```
from socket import *
import sys
import time

HOST = ''
PORT = 5008
SERVERHOST = ''
SERVERPORT = 5009
ADDRESS = "192.168.101.255"
WAITINGTIME = 10

def get_alivings(clientsocket):
    serversock = socket()
#    serversock.setsockopt(SOL_SOCKET, SO_REUSEADDR, 1)
    serversock.bind((SERVERHOST, SERVERPORT))              # IP と PORT を指定してバインド

    serversock.listen(300)                                 # 接続の待ち受けを準備する（キューの最大数を指定）
    print('Waiting for connections...')

    servernum = 0                                          # 生きてる（＝応答のある）サーバの数
    tstart = time.time()

    clientsocket.sendto(b'raspi_resreq', (ADDRESS, PORT))  # 応答リクエストをブロードキャストする

    while True:
        clientsock, client_address = serversock.accept()   # クライアントとの接続
        servernum += 1
        print("Received[", servernum, "] from", client_address, ": ", clientsock.recv(4096))

        clientsock.close()                                 # クライアントの切断

        tcurrent = time.time()
        pasttime = tcurrent - tstart
        print(f"Past time: {pasttime}")

        if (pasttime > WAITINGTIME):
            break                                          # 最大待ち時間を過ぎたら受付終了

    serversock.close()                                     # サーバソケットを閉じる

    return servernum

s = socket(AF_INET, SOCK_DGRAM)
s.setsockopt(SOL_SOCKET, SO_BROADCAST, 1)
s.bind((HOST, PORT))

while True:
#    msg = raw_input("> ")
#    s.sendto(msg, (ADDRESS, PORT))
    msg = input("> ")
    s.sendto(msg.encode(), (ADDRESS, PORT))

    if msg == ".":
        break

    if msg == "reboot":
        s.sendto(b'raspi_reboot', (ADDRESS, PORT))
        print('Paspberry-pi servers are rebooting')
        break

    if (msg == "shutdown" or msg == "halt"):
        while True:
            print('Raspberry-pi servers are shutting down')
            s.sendto(b'raspi_shutdown', (ADDRESS, PORT))
            if (get_alivings(s) == 0):
                break
            sleep(5)

    if (msg == "alivings"):
        print("Alivings:", get_alivings(s))

s.close()
sys.exit()

print('Done')
```