ASSIGNMENT – 7

1. What is recursion in programming?
    a) Repeating a set of instructions using loops
    b) A function calling itself directly or indirectly
    c) Breaking a problem into smaller sub problems
    d) Applying mathematical operations to a variable

2. Which of the following is the base case in a recursive function?
    a) The case where the function calls itself
    b) The first case in the function
    c) The case that terminates the recursion
    d) The case where a loop is used

3. What happens if a recursive function does not have a base case?
    a) The program will produce an error
    b) The recursive function will run indefinitely
    c) The program will terminate immediately
    d) The recursive function will skip the base case

4. Which data structure is commonly used in recursion?
    a) Stack
    b) Queue
    c) Array
    d) Linked List

5. What is the maximum depth of recursion in C?
    a) It depends on the operating system
    b) It depends on the compiler
    c) It depends on the available memory
    d) It is fixed and predetermined

6. What is tail recursion?
    a) A recursive function that calls itself at the beginning of the function
    b) A recursive function that calls itself at the end of the function
    c) A recursive function that calls another function
    d) A recursive function that has multiple base cases

7. When should recursion be used over iteration?
    a) When the problem can be easily solved using loops
    b) When the problem requires breaking it into smaller subproblems
    c) When the problem involves complex mathematical operations
    d) When the problem has a fixed number of iterations

8. Which of the following problems can be efficiently solved using recursion?
    a) Calculating the sum of an array
    b) Finding the maximum element in an array
    c) Sorting an array in ascending order
    d) Generating all possible permutations of a string

9. Which of the following is NOT a disadvantage of recursion?
    a. It can consume more memory compared to iteration
    b. It can be difficult to understand and debug
    c. It can lead to stack overflow for large input sizes
    d. It always results in slower execution time than iteration

10. Write a recursive function to calculate the factorial of a given positive integer. Implement this function in C and test it with different inputs.

11. Write a recursive function to find the nth Fibonacci number, where the nth number is the sum of the (n-1)th and (n-2)th Fibonacci numbers. Implement this function in C and test it with different values of n.

12. Write a recursive function to compute the sum of the digits. For example 1,2,3,4,5, the function should return 15. Implement this function in C and test it with different inputs.

13. Write a recursive function to calculate the power of a number. The function should take two arguments, the base number and the exponent, and return the result of raising the base to the exponent. Implement this function in C and test it with different inputs.

VIDEO

1. https://www.youtube.com/watch?v=kepBmgvWNDw

2. https://www.youtube.com/watch?v=fNl1McayZyY

3. https://www.youtube.com/watch?v=ggk7HbcnLG8

ASSIGNMENT – 8

1. Which operator is used to access the value pointed to by a pointer in C?
    A) * (Asterisk)
    B) & (Ampersand)
    C) -> (Arrow)
    D) . (Dot)
2. What is the purpose of the "sizeof" operator in C?
    A) To determine the size of a variable or data type
    B) To allocate memory dynamically
    C) To perform bitwise operations
    D) To access the address of a variable
3. Which of the following correctly declares a pointer variable in C?
    A) int *ptr;
    B) int ptr;
    C) *int ptr;
    D) pointer int;
4. What is the output of the following code snippet?

```
int  x = 5;
int *ptr = &x;
printf("%d", *ptr);
```

    A) 5
    B) 0
    C) Garbage value
    D) Compilation error
5. How do you pass a pointer to a function in C?
    A) Pass the pointer using call by value
    B) Pass the pointer using call by reference
    C) Pass the value of the pointer
    D) Pointers cannot be passed to functions
6. What is the NULL pointer in C?
    A) A pointer that stores the address of the main function
    B) A pointer that points to the end of a string
    C) A pointer that does not point to any memory location
    D) A pointer that contains the value zero
7. What is the purpose of the "const" keyword in a pointer declaration?
    A) It makes the pointer constant and unmodifiable
    B) It specifies the pointer type

C) It ensures that the pointer is always initialized
D) It allows the pointer to point to any data type

8. What is the result of the following code snippet?

```
int arr[] = {1, 2, 3, 4, 5};
int *ptr = arr;
printf("%d", *(ptr + 2));
```

A) 1
B) 2
C) 3
D) 4

9. What is the purpose of the "->" operator in C?
A) To access a member of a structure through a pointer
B) To compare two pointers
C) To perform logical AND operation on two pointers
D) To access the address of a variable

10. Write a program that takes two integer values from the user and swaps their values using pointers. Print the swapped values after the swap operation.

Example:
Input:
Enter the first integer: 10
Enter the second integer: 20

Output:
After swapping:
First integer: 20
Second integer: 10

11. Write a program that takes an integer array of size 5 as input from the user. Using pointers, find the maximum and minimum elements in the array and print them.

Example:
Input:
Enter 5 integers: 4 9 2 6 1

Output:
Maximum element: 9
Minimum element: 1

12. Write a program that takes an array of integers as input from the user. Using pointers, find the sum and average of all the elements in the array and print them.

