# Problem Set 03 - Pointers

Write a C++ programs named '`main31.cpp`' that performs the following tasks:

## Tasks:

1. Define a struct named *Entry* that contains a string member named *category*, a double pointer member named *values*, and an int member named *count*.
2. Define a *Entry* pointer function named `NewEntry()` that takes no parameters. Initially, it prompts the user to enter a name for the category and a positive number for the count of the values. Afterward, it allocates a *Entry* object, assigns the name and number provided to the *category* and *count* members of the object, respectively, allocates *values* to a double array of size *count*, then returns the object if the number provided is between 1 and 20 inclusive. Otherwise, it returns a null pointer.
3. Define a void function named `Populate()` that takes a *Entry* pointer parameter. If the parameter is not null, it prompts the user for values to assign to the elements of the *values* member of the parameter one at a time. It prohibits negative values, thus it continually prompts for values until a valid value is provided.
4. Define a void function named `Print()` that takes a *Entry* pointer parameter. If the parameter is not null, it displays the *category* member of the parameter on a separate line then, the value of each element of the *values* member of the parameter on separate lines with the string '`Value` *x*`:` ' preceding it where *x* is the element's position.
5. Define a void function named `KillEntry()` that takes a *Entry* pointer reference parameter. If the parameter is not null, it deallocates the *values* member and then, deallocates the parameter.
6. Define the main function so that it declares a *Entry* pointer, assigns the object a `NewEntry()` caller, invokes `Populate()`, `Print()`, and `KillEntry()` in that order with the object as their argument.