

INDICE

Indice.....	1
Introduzione ed obiettivo.....	2
Creazione del menu principale e introduzione al gioco.....	3-4
Funzione di gioco e gestione dell'input utente.....	5
Domande ed acquisizione risposte.....	6-7
Calcolo punteggio e funzione principale main().....	8-9
Esecuzione e verifica dell'output.....	10-11
Conclusione finale.....	12
Glossario.....	13

INTRODUZIONE ED OBIETTIVO

INTRODUZIONE:

L'esercizio proposto consisteva nella realizzazione di un piccolo gioco interattivo in linguaggio C, eseguibile da terminale.

L'obiettivo era applicare in modo pratico i concetti fondamentali della programmazione ossia **input** e **output** con le **funzioni printf() e scanf()** ; l'uso di **variabili** di tipo **char** e **int**, le **strutture condizionali if/else** ed infine l'utilizzo del **ciclo do...while** per creare un menu che si ripete finché l'utente non decide di uscire.

OBIETTIVO:

Creare un programma che mostri un messaggio introduttivo, presenti un menu principale con due opzioni :

- A)** avvia una nuova partita,
- B)** esci dal gioco .

Un altro obiettivo era che il programma chiedesse il nome del giocatore e lo facesse rispondere a tre domande a scelta multipla.

Infine v'era da valutare le risposte, mostrare per ciascuna se è corretta o sbagliata e calcolare il punteggio totale e per concludere restituire all'utente il risultato finale prima di tornare al menù.

CREAZIONE DEL MENÙ PRINCIPALE ED INTRODUZIONE AL GIOCO

```
#include <stdio.h> /* I/O standard */
#include <ctype.h> /* toupper */
#include <string.h> /* gestione stringhe */

/* INTRODUZIONE*/
void introduzione(void) {

    printf("=====\\n"); // \\n serve per andare a capo, come "Invio" sulla tastiera
    printf("  Forza e Coraggio – Tirati su le maniche!  \\n");
    printf("=====\\n\\n"); // il doppio \\n manda due righe vuote per separare visivamente
}

/* MENÙ PRINCIPALE*/
char menu_principale(void) {
    char selezione; // char = tipo di variabile per un solo carattere (es. 'A')

    printf("Scegli una delle due opzioni:\\n"); // mostra il testo
    printf("  A) Inizia una nuova partita\\n"); // stampa un'opzione
    printf("  B) Esci dal gioco\\n"); // stampa l'altra
    printf("Selezione: "); // prompt per l'input
    scanf(" %c", &selezione); // scanf serve per leggere un input da tastiera e "%c" = leggi un singolo carattere
    selezione = (char)toupper((unsigned char)selezione); // toupper() trasforma il carattere in maiuscolo

    return selezione;
}
```

Nella prima parte del programma vengono incluse tre **librerie** fondamentali: **stdio.h**, **ctype.h** e **string.h**.

La libreria **stdio.h** (Standard Input Output) consente di utilizzare le funzioni di input e output come **printf()** e **scanf()**, indispensabili per scrivere e leggere dati dal terminale.

La libreria **ctype.h** serve per la gestione dei caratteri e viene utilizzata in particolare per la funzione **toupper()**, che converte una lettera minuscola in maiuscola.

Infine, la libreria **string.h** permette di lavorare con le stringhe di testo, anche se in questa parte del codice è inclusa principalmente per compatibilità con le funzioni successive del programma.

Dopo le librerie, viene definita la **funzione introduzione()**, che visualizza a schermo il messaggio iniziale del gioco tramite la funzione **printf()**. Le sequenze di **escape** **\\n** vengono utilizzate per andare a capo e creare una formattazione più ordinata e leggibile all'interno del terminale.

Segue la **funzione menu_principale()**, che rappresenta il primo momento interattivo del programma.

All'interno di questa funzione viene dichiarata una **variabile** di tipo **char**, utilizzata per memorizzare la scelta dell'utente tra le due opzioni proposte:

- A)** per iniziare una nuova partita oppure
- B)** per uscire dal gioco.

Il comando **scanf**(" %c", &selezione); permette di leggere da tastiera il carattere digitato, e grazie alla **funzione toupper()** l'input viene automaticamente convertito in maiuscolo, garantendo che il programma riconosca correttamente la selezione anche se l'utente scrive in minuscolo. Questa fase costituisce quindi la base logica per la gestione delle scelte dell'utente nel menù principale.

