To implement a DNS resolver with DNSSEC verification, for each iterative query to a DNS server, the "want_dnssec" parameter in query message needs to be set as "True" and then get two types of information, i.e., "DNSKEY" and "A" as well as their digital signatures. The iterative query is very similar to that in PART A and PART B just add a verification function.

When the program gets the RRST of "A" record from the last DNS server, its signature can be verificated by the "DNSKEY" record. The "DNSKEY" record also has a digital signature and thus also needed to be verificated. The "DS" record can be got in the "authority" part of a DNS response and its digital signature can also be verificated by "DNSKEY".

The KSK verification is more complicated. The program stored "DS" record from the DNS server in the upper level, and then compared with the hash value with the "DNSKEY" record retrieved from the DNS server in current level. For root DNS server, they have no upper level DNS servers and their "DNSKEY" are verificated by their own "RRSIG" record.

There will be three cases, i.e., "DNSSEC is not supported", "DNSSEC is configured but verification failed", and "DNSSEC is configured and verification succeeded". Obviously, if the program cannot find the "RRSIG" of each records or the "DNSKEY" doesn't exist, the website will be considered as not supporting DNSSEC. If those records can be found, as long as one of the verifications mentioned above failed, the program will consider the whole verification failed. If everything is OK, then the program will classify the website into the third case.

This program adopts multiple threads to accelerate queries but personally I still think it is too slow. Besides, this program use TCP to send query message in case of hijack but it is not guranteed that the program can always retrieve demanded records from a DNS server.