    Example:
    Input:
    Enter the number of elements: 5
    Enter 5 integers: 1 2 3 4 5

    Output:
    Sum: 15
    Average: 3.0

ASSIGNMENT – 9

1. What is a structure in C?
    a) A function
    b) A data type that groups related data elements
    c) A loop construct
    d) A library in C

2. How do you declare a structure in C?
    a) struct myStruct;
    b) int myStruct;
    c) myStruct struct;
    d) struct = myStruct;

3. What is the keyword used to access structure members in C?
    a) dot (.)
    b) arrow (->)
    c) colon (:)
    d) exclamation (!)

4. Which of the following statements is true about structures in C?
    a) Structures cannot contain functions.
    b) Structures cannot contain arrays.
    c) Structures can only contain integers.
    d) Structures can contain members of different data types.

5. What is the purpose of the sizeof() operator in C?
    a) It returns the size of a structure in bytes.
    b) It determines the number of members in a structure.
    c) It calculates the sum of all integers in a structure.
    d) It checks if a structure is empty.

6. How can you initialize a structure variable in C?
    a) Using the assignment operator (=).
    b) Using the sizeof() operator.
    c) Using the malloc() function.
    d) Using the new keyword.

7. What is the purpose of typedef in C structures?
    a) It assigns a new name to an existing data type.
    b) It specifies the size of a structure.
    c) It defines a new structure data type.
    d) It declares a structure variable.

8. Which operator is used to access structure members through a pointer in C?
    a) asterisk (*)
    b) ampersand (&)
    c) arrow (->)
    d) dot (.)

9. What is the maximum number of members a structure can have in C?
    a) 10
    b) 100
    c) There is no maximum limit.
    d) It depends on the compiler.
10. What is the purpose of structure in programming?
11. Differentiate between a structure and Union.
12. Explain the concept of nested structures and provide an example.
13. Discuss the advantages of using structures in C programming.
14. What is the difference between a structure and an array?
15. How do you pass a structure to a function in C?
16. Provide an example of a real-life scenario where the use of structures would be beneficial.
17. Create a function called "printStudentInfo" that takes a structure variable of type "Student" as a parameter and prints the details of the student.
18. Write a program that creates an array of structures to store the information of five employees. Prompt the user to enter the details of all employees and display the entered information.
19. Create a function called "printStudentInfo" that takes a structure variable of type "Student" as a parameter and prints the details of the student.
20. Write a program that creates an array of structures to store the information of five employees. Prompt the user to enter the details of all employees and display the entered information.

ASSIGNMENT – 10

1. What is the purpose of dynamic memory allocation in C?
    a) To allocate memory for global variables
    b) To allocate memory for local variables
    c) To allocate memory at compile-time
    d) To allocate memory at runtime
2. Which library function is used to dynamically allocate memory in C?
    a) alloc()
    b) malloc()
    c) new()
    d) allocate()
3. What is the return type of the malloc function in C?
    a) void*
    b) int
    c) char*
    d) float*
4. What is the purpose of the calloc function in C?
    a) To allocate memory for arrays
    b) To allocate memory for structures
    c) To allocate memory for functions
    d) To allocate memory for pointers
5. What is the difference between malloc and calloc in C?
    a) malloc initializes allocated memory to 0, while calloc does not.
    b) calloc initializes allocated memory to 0, while malloc does not.
    c) malloc is used for single variable allocation, while calloc is used for array allocation.
    d) calloc is used for single variable allocation, while malloc is used for array allocation.
6. Which library function is used to reallocate memory in C?
    a) realloc
    b) resize
    c) allocate
    d) adjust
7. What happens if the realloc function fails to allocate memory?
    a) It returns NULL and leaves the original memory block intact.
    b) It returns the original memory block without any changes.
    c) It returns a memory block with zero size.
    d) It raises a runtime error.
8. What is the purpose of realloc in C?

a) To free dynamically allocated memory.
b) To resize dynamically allocated memory.
c) To initialize dynamically allocated memory.
d) To check if dynamically allocated memory is valid.

9. How do you handle an error when malloc or calloc fails to allocate memory?
    a) Print an error message and continue execution.
    b) Terminate the program.
    c) Retry the allocation process.
    d) Skip the memory allocation step.
10. When using malloc, calloc, or realloc, why is it important to release the allocated memory using free?
    a) To prevent memory leaks.
    b) To improve program performance.
    c) To avoid segmentation faults.
    d) To reduce the memory footprint of the program.
11. What is dynamic memory allocation in C?
12. What are the advantages of dynamic memory allocation over static memory allocation?
13. What are the drawbacks of dynamic memory allocation?
14. Explain the purpose of the malloc, calloc and realloc function in C.
15. What is the difference between malloc and calloc?
16. How do you allocate memory for a single variable and an array using malloc?
17. Create a structure called "Person" with members for name (string) and age (integer). Write a program that dynamically allocates memory for a "Person" structure, prompts the user to enter the name and age, and displays the entered information. De allocate the memory when done.
18. Write a program that dynamically allocates an array of integers based on user input for the size. Prompt the user to enter values for each element of the array and display the sum of all the entered values. De allocate the memory afterwards.