QUESTION BANK

Problem Solving in C (KCA-102)

UNIT -1

Short question

Q.1 What is the use of flow chart?
Q.2 What will be the output of the following code for(i=1; i<10;i++); printf("%d", i);
Q.3 Difference between const and static keyword.
Q.4 Explain the standard input/output function in C.
Q.5 What do you mean by precedence and associativity of operators? Explain with a suitable example.

Long Question

Q.6 Define algorithm. What are the attributes of a good algorithm?
Q.7 Differentiate between compiler and interpreter.
Q.8 Explain the various types of Data type and Storage classes?
Q.9 Explain the standard input/output function in C.
Q.10 What do you mean by type casting? Illustrate implicit type casting and explicit type casting with suitable example.

UNIT-2

Short question

Q.11 How do-while loop is different from a while loop? Give the syntax and working of both the loops.
Q.12 Discuss the recursion function.
Q.13 Write a for loop that prints the numbers from 1 to 10 in ascending order.
Q.14 Using the if-else statement, create a program that checks if a given number is even or odd and prints the result.
Q.15 Write a program that uses a while loop to print the Fibonacci sequence up to the 10th term.

Long Questions

Q.16 Write the syntax of switch statement and explain its working. Illustrate with a suitable example.
Q.17 Give the difference between break () and continue () statement with proper example?

Q.18 Explain if –statement and if-else-statement with proper example?
Q.19 Write a function to find factorial of a number.
Q.20 Write a program in C to print the pattern

A

A B

A B C

A B C D

A B C D E

UNIT-3

Short Questions

Q.21 Explain the concept of the "index" in an array. Why is it important?
Q.22 Explain how the pointer variable is declared and initialized. Explain pointer to pointer concept.
Q.23 Explain the difference between the strlen() and sizeof() functions when dealing with strings.
Q.24 Explain the concept of an "array of pointers" and provide a practical example.
Q.25 Discuss the differences between pointer arithmetic and array subscripting. Provide examples to illustrate.

Long question

Q.26 Explain the relationship between pointers and arrays/strings. Discuss how pointers can be used to iterate through arrays and strings. Provide examples demonstrating the advantages of using pointers in these contexts.
Q.27 Describe the concept of function pointers. Provide examples illustrating their use and discuss scenarios where function pointers are particularly useful.
Q.28 Discuss commonly used string functions in C such as strlen, strcpy, strcat, and strcmp. Provide examples of their usage and discuss potential pitfalls or limitations.
Q.29 Implement a function that takes a pointer to a function as an argument and applies that function to each element of an array
Q.30 Implement a function to reverse a string in place (without using additional storage).

UNIT-4

Short Questions

Q.31 What is a structure in C programming, and how is it used to group related data items?

Q.32 How do you initialize, define, and declare a structure in C?

Q.33 How is a union different from a structure, and what are its use cases in C programming?

Q.34 What are the functions of Enumerated data types?

Q.35 What is the purpose of storage classes in C programming, and how do they impact the lifetime and scope of variables?

Long Questions

Q.36 Explain the concept of nested structures in C. Provide examples to demonstrate how structures can be embedded within one another and discuss the advantages of using such structures.

Q.37 Describe the role of pointers in C structures. Explain how pointers can be used to access and manipulate data within structures. Provide practical examples to support your explanation.

Q.38 Discuss the role of the 'register' storage class in C programming. How does it affect variable storage and access, and in what scenarios would you choose to use the 'register' storage class? Provide practical examples to support your explanation.

Q.39 Provide a comprehensive overview of storage classes in C, including their significance in memory management and the different types such as automatic, register, static, and external. Illustrate with examples to enhance understanding.

Q.40 Describe operations that can be performed on unions and how unions contribute to optimizing memory utilization in certain programming situations.

UNIT-5

Short Questions

Q.41 Differentiate between malloc and calloc in terms of their functionality and usage.

Q.42 How does the realloc function contribute to dynamic memory management in C?

Q.43 Explain the concept of a file pointer in C and how it is used in file handling operations.

Q.44 Briefly describe the different file opening modes in C and provide an example for each.

Q.45 What are the key data types commonly used in graphics programming in C, and how do they contribute to graphical representation?

Long Questions

Q.46 Discuss the need for dynamic memory allocation. Illustrate with examples to demonstrate the step-by-step process of allocating memory using malloc, calloc, and static emphasizing their distinctions.

Q.47 Explain the significance of the free function in dynamic memory management and preventing memory leaks.

Q.48 Detail the various file handling operations, including reading, writing, and updating files in C.

Q.49 Explain how command-line arguments can be utilized for file handling in C programs and discuss the advantages of using command-line arguments for file handling.

Q.50 How is the process of drawing and filling images distinct in graphics programming, and why is it significant?