Objects: Practice implementing recursive functions and their faster dynamic-programming equivalent.

Highly Recursive Function
Consider the following recursive mathematical function, H(n):
\[ H(n) = H(n+5) + H(n+4) + H(n+2) \quad \text{for all value of } n \leq -8 \]
\[ H(n) = n \quad \text{for all value of } -8 < n < 10 \]
\[ H(n) = H(n-8) + H(n-5) + H(n-3) \quad \text{for all values of } n \geq 10. \]

I want you to write a program to compute values of the function H(n) two ways:

a) recursive Python function \( H_{\text{rec}}(n) \), and

b) non-recursive Python function \( H_{\text{dynPgm}}(n) \) that uses dynamic-programming. To transform from the recursive view of the problem to the dynamic programming solution you can do the following steps:
1) Store the solution to smallest problems (i.e., the base cases) in a dictionary or list
2) Loop (no recursion) from the base cases up to the biggest problem of interest. On each iteration of the loop we:
   • solve the next bigger problem by looking up the solution to previously solved smaller problem(s)
   • store the solution to this next bigger problem for later usage so we never have to recalculate it

Some known test cases are:
\[
\begin{align*}
H(-8) &= -13 \\
H(10) &= 14 \\
H(-13) &= -58 \\
H(-4) &= -4
\end{align*}
\]

Complete the hw4.py tester found in hw4.zip on the Homework website:
http://www.cs.uni.edu/~fienup/homework

Submit your homework electronically at: https://www.cs.uni.edu/~schafer/submit/which_course.cgi

Submit a single zipped file, hw4.zip containing your:
• hw4.py (your Python program)