import socket  
import os  
  
# http://data.pr4e.org/romeo.txt  
  
#1  
'''  
url = input('input url: ')  
word = url.split('/')  
host = word[2]  
print(host)  
try:  
 mysock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
 mysock.connect(('host', 80))  
 mysock.send(('GET '+url+' HTTP/1.0rnrn').encode())  
  
except:  
 print ("Try your best")  
  
while True:  
 data = mysock.recv(512)  
 if len(data) < 1:  
 break  
 print(data.decode(),end='')  
'''  
  
#2  
'''  
url = input('input url: ')  
mysock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
HOST = url.split('/')[2]  
mysock.connect((HOST, 80))  
cmd = ('GET' + ' ' + url + ' ' + 'HTTP/1.0\r\n\r\n').encode()  
mysock.send(cmd)  
  
count = 0  
while True:  
 data = mysock.recv(512)  
 count += len(data)  
 if (len(data) < 1) or (count >= 3000): break  
 print(data.decode(), end='')  
mysock.close()  
print(count)  
'''  
  
#3  
'''  
import urllib.request  
  
url = input('input url: ')  
fhand = urllib.request.urlopen(url)  
chars = 0  
char\_limit = 3000  
for line in fhand:  
 line = line.decode()  
 next\_count = chars + len(line)  
 if next\_count <= char\_limit:  
 print(line.rstrip('\n'))  
 elif chars < char\_limit:  
 char\_remain = char\_limit - chars - 1  
 print(line[:char\_remain])  
 chars = next\_count  
print(chars)  
'''  
  
#4  
  
'''  
import urllib.request  
import urllib.parse  
import urllib.error  
import ssl  
from bs4 import BeautifulSoup  
  
count = 0  
ctx = ssl.create\_default\_context()  
ctx.check\_hostname = False  
ctx.verify\_mode = ssl.CERT\_NONE  
  
url = input('input url: ')  
html = urllib.request.urlopen(url, context=ctx).read()  
soup = BeautifulSoup(html, 'html.parser')  
  
tags = soup('p')  
for tag in tags:  
 count += 1  
print(count)  
'''  
  
#5  
'''  
my\_sock = socket.socket(socket.AF\_INET, socket.SOCK\_STREAM)  
my\_sock.connect(('data.pr4e.org', 80))  
cmd = 'GET http://data.pr4e.org/romeo.txt HTTP/1.0\r\n\r\n'.encode()  
my\_sock.send(cmd)  
  
data = my\_sock.recv(512)  
message = data.decode()  
header\_end\_pos = message.find('\r\n\r\n') + 4  
print(message[header\_end\_pos:], end='')  
while True:  
 data = my\_sock.recv(512)  
 if not data:  
 break  
 print(data.decode())  
my\_sock.close()  
'''