CODE DOCUMENTATION

Introduction to Computer Communications
Programming Assignment 2 - DNS Client

Liron Cohen 207481268, Yuval Mor 209011543

Running Instructions

Running the program is as required in the assignment.

Header File

- Defines, includes and pragmas.
- **Struct header** - defines the fields of header part in DNS packet, according to RFC, ordered by byte.
- **Struct question** - defines the fields of question part in DNS packet, according to RFC (without QNAME).
- **Struct r_data** - defines the fields of rdata part in resource record of DNS packet, according to RFC.
- **Struct res_record** - defines the fields of resource record part in DNS packet, according to RFC.
- **Struct query** - defines the fields of query part in DNS packet, including QNAME and question.

Main Flow

- Parsing arguments and initializing Winsock.
- Creating socket and server address using **createSocketAndServerAddr** method
  - Creating socket.
  - Setting timeout to socket according to instructions.
  - Creating server address and setting server IP and port.
- Asking user for domain name. Lower-casing the input using **domainToLowercase** method.
- While user hasn't entered "quit":
  - Checking if the domain name is valid using **isDomainNameValid** method. Validation is based on rules from https://www.geeksforgeeks.org/how-to-validate-a-domain-name-using-regular-expression/ (checking length, legal characters, etc). If the name is not valid, printing relevant message to the user.

- o Sending a dnsQueryparsing the result into struct hostent* using **dnsQuery** method.
    - ▪ Creating DNS query packet using **createDnsQueryPacket** method, by filling the relevant structs according to instructions.
    - ▪ Sending the query packet to socket using sendto and check the returned value for errors.
    - ▪ Receiving the answer packet from socket using recvfrom and check the returned value for errors and timeout.
    - ▪ Parsing the answer packet using **parseAnswerFromAnswerPacket** method and returning the result.
        - • Checking the RCODE result, if it's not error printing the relevant error message.
        - • Going over the resource records and if they include IP address, reading it and parsing it into the relevant fields in the result struct.
  - o Printing the result (or results, see bonus part) and asking for another input.
- • Closing socket, cleaning up Winsock and exits successfully.

Bonus - Implementation for multiple addresses for the same domain name

- • Hereby listed the relevant additions and changes that were made to support multiple addresses for the same domain name:
  - o When parsing the answer packet result, we don't fill the struct and return it after one address was found, but going over all of the resource records in the packet.
  - o When writing into the relevant field in the hostent struct, we concatenate the IP addresses we found so far, and adding line separator between them, so when we print the field we get all of the addresses printed one after one.