FUNZIONE DI GIOCO E GESTIONE INPUT UTENTE

```
#include <stdio.h> /* I/O standard */
#include <ctype.h> /* toupper */
#include <string.h> /* gestione stringhe */

/* INTRODUZIONE */
void introduzione(void) {

    printf("=====\n"); // \n serve per andare a capo, come "Invio" sulla tastiera
    printf("  Forza e Coraggio – Tirati su le maniche!  \n");
    printf("=====\n\n"); // il doppio \n manda due righe vuote per separare visivamente
}

/* MENU PRINCIPALE */
char menu_principale(void) {
    char selezione; // char = tipo di variabile per un solo carattere (es. 'A')

    printf("Scegli una delle due opzioni:\n"); // mostra il testo
    printf("  A) Inizia una nuova partita\n"); // stampa un'opzione
    printf("  B) Esci dal gioco\n"); // stampa l'altra
    printf("Selezione: "); // prompt per l'input
    scanf(" %c", &selezione); // scanf serve per leggere un input da tastiera e " %c" = leggi un singolo carattere
    selezione = (char)toupper((unsigned char)selezione); // toupper() trasforma il carattere in maiuscolo

    return selezione;
}

/* FUNZIONE GIOCO */
int gioco(void) {
    char nome_giocatore[30]; // array di char per memorizzare una parola (stringa)
    int punteggio = 0; // contatore del punteggio (numero intero)

    printf("\nInserisci il tuo nome: ");
    scanf("%29s", nome_giocatore); // "%s" legge una stringa senza spazi, max 29 caratteri perché l'ultimo carattere serve al terminatore '\0' che indica fine stringa
    printf("Ciao %s!\n", nome_giocatore); // %s serve per stampare una stringa
}
```

La funzione **gioco()** rappresenta il cuore del programma e racchiude tutte le istruzioni necessarie per avviare una nuova partita.

All'interno vengono dichiarate due variabili principali: **nome_giocatore**, di tipo *array di caratteri*, utilizzata per memorizzare il nome inserito dall'utente, **punteggio**, di tipo intero e funge da contatore per registrare il numero di risposte corrette.

L'array **nome_giocatore[30]** consente di salvare una stringa composta da un massimo di ventinove caratteri, poiché l'ultimo è riservato al terminatore `\0` che indica la fine della stringa.

L'istruzione **printf("\nInserisci il tuo nome: ");** mostra a schermo la richiesta di input, mentre **scanf("%29s", nome_giocatore);** permette di leggere la stringa digitata da tastiera.

Il formato `%29s` limita la lunghezza dell'input, evitando eventuali errori dovuti all'inserimento di un nome troppo lungo.

Successivamente, il programma saluta il giocatore attraverso la funzione **printf("Ciao %s!\n", nome_giocatore);**, dove il segnaposto `%s` viene sostituito dal contenuto della variabile **nome_giocatore**.

Questa prima parte della funzione di gioco ha quindi lo scopo di personalizzare l'esperienza del giocatore e di inizializzare le variabili necessarie per la gestione delle domande e del punteggio, preparando il terreno per le sezioni successive dedicate alla logica delle domande e alla valutazione delle risposte.

DOMANDE ED ACQUISIZIONE RISPOSTE

```
/* PRIMA DOMANDA*/
char risposta; // dichiarata una volta, può essere riutilizzata

printf("\nDomanda 1: Chi ha vinto lo scudetto quest'anno?\n");
printf("A) MILAN\nB) CATANZARO\nC) NAPOLI\n");
printf("Risposta: ");
scanf(" %c", &risposta); // leggo il carattere scelto

if (toupper(risposta) == 'C') { // confronto con 'C'
    printf("Risposta esatta!\n");
    punteggio++; // punteggio = punteggio + 1
} else {
    printf("Sbagliato :(\n");
}

/* SECONDA DOMANDA*/
printf("\nDomanda 2: Quante dita ha un umano (mani + piedi)?\n");
printf("A) 13\nB) 36\nC) 40\n");
printf("Risposta: ");
scanf(" %c", &risposta);

if (toupper(risposta) == 'B') {
    printf("Risposta esatta!\n");
    punteggio++;
} else {
    printf("Sbagliato :(\n");
}

/* TERZA DOMANDA */
printf("\nDomanda 3: Chi ha scritto la Divina Commedia?\n");
printf("A) Socrate\nB) Dante\nC) Ariosto\n");
printf("Risposta: ");
scanf(" %c", &risposta);

if (toupper(risposta) == 'B') {
    printf("Risposta esatta!\n");
    punteggio++;
} else {
    printf("Sbagliato :(\n");
}
```

In questa parte della **funzione gioco()** viene gestito il ciclo domanda-risposta per tre quesiti a scelta multipla.

Per prima cosa si dichiara una sola **variabile char risposta**;, riutilizzata per tutte le domande così da non occupare memoria inutile.

