

ESERCIZIO W6D1: OTTIMIZZAZIONE

Prendendo come riferimento l'esercizio precedente, ci soffermiamo sull'ottimizzare il codice ottenuto. La traccia da seguire è la seguente:

Ieri abbiamo visto come scrivere un piccolo gioco di domande e risposte in C. Oggi pensiamo all'ottimizzazione del codice, ed alla gestione delle situazioni non previste. Riprendete il codice del programma che avete scritto ieri e facciamo le seguenti considerazioni:

- Cosa succede se l'utente inserisce una lettera diversa da A o B in fase di scelta iniziale? Il programma termina, ma non è una casistica che abbiamo gestito.
- Cosa succede se l'utente inserisce un nome che ha più caratteri della dimensione dell'array «nome» che abbiamo dichiarato inizialmente nella fase di avvio nuova partita? Riceveremo un errore (provate ad inserire una sequenza molto lunga di caratteri)
- Cosa succede se l'utente inserisce la lettera D per la risposta alle domande durante una partita? O un carattere numerico? Tutte queste situazioni vanno considerate in fase di programmazione in quanto errori logici o errori di mancata gestione di situazioni non standard potrebbero portare a bug nel codice che potrebbero essere sfruttati da un attaccante per prendere controllo dell'esecuzione del programma ed eseguire codice malevolo.

Casistica scelta differente tra A e B

Per evitare di incappare errori nel codice, basterà seguire un certo ordine. Con if abbiamo la prima condizione che punta allo start del gioco, con else if la seconda condizione, che punta ad uscire dal gioco, mentre la

terza possibilità ci darà un ultimo else, con dentro una stampa a video del testo “scelta non valida, riprova”.

```
if (scelta == 1) {
    printf("\nInserisci il tuo nome: ");
    fgets(nome, sizeof(nome), stdin);
    nome[strcspn(nome, "\n")] = 0;
}

} else if (scelta == 2) {
    printf("\nGrazie per aver giocato! Alla prossima!\n");
    return 0;
} else {
    printf("\nScelta non valida. Riprova.\n");
}
```

```
Benvenuto al gioco di domanda/risposta in C!  
Lo scopo del gioco è rispondere correttamente alle domande a risposta multipla.  
  
Scegli tra le seguenti opzioni:  
1. Inizia una nuova partita  
2. Esci dal gioco  
3.  
  
Inserisci la domanda di quale è il fiume più lungo d'Italia?  
Scelta non valida. Riprova.
```

Casistica stringa lunga

Per evitare che un nome molto lungo vada a creare errori capaci di assecondare un buffer overflow, applichiamo delle modifiche al codice.

```
void flushinputbuffer()
{
    int c;
    while ((c = fgetc(stdin)) != '\n' && c != EOF); // Svuota stdin
}

File System

int main() {
    int scelta, punteggio = 0;
    char nome[20];

    printf("Benvenuto al gioco di domanda/risposta in C!\n");
    printf("Lo scopo del gioco è rispondere correttamente alle domande a risposta multipla.\n");

    do {
        printf("\nScegli tra le seguenti opzioni:\n");
        printf("1. Inizia una nuova partita\n");
        printf("2. Esci dal gioco\n");
        scanf("%d", &scelta);
        flushinputbuffer();
        if (scelta == 1) {
            printf("\nInserisci il tuo nome: ");
            fgets(nome, sizeof(nome), stdin);
            nome[strcspn(nome, "\n")] = '\0';
            flushinputbuffer();
            printf("\nCiao %s, buona fortuna!\n", nome);
        }
    } while (scelta != 2);

    printf("Punteggio finale: %d\n", punteggio);
    return 0;
}
```

Precauzioni
contro lo
sforamento...

- 1) precauzione: Scriviamo una funzione void che ci permette di svuotare il buffer di input stdin. Legge i caratteri per mezzo del while fino a che non incontra un carattere di nuova riga \n oppure la fine del file EOF.
- 2) Precauzione: Al posto di scanf utilizziamo fgets() per leggere una riga di input dal buffer di input stdin, secondo la dimensione specificata da sizeof(nome), ovvero 20.
- 3) Precauzione: la linea di codice con strchr() andrà a rimuovere il carattere di nuova riga. Impostando l'ultimo carattere a \0, stiamo terminando correttamente la stringa

Casistica lettera sbagliata

Sotto indico il metodo usato per evitare di incorrere in errori durante la risposta ad una delle domande del quiz. Forse è un metodo un po' pigro, però possiamo semplicemente lasciare l'else, dopo la condizione if avente risposta corretta, che si prende carico di qualsiasi input immesso da tastiera.

```
printf("\nDomanda 1: Quale è la capitale dell'Italia?\n");
printf("1. Roma\n");
printf("2. Milano\n");
printf("3. Napoli\n");
int risposta1;
scanf("%d", &risposta1);
if (risposta1 == 1) {
    printf("Risposta corretta!\n");
    punteggio++;
} else {
    printf("Risposta sbagliata!\n");
}
```

Else risposta
sbagliata