```c
#include <stdio.h>

int main(void)
{
    //variable declarations
    int num = 5;
    const int* const ptr = &num; // Read this line from right to left => "ptr is a constant (const) pointer (*) to integer (int) constant (const)."

    //code
    printf("\n");
    printf("Current Value Of 'num' = %d\n", num);
    printf("Current 'ptr' (Address of 'num') = %p\n", ptr);

    // The following line does NOT give error ... as we are modifying the value of the variable individually
    num++;
    printf("\n\n");
    printf("After num++, value of 'num' = %d\n", num);


    // The following line gives error and is hence commented out.
    // We cannot alter the 'ptr' value as 'ptr' is "a constant pointer to constant integer".
    // With respect to the pointer, the value it points to is constant AND the pointer itself is also constant.
    // Uncomment it and see the error.

    //  ptr++;

    // The following line also give error and is hence commented out.
    // We cannot alter the value stored in 'ptr' as 'ptr' is "A constant pointer to constant integer"
    // With respect to the pointer, the value it points to is constant AND the pointer itself is also constant.
    // Uncomment it and see the error.

    //  (*ptr)++;

    return(0);
}

// CONCLUSION :
// As "ptr" is a "constant pointer to a constant integer" - we cannot change the value stored at address "ptr" AND we cannot change the 'ptr' (Address) itself.
// We can change the value of the variable (num) individually - whose address is contained in "ptr".
// We cannot also change the "the value at address of ptr" - we cannot change the value of "num" with respect to "ptr" => (*ptr)++ is NOT allowed
// We cannot change the value OF 'ptr' => That is we cannot store a new address inside 'ptr' => So, ptr++ is also NOT allowed
```