Ogni domanda viene mostrata con una serie di **printf()** che stampano il testo del quesito e le tre opzioni (A/B/C), seguite da un **prompt "Risposta: "**.

L'input dell'utente si acquisisce con `scanf(" %c", &risposta);`: il formato `%c` legge **un singolo carattere**, mentre lo **spazio prima di %c serve** a consumare eventuali `\n` rimasti nel buffer (per evitare letture "vuote" dopo l'Invio).

Subito dopo, un **blocco if/else** valuta l'esito usando `toupper(risposta)` per rendere l'input *case-insensitive* (sia 'c' sia 'C' sono considerati uguali) e confronta con la lettera corretta del quesito (es. 'C' per lo scudetto, 'B' per il numero di dita, 'B' per l'autore della Divina Commedia).

Se la risposta è giusta, viene stampato **"Risposta esatta!"** e il contatore punteggio viene incrementato con `punteggio++;`; in caso contrario viene mostrato **"Sbagliato :("**.

La struttura è ripetuta identica per le tre domande, garantendo coerenza di comportamento e una logica chiara: stampa - acquisisci - normalizza - confronta - aggiorna il punteggio - fornisci feedback.

CALCOLO PUNTEGGIO E FUNZIONE PRINCIPALE MAIN()

```
/* PUNTEGGIO FINALE*/
printf("\nHai totalizzato %d punti!\n\n", punteggio); // %d serve per stampare un numero intero
return punteggio; // restituisco il punteggio al main
}

/* FUNZIONE PRINCIPALE MAIN*/
int main(void) {
    char selezione; // memorizza la scelta del menu

    introduzione(); // richiamo la funzione introduttiva
    do {
        selezione = menu_principale(); // chiamo la funzione menu e salvo la risposta
        switch (selezione) { //switch controlla più casi (A, B, o altro)
            case 'A': // se l'utente ha scelto 'A'
                gioco(); // esegue il quiz
                break; // esce dallo switch

            case 'B': // se l'utente ha scelto 'B'
                printf("\nArrivederci!\n");
                break;

            default: // qualsiasi altra lettera
                printf("Scelta non valida. Riprova!\n\n");
                break;
        }
    } while (selezione != 'B'); // il ciclo continua finché non premi 'B'

    return 0; // termina il programma
}
```

Nella parte finale del programma viene gestita la **chiusura della funzione gioco()** e l'intera **logica di controllo** del menu principale all'interno della **funzione main()**.

Dopo aver posto le tre domande, la riga `printf("\nHai totalizzato %d punti!\n\n", punteggio);` mostra sullo schermo il punteggio ottenuto nella singola partita. **L'identificatore %d** serve per stampare valori di tipo intero, in questo caso la variabile `punteggio`.

Successivamente **l'istruzione return punteggio;** restituisce questo valore al programma principale, permettendo di usarlo in altre parti del codice (ad esempio per calcolare punteggi cumulativi).

La **funzione main()** rappresenta il punto di ingresso del programma: qui avviene la gestione del flusso logico generale.

La **variabile char** `selezione` viene dichiarata per memorizzare la scelta effettuata dall'utente nel menu ed in seguito viene richiamata la **funzione introduzione()** per mostrare il messaggio iniziale.

In questa parte del programma si entra in un **ciclo do while**, che consente di ripetere la visualizzazione del menu fino a quando l'utente non decide di uscire.

All'interno del ciclo, l'istruzione **selezione = menu_principale();** richiama la funzione del menu e salva la risposta.

Il **costrutto switch (selezione)** permette di gestire in modo ordinato i possibili casi: se la scelta è **'A'** viene avviata una nuova partita tramite la funzione `gioco()`; se la scelta è **'B'**, viene mostrato il messaggio di uscita "Arrivederci!" e il ciclo termina.

In tutti gli altri casi viene stampato un messaggio di errore invitando l'utente a riprovare; la condizione **while** (selezione `!= 'B'`); garantisce che il programma continui a riproporre il menu fino a quando non viene digitata la lettera **B**, rendendo così il flusso interattivo e continuo fino alla chiusura del gioco.

