Ngân hàng câu hỏi thi Lập Trình Mạng

1. Thư viện liên kết động của Windows Socket có tên là:
   a. WinSock.DLL
   b. WinSock2.DLL
   c. WS2.DLL
   d. WS2_32.DLL
2. Để có thể sử dụng thư viện Windows Socket, tệp tiêu đề cần khai báo là:
   a. Winsock.h
   b. Winsock2.h
   c. Ws2_32.h
   d. Ws2.h
3. Để có thể sử dụng dụng thư viện Windows Socket, tệp thư viện cần khai báo cho quá trình liên kết là.
   a. Winsock.lib
   b. WS2_32.LIB
   c. WS2_32.DLL
   d. Winsock2.dll
4. Công cụ chuẩn đoán mạng dùng để hiển thị các kết nối hiện có trên máy tính là
   a. Wireshark
   b. TCPView
   c. Netcat
   d. Tất cả các công cụ trên
5. Công cụ dùng để theo dõi tài nguyên sử dụng của máy tính trên hệ điều hành Windows là
   a. Task Manager
   b. Resource Monitor
   c. Wireshark
   d. Phương án a và b.
6. Công cụ dùng để bắt các gói tin gửi ra và vào một giao diện mạng của máy tính là
   a. Netcat
   b. Network Monitor
   c. Cain
   d. Không phương án nào đúng.
7. Công cụ đa năng, vừa có thể đóng vai trò client, vừa server, chạy trên giao thức TCP, UDP là.
   a. Netcat
   b. TCPDump
   c. Netstat
   d. Network Monitor
8. Nếu cần phải viết một ứng dụng cần tính đáp ứng nhanh và chấp nhận sai sót. Giao thức lựa chọn sẽ là
   a. TCP
   b. IP
   c. ICMP
   d. UDP
9. Dịch vụ phân giải tên miền chạy ở tầng nào trong các tầng sau đây
   a. Application
   b. Transport
   c. Internetwork
   d. Datalink
10. Giao thức nào được sử dụng để chia sẻ một địa chỉ toàn cục cho một nhóm các máy tính trong mạng LAN.
    a. VLAN
    b. MAC
    c. NAT
    d. Không đáp án nào đúng
11. Trong một mạng máy tính được chia sẻ chung một địa chỉ IP toàn cục. Thiết lập nào sau đây sẽ cho phép một máy tính từ Internet chủ động kết nối đến một máy chủ trong mạng.
    a. DMZ
    b. Virtual Server
    c. Port Forwarding
    d. Cả ba phương án trên.
12. Công cụ nào sử dụng để kiểm tra hoạt động của một máy chủ phân giải tên miền
    a. Ping
    b. Nslookup
    c. Netstat
    d. Ipconfig
13. Nếu cần phát triển thêm trình điều khiển cho một loại thiết bị mới. Ứng dụng sẽ phải viết ở tầng nào của Windows Socket
    a. Application
    b. Provider
    c. Transport Protocol
    d. Không tầng nào đúng
14. Nếu cần phải thiết kế một giao thức mới, ứng dụng sẽ phải tích hợp vào tầng nào của Windows Socket API
    a. Application
    b. Provider
    c. Transport Protocol
    d. Không tầng nào đúng.
15. Trình điều khiển cho một thiết bị mạng chạy ở chế độ
    a. User Mode
    b. Kernel Mode
    c. System Mode
    d. Không phương án nào đúng.
16. Biên của thông điệp trong các giao thức hướng dòng có được bảo toàn hay không ?
    a. Có
    b. Không
    c. Tùy từng trường hợp, nếu được tổ chức hợp lý.
17. Hàm nào sau đây thực hiện công việc khởi tạo Windows Socket
    a. WSStartup
    b. WinsockStartup
    c. SocketStartup
    d. WSAStartup
18. WSADATA là cấu trúc dùng để
    a. Truyền thông tin về phiên bản WinSock mà ứng dụng muốn khởi tạo
    b. Nhận thông tin về phiên bản WinSock có trên hệ thống.
    c. Cả hai đáp án đều sai.
19. Hàm nào sau đây sử dụng để giải phóng Windows Socket API.
    a. WSACleanup
    b. WSCleanup
    c. Cleanup
    d. Phương án khác:…
20. Để lấy mã lỗi của thao tác ngay trước đó. Hàm nào sau đây sẽ được sử dụng
    a. GetLastError
    b. WSAGetError
    c. WSAGetLastError
    d. WSALastError
21. Đoạn chương trình sau thực hiện kết nối đến một server, điền vào vị trí <A>, <B>, <C>các lệnh còn thiếu.

```
ret = connect(s, (sockaddr*)&serverAddr,sizeof(serverAddr));
if (ret ==SOCKET_ERROR) {
        printf("Loi ket noi: %d",<A>);
        <B>;
        <C>;
};
```

