

## Esercizio 1

Scrivere il client che interagisca con un server implementato dal seguente codice C che invia un messaggio di benvenuto e poi accetta i comandi "QUIT" per terminare la connessione e "GET NEWS" per inviare le notizie piu' recenti.

```
/* legge una linea terminate da "\n" */
int readLine(int socket, char* buffer, int size);
char * news[]={
    "crisi in borsa!!", "il real preferisce baggio a totti",
    "barrichello vuole un compagno piu' forte", NULL};
char *welcome = "HELLO NEWSOURCE\n";
int main(int argc, char** argv){
    struct sockaddr_in saddr, caddr;
    int s, client, result;

    s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
    saddr.sin_family = AF_INET;
    saddr.sin_port = htons(2000);
    saddr.sin_addr.s_addr = INADDR_ANY;
    result = bind(s, (struct sockaddr *) &saddr, sizeof(saddr));
    result = listen(s, 20);
    while(1){
        int addrlen = sizeof(struct sockaddr_in);
        client = accept(s, (struct sockaddr *) &caddr, &addrlen);
        send(client, welcome, strlen(welcome)+1, 0);
        char buffer[1024];
        while(1){
            result = readLine(client, buffer);
            if(result==-1){ break; }
            if(strncmp(buffer, "GET", 3)==0){ /* GET command */
                if(strncmp(buffer+4, "NEWS", 4)==0){ /* GET NEWS */
                    char **p_news = news;
                    for(;*p_news!=NULL;++p_news){
                        char* out=(char*)malloc(strlen(*p_news)+7);
                        strcpy(out, "NEWS ");
                        strcat(out, *p_news);
                        strcat(out, "\n");
                        send(client, out, strlen(out), 0);
                    }
                    send(client, "END\n", 5, 0);
                }
            }
            if(strncmp(buffer, "QUIT", 4)==0){ /* QUIT command */
                send(client, "BYE\n", 5, 0);
                break;
            }
        }
        close(client);
    }
    return 0;
}
```

## Esercizio 2

Dato il seguente codice HTML di un form, scrivere un servlet in grado di restituire una pagina HTML che contenga una tabella con tutte le permutazioni circolari della stringa composta dalla concatenazione di nome e cognome, ogni permutazione deve essere numerata a partire da 0 (uguale a nome + cognome).

La tabella deve presentare una permutazione per riga e le righe devono avere background ciclicamente bianco, rosso chiaro, blu chiaro.

Se viene premuto il tasto esempio, generare una pagina che spiega cosa si intende per permutazione circolare e mostri l'esempio usando nome="Uto" cognome="Ughi".

```
<html>
<head><title>Permutazioni Circolari</title></head>
<body>
<form action="MioServlet" method="GET">
Nome: <input type="text" name="nome" value="" size=20><br>
Cognome: <input type="text" name="cognome" value="" size=20>
<br>
<input type="submit" name="azione" value="permutazioni">
<input type="submit" name="azione" value="esempio">
</form>
</body>
</html>
```

Esempio

#	Permutazione
0	UtoUghi
1	toUghiU
2	oUghiUt
3	UghiUto
4	ghiUtoU
5	hiUtoUg
6	iUtoUgh

## Soluzione Esercizio 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h> /* rete internet */

#define MAX 1024
int readLine(int socket, char* buffer, int size);

int main (int argc, char **argv) {
    char rc[MAX];
    unsigned long taddr_n; /* IP addr.(net/host order) */
    struct sockaddr_in saddr; /* server address */
    int s;
    int result;

    taddr_n = inet_addr (argv[1]);
    printf ("Creating socket\n");
    s = socket (AF_INET, SOCK_STREAM, IPPROTO_TCP);
    saddr.sin_family = AF_INET;
    saddr.sin_port = htons(2000);
    saddr.sin_addr.s_addr = taddr_n;
    result=connect(s,(struct sockaddr*)&saddr, sizeof(saddr));
    result = readLine(s,rc,MAX);
    printf ("Connected to news server: %s\n",rc);
    send(s,"GET NEWS\n",10,0);
    while(1){
        result = readLine(s,rc,MAX);
        if(strncmp("END",rc,3)==0) break;
        printf("[%s]\n",rc+5);
    }
    send(s,"QUIT\n",6,0);
    close(s);
    return 0;
}
```

## Soluzione Esercizio 2

```
public class MioServlet extends HttpServlet {
    public void doGet(HttpServletRequest request,
                      HttpServletResponse response)
        throws IOException, ServletException
    {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String nome = request.getParameter("nome");
        String cognome = request.getParameter("cognome");
        String azione = request.getParameter("azione");

        out.println("<html><head>");
        out.println("<title>Permutazioni Circolari</title></head>");
        out.println("<body>");
        if(azione.equals("esempio")){
            out.println("<h1>Esempio di Permutazione Circolare</h1>");
            out.println("<p>Si ha una permutazione lineare spostando una");
            out.println("lettera dalla testa alla fine di una stringa</p>");
            out.println("<p>Ad esempio usando la string \"UtoUghi\"</p>");
            nome="Uto";
            cognome="Ughi";
        }else{
            out.println("<h1>Permutazioni Circolari</h1>");
        }
        out.println("Nome: " + nome + "<br>");
        out.println("Cognome: " + cognome + "<br>");
        out.println("<table border=1 cellpadding=3>");
        out.println("<tr><th>#<th>Permutazione");
        StringBuffer s = new StringBuffer(nome+cognome);
        for(int i=0; i<s.length(); ++i){
            String color="white";
            if(i%3==1) color = "#ff9999";
            if(i%3==2) color = "#9999ff";
            out.println("<tr bgcolor=\"\" + color + \"><td>" + i);
            out.println("<td>" + s);
            char ch = s.charAt(0);
            s.deleteCharAt(0);
            s.append(ch);
        }
        out.println("</table>");
        out.println("</body>");
        out.println("</html>");
    }
}
```