**Title: C Programming Language Proficiency Assessment**

**Instructions:**

* This assessment consists of 20 coding questions divided into three levels: Easy, Intermediate, and Hard.
* The questions are designed to evaluate your knowledge of the C programming language.
* Each question is worth a specific number of points, depending on its level of difficulty.
* Your task is to provide the solution for each question and write the prototype of the final function as requested.
* Write your code for each question in the provided space.

**Grading:**

* Easy questions: 1 point each
* Intermediate questions: 2 points each
* Hard questions: 3 points each

**Notes:**

1. The prototype of the final function should be written in C programming language. For example, if the question requires writing a function that adds two integers, the prototype would be: int add(int a, int b);
2. You have to use ListNode.h, ListNode.c, TreeNode.h, and TreeNode.c for the questions requiring them.
3. You MUST submit file called info.txt (already exist in the files i have sent to you) just fill your data like it is written.
4. If you want add any files, then add them in “header dependencies” and “source dependencies” in info.txt
5. Assessment.c shouldn’t include main function!!! I want you to deliver only the implementations of the functions sent to you.
6. Using ChatGPT, Google BARD or any AI tool for help is not allowed!!
7. The question numbers are intended don’t change anything.

**Easy (1 point each):**

* 1. Write a C function to find and return the maximum of two integers.
* **Prototype:** int max(int a, int b);
* 2. Implement a C function to check if a given number is even or odd.
* **Prototype:** int is\_even(int num);
* 3. Create a C function that calculates and returns the factorial of a given positive integer.
* **Prototype:** int factorial(int n);
* 4. Write a C function to check if a given character is a vowel or a consonant.
* **Prototype:** int is\_vowel(char ch);
* 5. Implement a C function that swaps the values of two integer variables.
* **Prototype:** void swap(int \*a, int \*b);

**Intermediate (2 points each):**

6. Create a C function to compute the n-th Fibonacci number using recursion.

**Prototype:** int fibonacci(int n);

* 7. Write a C function to reverse an array of integers in-place.
* **Prototype:** void reverse\_array(int arr[], int size);
* 8. Implement a C function to find and return the largest element in an integer array.
* **Prototype:** int find\_max(int arr[], int size);
* 9. Create a C function to check if a given string is a palindrome.
* **Prototype:** int is\_palindrome(char str[]);
* 10. Write a C function that sorts an array of integers in ascending order using the bubble sort algorithm.
* **Prototype:** void bubble\_sort(int arr[], int size);

11. Create a C function that calculates and returns the sum of elements in an integer array.

**Prototype:** int array\_sum(int arr[], int size);

* 12. Write a C function to check if a given number is prime.
* **Prototype:** int is\_prime(int num);
* 13. Implement a C function to find and return the second largest element in an integer array.
* **Prototype:** int find\_second\_largest(int arr[], int size);
* 14. Create a C function to compute the power of a given base raised to an exponent using recursion.
* **Prototype:** int power(int base, int exponent);
* 15. Write a C function that finds and returns the first occurrence of a target value in an integer array.
* **Prototype:** int find\_target(int arr[], int size, int target);

**Hard (3 points each):**

31. Implement a C function to find and return the k-th smallest element in an unsorted integer array.

**Prototype:** int find\_kth\_smallest(int arr[], int size, int k);

* 32. Write a C function that converts a binary search tree (BST) to a sorted doubly linked list.
* **Prototype:** struct TreeNode\* convert\_to\_dll(struct TreeNode\* root);
* 33. Create a C function to compute the factorial of a given positive integer using recursion.
* **Prototype:** int factorial\_recursive(int n);
* 34. Implement a C function that merges two sorted arrays into a single sorted array.
* **Prototype:** void merge\_sorted\_arrays(int arr1[], int size1, int arr2[], int size2, int result[]);
* 35. Write a C function to reverse a linked list in-place.
* **Prototype:** struct ListNode\* reverse\_linked\_list(struct ListNode\* head);

Ensure that your solutions are efficient, free from errors, and follow the specified prototype. Good luck!