    a. <A>: GetLastError, <B>: closesocket (s), <C>: WSACleanup().
    b. <A>: WSAGetLastError(), <B> close(s), <C>: Cleanup().
    c. <A>: WSALastError(), <B>: closesocket(s), <C>: để trống.
    d. Không phương án nào đúng.

cuu duong than cong . com

22. Trong Windows Socket, với các hàm không có tiền tố WSA thì mã lỗi trả về là 0 có nghĩa là:
    a. Thành công
    b. Thất bại
    c. Tùy trường hợp
23. Lệnh nào sau đây dùng để tạo một socket TCP
    a. socket(AF_INET,SOCK_STREAM,IPPROTO_TCP).
    b. socket(AF_INET,SOCK_DGRAM,IPROTO_TCP).
    c. Socket(AF_INET,SOCK_STREAM,IPPROTO_UDP).
    d. socket(AF_INET,SOCKSTREAM,IPPROTOTCP).
24. Lệnh nào sau đây dùng để tạo một socket UDP
    a. socket(AF_INET,SOCK_UDP, IPPROTO_TCP).
    b. socket(AF_INET,SOCK_DGRAM,IPROTO_TCP).
    c. Socket(AF_INET,SOCK_STREAM,IPPROTO_UDP).
    d. Không lệnh nào đúng.
25. Cấu trúc nào được sử dụng để khai báo địa chỉ socket internet:
    a. sock_addr
    b. sockaddr
    c. sockaddr_in
    d. sock_addr_in
26. Các giá trị lớn hơn 01 byte trong cấu trúc lưu trữ địa chỉ của socket được tổ chức theo kiểu:
    a. Đầu to
    b. Đầu nhỏ
    c. Tùy trường hợp
27. Để chuyển đổi một xâu sang địa chỉ IP, lệnh nào sau đây là đúng
    a. inet_ntoa("192.168.1.1");
    b. inet_aton("192.168.1.1");
    c. inet_addr("192.168.1.1");
    d. inet_stoi("192.168.1.1");
28. Để chuyển đổi một địa chỉ IP lưu trong biến serverAddr lưu trữ địa chỉ socket sang dạng xâu ký tự, lệnh nào sau đây là đúng
    a. inet_ntos(serverAddr.sin_addr);
    b. inet_ntoa(serverAddr.s_addr);
    c. inet_itos(serverAddr.sin_addr);
    d. inet_ntoa(serverAddr.sin_addr);
29. Để chuyển đổi giá trị cổng từ đầu nhỏ sang đầu to, lệnh nào sau đây sẽ được sử dụng
    a. ntohl
    b. ntohs
    c. htonl
    d. htons
30. Để thực hiện phân giải tên miền bằng WinSock, cần bổ sung tệp tiêu đề nào
    a. Winsock2.h
    b. Ws2_32.h
    c. Ws2ip.h
    d. Ws2tcpip.h
31. Để thực hiện phân giải tên miền bằng WinSock, hàm nào sau đây có thể sử dụng
    a. getaddrinfo
    b. gethostinfo
    c. getpeerinfo
    d. Cả ba hàm trên
32. Kết quả trả về của thao tác phân giải tên miền là <A> và phải giải phóng bằng hàm/toán tử <B>
    a. <A>: Mảng, <B> delete
    b. <A>: Danh sách liên kết đơn, <B> free
    c. <A>: Danh sách liên kết kép, <B> freeaddrinfo
    d. <A>: Danh sách liên kết đơn, <B> freeaddrinfo
33. Trường nào sau đây trong cấu trúc addrinfo chứa thông tin về địa chỉ socket phân giải được.
    a. ai_addr.
    b. sock_addr.
    c. addr.
    d. in_addr.
34. Trong trường hợp nào sau đây hàm **bind** sẽ thất bại
    a. Tường lửa chặn.
    b. Đã có ứng dụng khác mở sử dụng cổng trên.
    c. Socket không hợp lệ.
    d. Cả ba trường hợp trên.
35. Lệnh nào sau đây sẽ gắn một socket s vào giao diện mạng bất kỳ được mô tả trong cấu trúc địa chỉ serverAddr.
    a. bind(s,(sockaddr_in*)&serverAddr, sizeof(serverAddr));
    b. bind(s,(sockaddr*)serverAddr,sizeof(serverAddr));
    c. bind(s,(sockaddr*)&serverAddr,sizeof(sockaddr_in));
    d. Không lệnh nào đúng.
36. Lệnh nào sau đây sẽ gán giá trị cổng 8080 cho cấu trúc địa chỉ serverAddr.
    a. serverAddr.sin_port = 8080.
    b. serverAddr.s_port = 8080.
    c. serverAddr.sin_port = htonl(8080).
    d. Lệnh khác...
37. Đoạn chương trình sau sẽ thực hiện chấp nhận kết nối từ client. Điền vào chỗ trống các phương án cần thiết

```
SOCKET   server,client;
sockaddr_in      clientAddr;
int              len;
len = <A>;
client = accept(server,<B>, <C>);
```

