```
"""
Server program for communication via UDP sockets. Takes three port numbers to
bind sockets to and receive communication on. Waits for a request packet from
the client and then rsends a response packet back with appropriate information.
Author: Vikas Shenoy vsh33
Date: 16/8/2018
"""
import socket
import select
import sys
import datetime
from sys import argv



def isValid(packet):
    """Takes a bytearray request packet and checks if it's valid. Returns
    a boolean true if valid, false if not. Prints the packet contents."""

    valid = True
    magicNumber = packet[0] << 8 | packet[1]
    packetType = packet[2] << 8 | packet[3]
    requestType = packet[4] << 8 | packet[5]
    length = len(packet)

    if length != 6:
        print("Request packet must be 6 bytes long.")
        valid = False
    elif magicNumber != 0x497E:
        print("Magic number must be 0x497E")
        valid = False
    elif packetType != 0x0001:
        print("Packet type must equal 0x0001")
        valid = False
    elif requestType > 2 or requestType < 0:
        print("Request type must equal 0x0001 or 0x0002")
        valid = False

    # Print packet contents for debugging purposes
    print("Request packet contents: ")
    print("Magic number = {}".format(hex(magicNumber)))
    print("Packet type = {}".format(packetType))
    print("Request type = {}".format(requestType))
    return valid



def getMonthName(monthNum, languageCode):
    """Takes the current month number and language requested by the user.
    Returns the month's proper noun name."""
    monthNameList = [None,
                     [None, 'January', 'February', 'March', 'April', 'May',
                      'June', 'July', 'August', 'September', 'October',
                      'November', 'December'],
                     [None, 'Kohit¿tea', 'Hui-tanguru', 'Poutu-te-rangi',
                      'Paenga-whawha', 'Haratua', 'Pipiri', 'Hongongoi',
                      'Here-tuki-koka', 'Mahuru', 'Whiringa-a-nuku',
                      'Whiringa-a-rangi', 'Hakihea'],
                     [None, 'Januar', 'Februar', 'Marz', 'April', 'Mai', 'Juni',
                      'Juli', 'August', 'September', 'Oktober', 'November',
                      'Dezember']]
    return monthNameList[languageCode][monthNum]



def constructResponse(requestType, languageCode):
    """Takes a request type (date or time), and language code. Returns
    an appropriate response packet to send back to the client."""
    # Constructs the response packet header
    packet = bytearray(13)
    dateTime = datetime.datetime.now()
    magicNumber = 0x497E
    packetType = 0x0002
```

```
    year = dateTime.year
    monthNum = dateTime.month
    day = dateTime.day
    hour = dateTime.hour
    minute = dateTime.minute

    packet[0] = (magicNumber >> 8) & 0xff
    packet[1] = magicNumber & 0xff
    packet[2] = (packetType >> 8) & 0xff
    packet[3] = packetType & 0xff
    packet[4] = (languageCode >> 8) & 0xff
    packet[5] = languageCode & 0xff
    packet[6] = (year >> 8) & 0xff
    packet[7] = year & 0xff
    packet[8] = monthNum
    packet[9] = day
    packet[10] = hour
    packet[11] = minute
    packet[12] = 0

    message = ''
    monthName = getMonthName(monthNum, languageCode)
    # date requested
    if requestType == 0x0001:
        if languageCode == 0x0001:
            message = "Today's date is {} {}, {}".format(monthName, day, year)
        elif languageCode == 0x0002:
            message = "Ko te ra o tenei ra ko {} {}, {}".format(monthName
                                                                , day, year)
        else:
            message = "Heute ist der {}. {} {}".format(day, monthName, year)
    # time requested
    else:
        if languageCode == 0x0001:
            message = "The current time is {0}:{1:02d}".format(hour, minute)
        elif languageCode == 0x0002:
            message = "Ko te wa o tenei wa {0}:{1:02d}".format(hour, minute)
        else:
            message = "Die Uhrzeit ist {0}:{1:02d}".format(hour, minute)

    # Add the message and return the packet
    messageBytes = message.encode('utf-8')
    packet[12] = len(messageBytes)
    for byte in messageBytes:
        packet.append(byte)
    return packet



def serverLoop(s1, s2, s3, p1, p2, p3):
    """Takes three sockets and three port numbers. Waits for a request packet
    from the client. If valid, returns a response packet with the correct
    info."""
    bufferSize = 1024
    while True:
        print("Server waiting for request packets...")
        readList, writeList, error = select.select([s1, s2, s3], [], [], None)
        if len(readList) > 0:
            clientSkt = readList[0]
            requestPkt, addr = clientSkt.recvfrom(bufferSize)
            packetValid = isValid(requestPkt)
            clientPort = clientSkt.getsockname()[1]

            # Construct and send the response packet
            if packetValid:
                requestType = requestPkt[4] << 8 | requestPkt[5]
                languageCode = 0x0001
                if clientPort == p1:
                    languageCode = 0x0001
                elif clientPort == p2:
                    languageCode = 0x0002
                elif clientPort == p3:
                    languageCode = 0x0003
```

```
                responsePkt = constructResponse(requestType, languageCode)
                clientSkt.sendto(responsePkt, addr)
                print("Reponse packet sent")
            else:
                print("Packet is invalid and has been discarded")


def runServer(port1, port2, port3):
    """Takes three port numbers which are checked for correctness. Creates
    sockets which are bound to these ports, then starts the server loop."""
    # Port numbers checked for correcteness
    if (port1 == port2 or port1 == port3 or port2 == port3):
        print("Port numbers must be unique.")
        return
    elif (port1 < 1024 or port2 < 1024 or port3 < 1024):
        print("Port numbers must be between 1024 and 64000")
        return
    elif (port1 > 64000 or port2 > 64000 or port3 > 64000):
        print("Port number must be between 1024 and 64000.")
        return

    # Create sockets and bind to the given ports
    skt1 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    skt2 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    skt3 = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

    try:
        skt1.bind(('', port1))
        skt2.bind(('', port2))
        skt3.bind(('', port3))
    except:
        print("Failed to bind sockets to given port numbers")
        return
    serverLoop(skt1, skt2, skt3, port1, port2, port3)


def main():
    name, port1, port2, port3 = argv
    runServer(int(port1), int(port2), int(port3))
main()
```