Most DAQ (Data Acquisition) cards return the voltages read by the ADC (analog to digital converter) as an integer value that is proportional to the actual voltage measured. The conversion of this integer value, \( k \), to the actual voltage, \( v \), depends on the number of bits used by the ADC and its input voltage range. For this exercise, we assume that a 16-bit ADC is used, i.e., we can distinguish between \( 2^{16} = 65536 \) voltage levels, and a 10-V input range is used. This results in the following conversion equation:

\[
v(k) = \frac{10k}{65536} - 5 \tag{1}
\]

a) Write a C-function called \( DtoV( k ) \) that calculates and returns the corresponding voltage for a given integer value, \( k \), based on the equation shown above. You are not allowed to use ANY global variables in your function! Specifically, pass the argument, \( k \), (type “int”) by value; calculate the voltage and return it (as a type “double”) by the function itself. (6 points)

b) Assume that \( n \) (digital) readings were previously taken by the DAQ and stored in an array called \( DRead \). Write a second C-function, called \( DtoVArray( DRead, VRead, n) \), that converts each integer value in the array \( DRead \) to its corresponding voltage and then stores the result in an array \( VRead \), as type “double.” The order remains preserved, i.e., the \( i \)th array element of \( DRead \) corresponds to the \( i \)th element in the array \( VRead \). Additionally:
   1. Pass the arguments for the arrays by reference;
   2. You must call your function \( DtoV(k) \), written in part a) for the conversion;
   3. The function itself does not return anything and, therefore, is of type “void.”

Again, you are not allowed to use ANY global variables in your function! (9 points)

Additional Comments for part a) and b)
- Write the entire functions, including function prototypes, headers, body and return statement, if applicable.
- Do not include any input or output statements such as \( scanf \) or \( printf \) or any include files.
- You may write the C-code using as many or as few additional local variables as you consider necessary.
- You are not asked to write a complete program, so do not show “main”!
You will be graded on program logic and syntax mistakes.

**SOLUTION:**

double DtoV( int k);        //Funct. Prototypes
void DtoVArray( int* DRead, double* VRead, int n );

double DtoV( int k)
{
    return (10.0*k/65536.0)-5.0;
}

void DtoVArray( int* DRead, double* VRead, int n )
{
    int i;
    for( i = 0; i < n; i++)
        VRead[i] = DtoV( DRead[i] );
}