    a. <A> = 0, <B> = clientAddr, <C> = len.
    b. <A> = sizeof(client), <B> = &clientAddr, <C> =len.
    c. <A> = 0, <B> = (sockaddr*)&clientAddr, <C> = &len.
    d. <A> = sizeof(clientAddr), <B> = (sockaddr*)&clientAddr,&len.
38. Đoạn chương trình sau thực hiện đọc dữ liệu từ bàn phím và gửi đi qua socket s. Điền vào chỗ trống những lệnh còn thiếu.

```
char     str[1024];
int      ret = 0;
gets(str);
<A> = send(s,str,<B>,0);
if (<A><= 0)
{
        printf("Loi gui xau!");
        <C>;
}
```

    a. <A> = ret, <B> = 1024, <C> = để trống.
    b. <A> = ret, <B> = sizeof(str), <C> = close(s).
    c. <A> = str, <B> = strlen(str), <C> = close(s).
    d. <A> = ret, <B> = strlen(str), <C> = closesocket(s).
39. Trong lời gọi hàm **recv(s,buff,1024,0)**, giá trị 1024 có nghĩa là:
    a. Số byte muốn nhận
    b. Số byte tối đa muốn nhận.
    c. Kích thước bộ đệm.
    d. Cả b và c đều đúng.
40. Khi socket hoạt động ở chế độ đồng bộ, mỗi lời gọi hàm sẽ:
    a. Chặn tất cả các luồng của chương trình cho đến khi thao tác vào ra hoàn tất
    b. Chặn tất cả các luồng trừ luồng chứa lời gọi
    c. Chỉ chặn luồng chứa lời gọi, các luồng khác vẫn chạy bình thường.
    d. Không chặn luồng nào cả.
41. Khi socket hoạt động ở chế độ bất đồng bộ, mỗi lời gọi hàm sẽ
    a. Chặn tất cả các luồng của chương trình cho đến khi thao tác vào ra hoàn tất
    b. Chặn tất cả các luồng trừ luồng chứa lời gọi
    c. Chỉ chặn luồng chứa lời gọi, các luồng khác vẫn chạy bình thường.
    d. Không chặn luồng nào cả.
42. Mặc định socket khi được tạo ra hoạt động ở chế độ <A>, hàm <B> sẽ thay đổi chế độ hoạt động của socket.
    a. <A>: đồng bộ, <B>: ioctlsocket.
    b. <A>: bất đồng bộ, <B>:ioctlsocket.
    c. <A>: đồng bộ, <B>: WSAAsyncSelect.
    d. Cả a và c đều đúng

