.

Quick Tip: Using **static** on functions is a common and excellent practice in C programming to hide helper functions from the global scope, effectively achieving a form of encapsulation.

---

7. What is recursion?

(A) looping
(B) a function calls another function repeatedly
(C) a function calls repeatedly
(D) function calls itself repeatedly

Correct Answer: (D) function calls itself repeatedly

Solution:

Step 1: Understanding the Question:
The question asks for the fundamental definition of recursion in the context of computer programming and algorithms.

Step 2: Detailed Explanation:
In computer science, recursion is a programming technique and a powerful algorithm design concept.
It specifically occurs when a function solves a problem by making one or more calls to itself during its own execution block.
This process is designed to break down a larger, complex problem into smaller, structurally

identical sub-problems.

A properly implemented recursive function must always have a defined base case to eventually stop the repeated self-calling and prevent an infinite loop, which would lead to a stack overflow error.

Step 3: Final Answer:

Recursion is when a function calls itself repeatedly.

Quick Tip: Every recursive function must contain two vital components: a base case (the termination condition) and a recursive step (the mechanism where the function calls itself with modified parameters).
