

ESERCITAZIONE W6D2

In questa esercitazione ci viene chiesto di creare un gioco, usando il linguaggio C, nel quale un utente deve rispondere a delle domande con lo scopo di aumentare il proprio punteggio.

Prima di tutto andiamo a creare una introduzione, attraverso “char *intro” per poi renderla visibile all’utente tramite “printf”. Successivamente diamo la possibilità all’utente di scegliere se iniziare una nuova partita oppure uscire dal gioco.

```
15 int main() {
16     // Introduzione all'utente con lo scopo del programma
17     char *intro = "Benevenuto nel gioco di quiz!\n"
18                 "In questo gioco dovrai rispondere a delle domande e ottenere un punteggio.\n"
19                 "Il tuo obiettivo è rispondere correttamente a più domande possibile.\n"
20                 "Buona fortuna!";
21     printf("%s\n\n", intro);
```

Con il comando “while (true)” nel modo in cui l’abbiamo utilizzato non fa altro che creare un ciclo infinito del gioco che è possibile interrompere scegliendo di uscire dal gioco. Infatti serve, una volta terminate le domande, a ripetere il menù principale in modo da giocare più partite.

```
29
30     while(true) {
31         // Mostrare all'utente un menu di scelta iniziale tra: A) Iniziare una nuova partita; B) Uscire dal gioco
32         printf("%s\n", prompt_scelta);
33
34
35         // Ricevere la scelta dell'utente
36         printf("\nAllora?: ");
37         scanf(" %c", &scelta);
38     }
```

Con il comando “switch(tolower(scelta))” andiamo a dare la possibilità di eseguire un blocco del codice a seconda del valore inserito. Con tolower invece andiamo a dare la possibilità di leggere sia le maiuscole che le minuscole quando andiamo a scegliere tra “A) inizia partita” e “B) esci dal gioco”. Abbiamo inoltre il “default” che ci permette di mostrare all’utente di aver inserito un input non valido. Questo è un modo per evitare di usare tanti “if” rendendo il codice così un po' più ordinato.

```
switch(tolower(scelta)) {
    case 'a':
        if(strcmp(nome, DEFAULT_NAME) == 0) {
            printf("\nCome ti chiami?: ");
            scanf("%s", nome);
            printf("\n");
        }
        score += partita();
        break;
    case 'b':
        printf("Complimenti %s, il tuo risultato è: %d\n", nome, score);
        return 0;
    default:
        printf("Riesci a dare una risposta valida\n");
        break;
}
```

Con il comando “bool contains” come scritto all’interno della figura sottostante serve a controllare se l’input immesso dall’utente è tra le opzioni disponibili, riceve un array con le opzioni valide e l’elemento da controllare. Se quest’ultimo è presente restituisce “true”, altrimenti “false”. Questo porta praticamente ad

esempio che se l'utente digita 45 il programma lo riconosce come input non valido chiedendo di nuovo.

```
91
92 // controlla che l'elemento sia contenuto nell'array
93 // arr = [1, 2 ,3] | element = 3
94 bool contains(int *arr, int element) {
95     for(int i=0; i<sizeof(arr); i++) {
96         if(arr[i] == element) {
97             return true;
98         }
99     }
100     return false;
101 }
```

La logica della domanda invece è molto più semplice. Attraverso “int domanda(char *d, int correct” andiamo a mostrare le domande ricevendo le risposte. Se il giocatore risponde correttamente restituisce 1 punto altrimenti 0, e come in precedenza se da una risposta non valida continua a chiedere.

```
103
104 // Se la risposta è corretta, ritorna +1 al punteggio, altrimenti 0
105 // Se la risposta non è tra quelle possibili, richiede nuovamente l'input
106 int domanda(char *d, int correct) {
107     int scelta;
108     int possible_answers[] = {1, 2, 3};
109
110     while(true) {
111         printf("%s\nRisposta: ", d);
112         scanf("%d", &scelta);
113
114         if(scelta == correct) {
115             return 1;
116         } else if(contains(possible_answers, scelta)) {
117             return 0;
118         } else {
119             printf("Scegli un'opzione valida\n");
120         }
121     }
122 }
```

Alla fine quando l'utente sceglie di uscire dal gioco restituisce un messaggio finale con il suo punteggio accumulato fino a quel momento.

Come vediamo dalla figura sottostante dimostriamo come è il gioco una volta completato il tutto. Possiamo vedere il benvenuto al giocatore, la possibilità di scegliere se iniziare o meno una nuova partita, l'attesa della risposta, la scelta del nome da parte del giocatore. Dopo le 5 domande previste dallo gioco ripete la domanda se giocare di nuovo e il risultato finale quando scegliamo di non proseguire nel gioco.

```

The Actions Edit View Help
Game
Benevenuto nel gioco di quiz!
In questo gioco dovrai rispondere a delle domande e ottenere un punteggio.
Il tuo obiettivo è rispondere correttamente a più domande possibile.
Buona fortuna!
score += domanda;

Vuoi:
A) Iniziare una nuova partita
B) Uscire dal gioco
return false;

Allora?: a
score += domanda;

Come ti chiami?: FRANCESCO
return true;

Quale di questi è il protagonista di una favola?
1) Pinocchio
2) Hitler
3) Tu
score += domanda;

Quale di questi è un colore?
1) Blu
2) Limone
3) Sedia
score += domanda;

Quale di questi è un animale?
1) Gatto
2) Tavolo
3) Computer
score += domanda;

Quale animale è da compagnia?
1) Giraffa
2) Cane
3) Elefante
return true;

Risposta: 2
Vuoi:
A) Iniziare una nuova partita
B) Uscire dal gioco
return false;

Allora?: B
Complimenti FRANCESCO, il tuo risultato è: 5

```