ESECUZIONE E VERIFICA DELL'OUTPUT

```
(kali㉿kali)-[~/Desktop/EPICODE/M1W6D1]
$ ./es1

=====
Forza e Coraggio – Tirati su le maniche!
=====

Scegli una delle due opzioni:
  A) Inizia una nuova partita
  B) Esci dal gioco
Selezione: A

Inserisci il tuo nome: VIKI
Ciao VIKI!

Domanda 1: Chi ha vinto lo scudetto quest'anno?
A) MILAN
B) CATANZARO
C) NAPOLI
Risposta: C
Risposta esatta!

Domanda 2: Quante dita ha un umano (mani + piedi)?
A) 13
B) 20
C) 40
Risposta: B
Risposta esatta!

Domanda 3: Chi ha scritto la Divina Commedia?
A) Socrate
B) Dante
C) Ariosto
Risposta: B
Risposta esatta!

Hai totalizzato 3 punti!

Scegli una delle due opzioni:
  A) Inizia una nuova partita
  B) Esci dal gioco
Selezione: B

Arrivederci!
```

All'avvio il programma mostra l'introduzione con le tre righe di intestazione e, subito dopo, il menù principale con le opzioni

A) Inizia una nuova partita ,

B) Esci dal gioco.

Selezionando **A** viene richiesto il nome dell'utente ed ha inizio il quiz composto da tre domande a scelta multipla.

Per ciascun quesito il programma stampa il testo della domanda con le opzioni *A/B/C*, attende l'input di un singolo carattere, normalizza la lettera in maiuscolo e fornisce un feedback immediato ("**Risposta esatta!**" oppure "**Sbagliato :(**").

Al termine delle tre domande calcola e visualizza il risultato con la riga "**Hai totalizzato 3 punti!**".

Conclusa la partita, il flusso ritorna automaticamente al menù iniziale: scegliendo **B** viene stampato "**Arrivederci!**" e l'esecuzione termina.

Lo screenshot allegato mostra l'intera sequenza: scelta del menù, inserimento del nome, tre risposte corrette e ritorno al menù con uscita finale.

CONCLUSIONE FINALE

L'esercizio ha permesso di consolidare le basi della programmazione in linguaggio C attraverso la realizzazione di un piccolo gioco a domande e risposte completamente funzionante.

Il progetto ha unito la logica procedurale alla gestione dell'input da tastiera, all'uso delle strutture condizionali if/else, al costrutto switch, e ai cicli do while permettendo di creare un flusso interattivo chiaro e controllato.

La suddivisione del codice in funzioni (introduzione(), menu_principale(), gioco()) ha reso il programma più leggibile e modulare, facilitandone la manutenzione e l'estensione.

L'utilizzo delle librerie standard (stdio.h, ctype.h, string.h) ha inoltre mostrato come combinare strumenti fondamentali per la gestione dell'input/output, dei caratteri e delle stringhe.

Nel complesso, l'attività ha rappresentato un esercizio completo e formativo, utile per comprendere le basi della struttura di un programma in C e per acquisire maggiore familiarità con la logica di controllo, la gestione delle variabili e l'interazione con l'utente.

GLOSSARIO

#include – Direttiva del preprocessore che serve per importare librerie esterne, le quali contengono funzioni già pronte all'uso.

stdio.h – Libreria standard di input/output che permette di utilizzare funzioni come `printf()` e `scanf()`.

ctype.h – Libreria utilizzata per la gestione dei caratteri; include funzioni come `toupper()` per convertire lettere minuscole in maiuscole.

string.h – Libreria dedicata alla manipolazione delle stringhe (insiemi di caratteri terminati dal simbolo `\0`).

void – Parola chiave che indica che una funzione non restituisce alcun valore.

char – Tipo di dato che memorizza un singolo carattere (es. 'A', 'b', '1').

int – Tipo di dato numerico intero, usato per contatori o valori numerici non decimali.

printf() – Funzione che serve per stampare testo o valori a schermo.

scanf() – Funzione che consente di leggere dati inseriti dall'utente da tastiera.

%c, %s, %d – Specificatori di formato usati da `printf()` e `scanf()` per indicare il tipo di dato (carattere, stringa, intero).

\n – Sequenza di escape che serve per andare a capo nel terminale.

if / else – Struttura condizionale che permette di eseguire blocchi di codice diversi a seconda che una condizione sia vera o falsa.

switch / case – Struttura di selezione multipla che consente di gestire più scelte possibili in modo ordinato.

do...while – Struttura ciclica che esegue un blocco di codice almeno una volta e poi lo ripete finché la condizione specificata è vera.

return – Istruzione che restituisce un valore alla funzione chiamante o termina l'esecuzione di una funzione.

punteggio++ – Operatore di incremento che aumenta di 1 il valore della variabile `punteggio`.

toupper() – Funzione che converte un carattere minuscolo in maiuscolo, utile per rendere l'input dell'utente non sensibile al caso.

& – Operatore "address of": serve per passare l'indirizzo di memoria di una variabile a una funzione come `scanf()`.

== – Operatore di confronto utilizzato per verificare l'uguaglianza tra due valori.