

```

1  /*
2  Traccia:
3  Lo scopo di oggi è realizzare un piccolo gioco di domanda/risposta in C, con X
4  domande casuali, e ognuna deve avere 3 risposte possibili.
5  Il gioco dovrà funzionare in modo tale da:
6  Presentare una rapida introduzione all'utente con lo scopo del programma
7  Mostrare all'utente un menu di scelta iniziale tra: A) Iniziare una nuova partita; B)
8  Uscire dal gioco
9  Ricevere in input la scelta dell'utente
10 - Creare o meno una nuova partita in base all'input utente
11 Ricevere in input nome dell'utente in caso di nuova partita
12 Presentare un set di domande all'utente a risposta multipla (almeno 3 risposte a
13 domanda)
14 Valutare la risposta utente per ogni domanda ed aggiornare una variabile «punteggio
15 in caso di risposta esatta»
16 - Scrivere a schermo a fine partita il punteggio totalizzato dal giocatore corrente
17 Presentare nuovamente il testo per la scelta tra: A) Iniziare una nuova partita; B)
18 Uscire dal gioco
19 Peppoli Andrea Paolo
20 */
21
22 #include <stdio.h>
23 #include <stdbool.h> //libreria per il tipo boolean
24 #include <unistd.h> //libreria per le pause
25
26 void messaggioSconfitta(){ //scrive 10 volte un messaggio di sconfitta
27     for (int i=0; i<10; i++){
28         printf("LOOSER\n");
29         sleep(1);
30     }
31 }
32
33 void messaggioVittoria(){ //scrive 10 volte un messaggio di vittoria
34     for (int i=0; i<10; i++){
35         printf("You are the CHAMPION!\n");
36         sleep(1);
37     }
38 }
39
40 bool risposta(int dom){
41     /*
42     Funzione che prende in input il numero della domanda,
43     e restituisce true se la risposta che abbiamo dato è vera,
44     altrimenti restituisce false.
45     */
46     int num;
47     scanf("%d", &num);
48     switch(dom){
49         case 1: return (num == 2) ? true : false;
50         case 2: return (num == 3) ? true : false;
51         case 3: return (num == 2) ? true : false;
52         case 4: return (num == 3) ? true : false;
53         case 5: return (num == 3) ? true : false;
54         case 6: return (num == 2) ? true : false;
55         case 7: return (num == 1) ? true : false;
56         case 8: return (num == 3) ? true : false;
57         case 9: return (num == 1) ? true : false;
58         case 10: return (num == 2) ? true : false;
59         case 11: return (num == 1) ? true : false;
60         case 12: return (num == 2) ? true : false;
61         case 13: return (num == 1) ? true : false;
62         case 14: return (num == 2) ? true : false;
63         case 15: return (num == 3) ? true : false;
64         default: return false;
65     }
66 }
67
68 bool domande(int numeroDomanda) {
69     /*
70     Funzione per gestire tutte le domande, prende in input il numero domanda e
71     a seconda del numero di domanda che stiamo analizzando
72     richiama la funzione risposta che abbiamo visto poco fa
73     */
74     switch(numeroDomanda) {

```

```

69     case 1:
70         printf("Domanda numero 1)\nIn una rete /28 quanti dispositivi possono
           connettersi?:\n1) 30\n2) 14\n3) 6\nScegli:\t");
71         return risposta(1);
72     case 2:
73         printf("Domanda numero 2)\nQuale dispositivo di rete permette la
           comunicazione tra client su reti diversi?\n1) Hub\n2) Switch\n3)
           Router\nScegli:\t");
74         return risposta(2);
75     case 3:
76         printf("Domanda numero 3)\nQuali dei seguenti sono tipi di
           hacker?\n1) White, red, gray\n2) Black, gray, white\n3) Gray, black, red\nScegli:\t
           ");
77         return risposta(3);
78     case 4:
79         printf("Domanda numero 4)\nQuali dei seguenti protocolli sono protocolli
           del livello di trasporto?\n1) UDP, MAC\n2) TCP, IP\n3) UDP, TCP\nScegli:\t");
80         return risposta(4);
81     case 5:
82         printf("Domanda numero 5)\nQuali tra i seguenti sono strumenti utilizzati
           dagli Ethical Hacker per raccogliere informazioni sui
           target?\n1) ss, ifconfig, Google Hacking\n2) Nslookup, ping, netstat\n3) Google
           Hacking, Nslookup, whois\nScegli:\t");
83         return risposta(5);
84     case 6:
85         printf("Domanda numero 6)\nChe tipo di strumento permette
           l'identificazione delle potenziali vulnerabilità di un sistema
           informatico?:\n1) Proxy\n2) Vulnerability scanner\n3) Firewall\nScegli:\t"
           );
86         return risposta(6);
87     case 7:
88         printf("Domanda numero 7)\nQuanti livelli o anche detti layer prevede il
           modello ISO/OSI?:\n1) 7\n2) 6\n3) 8\nScegli:\t");
89         return risposta(7);
90     case 8:
91         printf("Domanda numero 8)\nChe cos'è l'incapsulamento?:\n1) Tecnica che
           prevede l'inserimento di un pacchetto più grande all'interno di un
           pacchetto di dimensione inferiore\n2) Tecnica che permette l'invio di un
           pacchetto cifrato, in modo tale che un potenziale attaccante non ne possa
           intercettare il contenuto\n3) Tecnica mediante la quale un pacchetto
           composto da header e payload diventa il payload del pacchetto del livello
           immediatamente successivo\nScegli:\t");
92         return risposta(8);
93     case 9:
94         printf("Domanda numero 9)\nIn quali delle circostanze sotto un host invia
           una richiesta ARP al default gateway?:\n1) Per associare indirizzi IP ad
           indirizzi MAC\n2) Per comunicare agli altri host che è un nodo attivo\n3)
           Per richiedere al dhcp un indirizzo IP\nScegli:\t");
95         return risposta(9);
96     case 10:
97         printf("Domanda numero 10)\nQuali sono i componenti di un pacchetto?:\n1)
           Indirizzo IP, MAC\n2) Header, Payload\n3) Indirizzo MAC,
           header\nScegli:\t");
98         return risposta(10);
99     case 11:
100         printf("Domanda numero 11)\nQuanti livelli sono previsti nel modello
           TCP/IP?:\n1) 4\n2) 7\n3) 5\nScegli:\t");
101         return risposta(11);
102     case 12:
103         printf("Domanda numero 12)\nCome sono anche detti i pacchetti del livello
           data link?:\n1) Datagrammi\n2) Frame\n3) Bit\nScegli:\t");
104         return risposta(12);
105     case 13:
106         printf("Domanda numero 13)\nIndicare quale delle seguenti è una rete
           geografica?:\n1) WLAN\n2) VLAN\n3) Bus\nScegli:\t");
107         return risposta(13);
108     case 14:
109         printf("Domanda numero 14)\nQuale comando si utilizza da terminale
           Windows per controllare le impostazioni di rete?:\n1) ifconfig\n2)
           ipconfig\n3) igconfig\nScegli:\t");
110         return risposta(14);
111     case 15:
112         printf("Domanda numero 15)\nQuale delle seguenti caratteristiche è vera

