The first function takes a double float as input and returns the result given by qfloat.dll as float.
Since Excel displays only 15 trailing digits this will end with a 0, even if 16 digits are used for formating:

x	cdfN(x)
-4,2	1,3345749015906300E-05

To have acces to the libraries capabilities the next takes a float as input and returns the result as string

-4,2	1.33457490159063278826999633294344296505824379192332962782810836659924641015811834284940993900190520254561E-5

One can see: the rounding is correct.

But there is an issue: -4.2 as a float number is not represented as such, for a compiler it reads like -4.20000000000000016 and qfloat.dll will just take that as input for its processing (since it attempts to be as exact as possible). This may cause a false interpretation of the result.

So one needs a string as input, this will be seen by qfloat as it is entered:

"-4.2"	1.33457490159063383530921177856273702507127391679764436207208678805135530934144568658209720285026032240913E-5

And actually one can recognize different results!

There are some options in Excel (like regarding language dependencies and automatic conversions for dates, which may even depend on settings for the operating system). One way to work with that is to format cells as being "text" for its numbers as input (enter numbers in scientific notation if needed):

-4,2	1.33457490159063383530921177856273702507127391679764436207208678805135530934144568658209720285026032240913E-5

Note that now both input and output are formated as "text" (check it through format/cell/numbers).

A cross-check with Maple (using a very high precsion) gives the following exact result for x = - 4.2 :

1.33457490159063383530921177856273702507127391679764436207208678805135530934144568658209720285026030765359E-5

The difference for the last 7 decimal digits should have the following reason (1 + 106 digits are shown here):

The above number in usual decimal notation starts with 4 leading zeros after the decimal point followed by 13345 ... 6030765359. This has length = 4 + 107 = 111 after the decimal point. The qfloat will take care for the first 104 decimals after the decimal point. So one has to ignore the last 111 - 104 = 7 decimal ciphers.

Pass to the VBA project by <Alt> + F11 and open the debug window to find the VBA code and watch tests.