tower of hanoic

Absolutely, let's outline the aim, procedure, and C code for the Tower of Hanoi problem.

Aim

To implement a C program that solves the Tower of Hanoi puzzle, demonstrating the recursive solution and calculating the minimum number of moves required.

Procedure

1. **Recursive Solution:**
    - **Base Case**: If there's only one disk (n == 1), simply move it from the source peg to the destination peg.
    - **Recursive Steps**:
        - Move the top n-1 disks from the source peg to the auxiliary peg, using the destination peg as a temporary holding area.
        - Move the largest disk (the nth disk) from the source peg to the destination peg.
        - Move the n-1 disks from the auxiliary peg to the destination peg, using the source peg as a temporary holding area.
2. **Main Function:**
    - Prompt the user to enter the number of disks.
    - Call the recursive `tower_of_hanoi` function to solve the puzzle.
    - Calculate and print the total number of moves (2^n - 1).

Code

C

```
#include <stdio.h>
#include <math.h>

void tower_of_hanoi(int n, char source, char temp, char destination) {
    if (n == 1) {
        printf("Move disk 1 from %c to %c\n", source, destination);
        return;
    }

    tower_of_hanoi(n - 1, source, destination, temp);
    printf("Move disk %d from %c to %c\n", n, source, destination);
    tower_of_hanoi(n - 1, temp, source, destination);
```

```
}

int main() {
    int n;

    printf("Enter the number of discs: ");
    scanf("%d", &n);

    tower_of_hanoi(n, 'A', 'B', 'C');
    printf("\nTotal number of moves: %d\n", (int)pow(2, n) - 1);

    return 0;
}
```

Let me know if you'd like a deeper dive into the recursion or want to explore any enhancements to this code!