```

```

        per i router?:\n1) Servono solo per scambio di pacchetti su stessa
        rete\n2) Includono controlli per la sicurezza di rete\n3) Separano i
        domini di broadcast\nScegli:\t");
113     return risposta(15);
114     default:
115         return false;
116     }
117 }
118
119 void fineGioco(int punteggio, char utente[]) {
120     /*
121     Funzione implementata per la fine del gioco
122     prende in input il punteggio e il nome dell'utente che sta giocando
123     restituisce il messaggio a seconda del punteggio fatto
124     */
125     if(punteggio == -1){
126         printf("Hai deciso di uscire dal gioco!\n");
127         return 0;
128     }
129     else{
130         printf("Hai ottenuto un punteggio di %d punti\n", punteggio);
131         sleep(1);
132         if(punteggio == 0){
133             printf("%s hai ottenuto il punteggio minimo, sei una schiappa!\n", utente
134             );
135             messaggioSconfitta();
136         }
137         else if(punteggio < 15){
138             printf("%s hai ottenuto un punteggio che è migliorabile, riprova!\n",
139             utente);
140         }
141         else{
142             printf("Complimenti %s .Hai ottenuto il punteggio massimo!\nSei veramente
143             un campione!\n", utente);
144             sleep(1);
145             messaggioVittoria();
146         }
147         inizioGioco();
148     }
149 }
150
151 void nuovaPartita() {
152     /*
153     Logica del gioco, tramite un ciclo for scorriamo
154     il numero della domanda che stiamo facendo in quel momento
155     e aggiorniamo il punteggio
156     */
157     printf("Bene, stai per cominciare una nuova partita.\nPer prima cosa inserisci il
158     tuo nickname (max 20 caratteri): \t");
159     char utente[20];
160     scanf("%s", &utente);
161     printf("*****\n");
162     int numDomande = 15;
163     int punteggio = 0;
164
165     for(int i = 1; i <= numDomande; i++) {
166         if (domande(i)) {
167             punteggio++;
168         }
169     }
170     fineGioco(punteggio, utente);
171 }
172
173 void inizioGioco() {
174     /*
175     Fase iniziale del gioco, richiede in input una lettera
176     all'utente, se sceglie A inizia il gioco,
177     se sceglie B il gioco termina,
178     se sceglie qualsiasi altra lettera ripropone la domanda
179     finché non avrà una risposta corretta.
180     */
181     printf("A) Iniziare una nuova partita;\nB) Uscire dal gioco.\nScegli:\t");
182     char scelta;

```

```

179     scanf(" %c", &scelta);
180     printf("*****\n");
181     while(1) {
182         if(scelta == 'A' || scelta == 'a') {
183             nuovaPartita();
184             break;
185         } else if(scelta == 'B' || scelta == 'b') {
186             fineGioco(-1, "");
187             break;
188         } else {
189             printf("Inserisci un carattere valido (A/a o B/b)\nScegli:\t");
190             scanf(" %c", &scelta);
191             printf("*****\n");
192         }
193     }
194 }
195
196 int main() {
197     /*
198     Main, inizio del programma, fornisce delle indicazioni
199     sullo svolgimento del gioco, e richiama subito la funzione
200     inizioGioco();
201     */
202     printf("Benvenuto nel mio gioco testuale, segui le indicazioni a schermo per
    giocare!\nNel caso in cui indovinerai le risposte di tutte le domande, vincerai,
    altrimenti dovrai riprovare!\n");
203     inizioGioco();
204     return 0;
205 }

```