```
import socket
import sys
import select

MAGIC_NUMBER = 0x497E

def check_args(argv):
    """Checks that the arguments given are valid, and returns the type of
       information the user wants to send and the required socket information"""
    if len(argv) > 3:
        print("Too many arguments")
        sys.exit()
    elif len(argv) < 3:
        print("Too few arguments")
        sys.exit()

    request_type = argv[0]
    host = argv[1]
    port = argv[2]

    if not (request_type == "date" or request_type == "time"):
        print("First argument must be 'date' or 'time'")
        sys.exit()
    try:
        sock_family, sock_type, _, _, address = socket.getaddrinfo(host, port)[1]
    except: #It can raise multiple types of error
        print("Destination address incorrect")
        sys.exit()

    return request_type, sock_family, sock_type, address


def check_response(packet):
    """Checks whether the given response is a valid DT-Response packet"""
    #Check packet length
    if not len(packet) >= 13:
        print("Response packet too short ({})".format(len(packet)))
        sys.exit()

    #Check magic number
    magicNum = packet[0] << 8 | packet[1]
    if magicNum != MAGIC_NUMBER:
        print("Magic number not correct ({})".format(magicNum))
        sys.exit()

    #Check PacketType field
    packetType = packet[2] << 8 | packet[3]
    if packetType != 2: #0x0002
        print("Packet type not valid ({})".format(packetType))
        sys.exit()

    #Check LanguageCode field
    languageCode = packet[4] << 8 | packet[5]
    if not languageCode in [1, 2, 3]: #0x0001, 0x0002, 0x0003
        print("Language code not valid ({})".format(languageCode))
        sys.exit()

    #Check year is in the correct range
    year = packet[6] << 8 | packet[7]
    if year >= 2100:
        print("Year field out of range ({} is not less than 2100)".format(year))
        sys.exit()

    #Check month is in the correct range
    if not (packet[8] >= 1 and packet[8] <= 12):
        print("Month field out of range ({} is not between 1 & 12)".format(packet[8]
))
        sys.exit()

    #Check day is in the correct range
    if not (packet[9] >= 1 and packet[9] <= 31):
        print("Day field out of range ({} is not between 1 & 31)".format(packet[9]))
        sys.exit()
```

```
    #Check hour is in the correct range
    if not (packet[10] >= 0 and packet[10] <= 23):
        print("Hour field out of range ({} is not between 0 & 23)".format(packet[10]
))
        sys.exit()

    #Check minute is in the correct range
    if not (packet[11] >= 0 and packet[11] <= 59):
        print("Minute field out of range ({} is not between 0 & 59)".format(packet[1
1]))
        sys.exit()

    #Check that the length field is equal to the length of the received packet
    if packet[12] != len(packet):
        print("Length field ({0}) does not equal actual packet length ({1})".format(
packet[12], len(packet)))
        sys.exit()

    return True

def print_resp_packet(packet):
    """Prints out all the contents of the response packet"""
    print("Magic Number = {}".format(hex(packet[0] << 8 | packet[1])))
    print("Packet Type = {}".format(packet[2] << 8 | packet[3]))
    print("Language Code = {}".format(packet[4] << 8 | packet[5]))
    print("Year = {}".format(packet[6] << 8 | packet[7]))
    print("Month = {}".format(packet[8]))
    print("Day = {}".format(packet[9]))
    print("Hour = {}".format(packet[10]))
    print("Minute = {}".format(packet[11]))
    print("Packet length = {}".format(packet[12]))
    print(packet[13:].decode('utf-8'))

def send_packet(sock_family, sock_type, address, packet):
    """Creates the socket, send the packet, and awaits a response"""
    sock = socket.socket(sock_family, sock_type)    #Create socket
    sock.sendto(packet, address)                    #Send packet
    readable, _, _ = select.select([sock], [], [], 1)
    if readable:
        resp_packet, resp_address = sock.recvfrom(4096)
        if check_response(resp_packet):
            print_resp_packet(resp_packet)
            sock.close()
            sys.exit()
    else:
        print("No response recieved")
        sys.exit()

def create_packet(request_type):
    """Creates a valid DT-Request packet and returns it"""
    packet = bytearray(6)

    #Add the magic number to the packet
    magicNum_16 = format(MAGIC_NUMBER, "016b")
    packet[0] = int(magicNum_16[:8], 2)
    packet[1] = int(magicNum_16[8:], 2)


    #Add the PacketType information to the packet
    packet[2] = 0 #Since it will always be lead by 8 0s
    packet[3] = 1 #Since this is will be 7 0s followed by a 1

    #Add the RequestType information to the packet
    packet[4] = 0 #Since it will always be lead by 8 0s
    if request_type == 'date':
        packet[5] = 1
    elif request_type == 'time':
        packet[5] = 2

    return packet

def main(argv):
```

```
    request_type, sock_family, sock_type, address = check_args(argv)
    packet = create_packet(request_type)
    send_packet(sock_family, sock_type, address, packet)


if __name__ == '__main__':
    main(sys.argv[1:]) #Cuts out the 'client.py' argument
```