cuu duong than cong . com

43. Khi socket s hoạt động ở chế độ đồng bộ, hàm recv(s,buff,1024,0) sẽ không chặn luồng chứa lời gọi trong trường hợp nào sau đây.
    a. Có dữ liệu từ bộ đệm hệ thống nhưng không đủ 1024 byte.
    b. Có đủ 1024 byte dữ liệu từ bộ đệm hệ thống.
    c. Kết nối bị đóng.
    d. Cả ba phương án trên đều đúng.
44. Nếu cần xây dựng server đáp ứng được tối thiểu 10 kết nối, chương trình sẽ cần khai báo bao nhiêu socket ?
    a. 11
    b. 10
    c. 20
    d. 21
45. Nếu việc gửi dữ liệu cho các kết nốiđược tập trung vào trong một luồng, thì mô hình Blocking cần tối thiểu bao nhiêu luồng để đáp ứng được 100 kết nối.
    a. 100
    b. 200
    c. 201
    d. 101
46. Trong mô hình **Select**, để thăm dò sự kiện kết nối đến server thành công, client cần cho socket vào tập nào
    a. readfds
    b. writefds
    c. exceptfds
    d. Cả ba tập đều được.
47. Nếu dùng mô hình **Select** và thăm dò sự kiện cho 100 kết nối, ứng dụng sẽ cần bao nhiêu luồng ?
    a. 100
    b. 2
    c. 101
    d. 1.
48. Trong mô hình Select, socket chạy ở chế độ nào ?
    a. Đồng bộ
    b. Bất đồng bộ
    c. Không xác định
49. Hàm callback WindowProc được gọi trong ngữ cảnh của:
    a. Một luồng riêng được hệ thống tạo ra.
    b. Luồng chính xử lý giao diện.
    c. Luồng phụ do chương trình tạo ra.
    d. Phương án khác…
50. Trong các hàm xử lý sự kiện của chương trình có giao diện đồ họa, có nên gọi các hàm đồng bộ của WinSock ?
    a. Không, vì sẽ làm việc gửi nhận dữ liệu của socket bị chậm đi.
    b. Không, vì sẽ làm hệ điều hành bị treo.
    c. Có, vì không ảnh hưởng gì đến chương trình.
    d. Không, vì sẽ làm giảm khả năng đáp ứng của chương trình với các sự kiện người dùng.
51. Trong mô hình WSAEventSelect, giả sử ứng dụng có 10 SOCKET, cần mấy đối tượng WSAEVENT tương ứng?
    a. 10
    b. 20
    c. 11
    d. Số khác…
52. Đối tượng WSAEVENT được tạo ra bởi WSACreateEvent có thuộc tính:
    a. Signaled, auto reset
    b. Non-signaled, auto reset
    c. Signaled, manual reset
    d. Non-signaled , manual reset
53. Hàm WSAWaitForMultipleEvent sẽ chặn luồng có lời gọi đến khi
    a. Các đối tượng EVENT chuyển sang trạng thái báo hiệu
    b. Hết giờ
    c. Các đối tượng EVENT chuyển sang trạng thái chưa báo hiệu
    d. Cả a và b
54. Giả sử s là socket dùng để kết nối đến server khác, lệnh nào sau đây thích hợp nhất.
    a. WSAEventSelect(s, event, FD_ACCEPT | FD_CLOSE);
    b. WSAEventSelect(s, event, FD_CONNECT| FD_WRITE|FD_READ|FD_CLOSE);
    c. WSAEventSelect(s, event, FD_ACCEPT| FD_CONNECT);
    d. WSAEventSelect(s, event, FD_CONNECT| FD_CLOSE);
55. Những hàm nào sau đây có thể sử dụng mô hình vào ra Overlapped
    a. WSAConnect
    b. accept
    c. WSARecv
    d. a và c
56. Có thể sử dụng cùng một đối tượng EVENT cho hai socket khác nhau trong mô hình Overlapped được không
    a. Có
    b. Không
57. Trong mô hình vào ra Overlapped, **completion routine** sẽ được gọi bởi:
    a. Chương trình trong cùng luồng có yêu cầu vào ra.
    b. Chương trình trong một luồng khác với luồng có yêu cầu vào ra.
    c. Hệ điều hành trong luồng khác với luồng có yêu cầu vào ra.
    d. Hệ điều hành trong luồng cùng với luồng có yêu cầu vào ra.
58. Alertable là trạng thái:
    a. Đang ngủ và sẵn sàng thực hiện hàm callback từ hệ điều hành
    b. Đang thực thi và sẵn sàng thực hiện hàm callback từ hệ điều hành
    c. Đang ngủ và chưa sẵn sàng thực hiện hàm callback từ hệ điều hành
    d. Đang thực thi và chưa sẵn sàng thực hiện hàm callback từ hệ điều hành
59. Hàm nào sau đây có thể đưa luồng về trạng thái alertable
    a. SleepEx
    b. WSAConnect
    c. Sleep
    d. Cả a và c
60. CSocket là lớp
    a. Cơ sở của CAsyncSocket
    b. Dẫn xuất của CAsyncSocket
    c. Kế thừa của CAsyncSocket
    d. Không có quan hệ gì với CAsyncSocket
61. Các phương thức của CSocket đều hoạt động
    a. Đồng bộ
    b. Bất đồng bộ
62. Các phương thức của CAsyncSocket đều hoạt động
    a. Đồng bộ
    b. Bất đồng bộ
63. Luồng A tạo đối tượng m có kiểu CSocket. Trong luồng B, lệnh nào sau đây là hợp lệ
    a. m.Connect("127.0.0.1",8888);
    b. m.Close();
    c. m.ShutDown();
    d. Cả ba đều không hợp lệ.
64. Để xử lý sự kiện cho CAsyncSocket cần phải
    a. Gắn đối tượng vào một biến có kiểu WSAEVENT.
    b. Gắn đối tượng vào một cửa sổ qua hàm WSAAsyncSelect.
    c. Truyền đối số là một hàm callback cho các thao tác vào ra.
    d. Kế thừa ra một lớp mới và viết các phương thức chồng.
65. Đoạn chương trình sau sử dụng trong chương trình chat Voice xử lý việc nhận dữ liệu từ server, hãy điền vào chỗ trống những lệnh cần thiết.

```
enum    PACKET_TYPE
{
        PACKET_TYPE_TEXT,
        PACKET_TYPE_VOICE,
        PACKET_TYPE_IMAGE
};
```

cuu duong than cong . com

```
struct	Packet
{
	char	type;
	int	len;
	char	data[65536];
};
int		total = 0;
int		len = 0;
SOCKET	s;
Packet	p;
recv(s,&p.type,<A> ,MSG_WAITALL);
recv(s,(char*)&p.len,<B>,MSG_WAITALL);
switch (p.type)
{
	case	<C>:
		while (total<p.len)
		{
			len = recv(s,<D>,<E>,0);
			if (len<=0) break;
			total+=<F>;
		};
		p.data[total] = <G>;
		printf("Text:%s",p.data);
		…
}
```

Giá trị thích hợp cho <A> là

a. 1 b. 2 c. 4 d. sizeof(p)

66. Giá trị thích hợp cho <B> là
    a. 1
    b. 2
    c. 4
    d. sizeof(Packet)
67. Giá trị thích hợp cho <C> là
    a. PACKET_TYPE_VOICE
    b. PACKET_TYPE_IMAGE
    c. PACKET_TYPE_TEXT
    d. Giá trị khác…
68. Giá trị thích hợp cho <D> là
    a. &p
    b. p.data
    c. p.data+p.len
    d. Phương án khác…
69. Giá trị thích hợp cho <E> là
    a. p.len
    b. p.len – total
    c. total
    d. 65536
70. Giá trị thích hợp cho <F> là
    a. len
    b. pk.len
    c. sizeof(Packet)
    d. 4
71. Giá trị thích hợp cho <G> là
    a. 0xFF
    b. 0
    c. 1
    d. Phương án khác…
72. Đoạn chương trình sau đây thực hiện nhận dữ liệu từ một socket UDP và kiểm tra lại checksum, điền vào chỗ trống những lệnh thích hợp

```
struct Packet
{
	int		offset;
	int		len;
	unsigned short	checksum;	// XOR
	char	data[1024];
};
Packet		p;
SOCKADDR_IN	from;
int		fromLen = <A>;
int		total = 0,  len;
unsigned short	tmpchecksum = 0;
SOCKET	s;
memset(&p,0,sizeof(p));
len = recvfrom(s,(char*)&p,10,0,<E>,&fromLen);
while (<B>)
{
	len = recvfrom(s,<C>,<D>,0,<E>,&fromLen);
	if (len <=0 ) break;
	total += len;
};
for (int i=0;i<(p.len+1)/2;i++)
	tmpchecksum = tmpchecksum <F> *((unsigned short*)(&p.data[<G>]));
if (tmpchecksum!=p.checksum)
	printf("Goi tin bi loi!");
```

cuu duong than cong . com

Giá trị thích hợp cho <A> là
a. 1024
b. sizeof(p)
c. sizeof(from)
d. Phương án khác…

73. Giá trị thích hợp cho <B> là
a. true
b. 1
c. total >= p.len
d. total < p.len

74. Giá trị thích hợp cho <C> là
a. p.data + len
b. p.data
c. p.data + (p.len – total)
d. Cả ba đều sai

75. Giá trị thích hợp cho <D> là
a. 1024
b. p.len – total
c. p.len + total
d. p.len

76. Giá trị thích hợp cho <E> là
a. &from
b. sizeof(from)
c. (sockaddr*)from
d. (sockaddr*)&from

77. Giá trị thích hợp cho <F> là
a. ^
b. |
c. &
d. ~

78. Giá trị thích hợp cho <G> là
a. i*2
b. i*3
c. i*4
d. i

79. Đoạn chương trình server sau quản lý các kết nối của client dưới dạng danh sách liên kết kép. Điền vào chỗ trống các lệnh thích hợp

```
struct	Connection
{
	SOCKADDR_IN	addr;
	SOCKET	s;
	HANDLE	hThread;
	Connection * pNext;
	Connection * pPrev;
};
Connection *	pHead = 0,*pCur = 0,*pTmp = 0;
SOCKET	listen;
SOCKET	c;
SOCKADDR_IN	cAddr;
int	ret , cAddrLen = sizeof(cAddr);
while (1)
{
	<A> = accept(listen,(sockaddr*)&cAddr,<B>);
	if (!pHead)
	{
		pTmp = pCur = pHead = <C>;
		memset(pTmp,0,sizeof(Connection));
	}
	else
	{
		pTmp = <C>;
		memset(pTmp,0,sizeof(Connection));
		pCur->pNext =<D>;
		pTmp->pPrev = <E>;
		pCur = pTmp;
	};
	pTmp->s = <F>;
	pTmp->hThread = CreateThread(0,0,ReceiverThread,<G>,0,0);

};
```

Giá trị thích hợp cho <A> là
a. ret
b. c
c. Để trống
d. pHead

80. Giá trị thích hợp cho <B> là
a. sizeof(cAddr)
b. cAddrLen
c. &cAddrLen
d. &ret

81. Giá trị thích hợp cho <C> là
a. new Connection
b. new SOCKET
c. 0

cuu duong than cong . com
cuu duong than cong . com

d. Phương án khác…

82. Giá trị thích hợp cho <D> là
    a. pTmp
    b. pCur
    c. pHead
    d. pHead->pNext
83. Giá trị thích hợp cho <E> là
    a. pHead
    b. pCur
    c. pTmp
    d. pCur->pNext
84. Giá trị thích hợp cho <F> là
    a. c
    b. listen
    c. 0
    d. Phương án khác…
85. Giá trị thích hợp cho <G> là
    a. pHead
    b. pCur
    c. pTmp
    d. 0
86. Đoạn chương trình sau thực hiện phân giải tên miền nhập từ bàn phím với cổng 8888 sau đó hiện các kết quả ra màn hình, điền vào chỗ trống các lệnh thích hợp.

```
addrinfo   * info,*pCur;
char                host[1024];
SOCKADDR_IN         addr;
int                 ret;
printf("Nhap dia chi server:");
gets(host);
ret = getaddrinfo(host,<A>,0,<B>);
if (ret)
{
        printf("Khong tim thay server");
        return 0;
};
pCur = info;
while (pCur)
{
        memcpy(&addr,<C>,pCur->ai_addrlen);
        printf("%s\n",<D>);
        pCur = pCur->ai_next;
};
```

Giá trị thích hợp cho <A> là
    a. "8888"
    b. 8888
    c. htons(8888)
    d. ntohs(8888);
87. Giá trị thích hợp cho <B> là
    a. info
    b. &info
    c. 0
    d. &addr
88. Giá trị thích hợp cho <C> là
    a. info->ai_addr
    b. pCur
    c. info
    d. pCur->ai_addr
89. Giá trị thích hợp cho <D> là
    a. inet_addr(addr.sin_addr)
    b. inet_ntoa(pCur)
    c. inet_ntoa(pCur->sin_addr)
    d. Phương án khác…
90. Đoạn chương trình sau minh họa việc xử lý lệnh của HTTP server để phân tách URL mà client yêu cầu. Server sẽ dừng việc nhận dữ liệu cho đến khi gặp cặp ký tự "\r\n\r\n". Điền vào chỗ trống các lệnh thích hợp

```
SOCKET  s;
char    command[1024];
char    url[1024];
int     len = 0,ret;
char    *pos;
while (1)
{
        ret = recv(s,<A>,1,MSG_WAITALL);
        if (ret<=0) break;
        len += ret;
        if ((command[<B>-1]=='\n')&&(command[<B>-2]=='\r')&&
                (command[<B>-4]=='\n')&&(command[<B>-3]=='\r'))
                break;
};
command[len] = 0;
if (strstr(command,"GET ")==0)
        printf("Unknown command");
pos = strstr(command,"HTTP/1.0");
if (pos==0)
```

cuu duong than cong . com

```
                pos = strstr(command,"HTTP/1.1");
        if (pos<C>0)
        {
                printf("Invalid format");
                return 0;
        };
        strncpy(url,<D>,pos-command-4);
        url[<E>] = 0;
```

Giá trị thích hợp cho <A> là
a. command
b. command+len
c. command+1
d. &command

91. Giá trị thích hợp cho <B> là
a. ret
b. len
c. pos
d. len-1

92. Giá trị thích hợp cho <C> là
a. ==
b. >
c. <
d. <=

93. Giá trị thích hợp cho <D> là
a. command
b. command+4
c. command-4
d. &command+4

94. Giá trị thích hợp cho <E> là
a. pos
b. pos-command
c. pos-command+4
d. pos-command-4

cuu duong than cong . com

cuu duong than cong . com