

Progetto finale per il corso di Laboratorio 2A - a.a 2023-24 - appello estivo Frequently Asked Questions

Q1: Visto che consideriamo la coppia Qu come parola unica mi era venuto in mente di considerarla come carattere speciale, però mi è venuto un dubbio nella matrice di caratteri possono comparire singolarmente Q ed U?

A1: il suggerimento è in effetti di trattarla come carattere speciale, mantenendo caratteri come singoli elementi della matrice. U potrebbe comparire senza la Q ma la Q non può comparire senza U (ovvero la lettera Q appare solo nella forma Qu).

Q2: Quando viene detto che il server deve poter generare le matrici casualmente si intende che ogni carattere può essere generato casualmente, ossia potrebbe succedere di avere una matrice di sole lettere uguali, ad esempio tutte a, o con molte lettere uguali? ed in caso bisogna controllare che la matrice contenga almeno una parola componibile della matrice sia effettivamente di senso compiuto mediante ricerca nel dizionario?

A2: Non è necessario controllare alcunché.
Ogni carattere generato casualmente, se ne esce una matrice che non ammette parole... pazienza.

Q3: Nella registrazione dell'utente è possibile inserire caratteri speciali? inoltre bisogna distinguere i nomi utente con lettere in upper e lower case? ad esempio TaleWind e Talewind. Inoltre la matrice ha bisogno di essere in upper case?

A3: Suggerisco di limitarsi alle lettere e alle cifre. In altre parole
abcdefghijklmnopqrstuvwxyz0123456789
La scelta se distinguere tra maiuscole o minuscole la lasciamo a voi.
Qualunque sia la scelta, riportatela nella relazione.

Q4: Nel testo non è specificato il numero massimo di client che possono connettersi al server, è effettivamente così o c'è un numero di client massimo che dobbiamo predisporre noi?

A4: Giusta osservazione, che ha anche implicazioni sulla scelta riguardante le strutture dati da utilizzare. Suggerisco di definire una costante (es. MAX_NUM_CLIENTS) e di utilizzare quella come riferimento. Si scelga un valore ragionevole, ad esempio 32.

Q5: Nel protocollo di comunicazione è scritto che i messaggi tra client e server devono avere una determinata struttura, mi chiedevo se dovessero essere mandati con quella struttura oppure si potessero fare più letture per singolo messaggio. Mi spiego meglio, quando il client richiede una matrice il messaggio sarebbe del tipo MSG_MATRICE,0,/*campo vuoto*/ che andrebbe tokenizzato per ottenere i singoli campi del messaggio, oppure potrei mandarlo con, in questo caso 2 write perchè len = 0?

A5: Ti rispondo con un suggerimento. Io indicherei la struttura del messaggio in questo modo: int size, char type, char* payload, proprio in questo ordine. Il che ti garantisce che il primo elemento è un intero, la dimensione del payload. In altri termini, io scriverei la dimensione, poi il tipo e, nel caso, il payload. In questo modo sai sempre quanto scrivere e quanto leggere.

Q6: Esistono funzioni di libreria per monitorare il passaggio del tempo, oppure dobbiamo implementarle noi?

A6: Ricordi quando abbiamo parlato della alarm ?

Q7: Esiste un modo "carino" per controllare i parametri opzionali passati da riga di comando invece di controllare parametro per parametro di cosa si tratta con un if?

A7: Io darei un occhio alla funzione getopt

Q8: E' possibile aggiungere funzioni di librerie NON POSIX o cmq non affrontate nel corso?

A8: Potete utilizzarle se:

1- è software open source (cioè rilasciato con licenza GPL, LGPL, creative commons, o simili)

- 2- inserite tutti i file per compilare e includere la libreria nel progetto e create un Makefile che ne comprenda l'uso
- 3- ne aggiungete l'uso, il perché nella relazione finale e come si usano e cosa fanno le funzioni della libreria che usate nel progetto.
- 4- non servono a sostituire le librerie POSIX per quanto riguarda socket, thread, processi o chiamate di sistema.
- 5- non automatizzano eccessivamente le procedure il cui sviluppo è alla base del progetto

Q9: Nella spiegazione dei parametri opzionali del server, viene indicato: "il parametro opzionale --diz. Vuol dire che il comportamento standard nel caso in cui questo non sia presente è quello di utilizzare il file dictionary_ita.txt fornito con il testo?

A9: il parametro `--diz` è opzionale e serve a sostituire il file di dizionario di default, il file fornito `dictionary_ita.txt`, con un eventuale dizionario alternativo, avente lo stesso formato.

Q10: il parametro `seed` serve per generare le lettere della matrice. Nel caso in cui sia presente anche il parametro `--matrici` che dovrebbe fornire il file txt per le matrici, è scelta nostra implementativa se dare priorità a uno dei due (generazione casuale delle matrici con `seed` o file fornito) o restituire errore?

A10: Non serve soltanto per generare le lettere della matrice: Il parametro `seed` deve essere usato ovunque nel progetto abbiate bisogno di un numero casuale per inizializzare il generatore di numeri casuali e garantire la riproducibilità dei tests. Il consiglio è di utilizzare l'opportuna chiamata `srand` all'inizio della/le funzioni `main`, passando il `seed` come parametro della funzione `srand`, e successivamente chiamare le `rand` ove necessario (nel server e nei suoi threads, nel client e nei suoi threads). Nel caso sia usato il parametro `--matrici`, la matrice non deve essere generata casualmente ma utilizzare il file passato come argomento. Qualora entrambi i parametri `seed` e `matrici` fossero passati da riga di comando, il sistema deve ritornare un errore.

Q11: nel caso in cui venga specificato un file contenente le matrici, questo è passato via riga di comando tramite il parametro `--matrici` all'avvio del server. All'arrivo di un qualsiasi client, verrà creato un server thread per gestirlo. Ognuno di questi thread andrà a proporre al proprio client una matrice presa dal file. è necessario che ogni thread dia al client una matrice diversa rispetto a quelle assegnate da parte degli altri thread ai loro client nel rispettivo round?

penso che un ulteriore modo di porla sia, il file delle matrici va gestito come memoria condivisa tra i server thread? oppure ognuno può leggere sequenzialmente e fornire matrici al proprio client?

A11: tutti i client collegati in un certo momento partecipano al medesimo gioco "corrente" (a meno che il gioco non sia in pausa), quindi il server presenta la stessa matrice a tutti. La struttura dati corrispondente sarà quindi condivisa tra tutti i thread e generata da uno solo (un thread apposito, o il thread principale del server, a voi la scelta implementativa più idonea). Alla fine del gioco, verrà stilata una classifica dal thread scorer in modo da identificare un vincitore e comunicare tale informazioni ai clients.

Q12: Quando una partita termina e il server deve inviare i risultati ai client, questo procedimento avviene in modo asincrono oppure è il client che li richiede? Nel testo c'è scritto che quando il gioco è in pausa il server risponde anche a richieste del tipo MSG_PUNTI_FINALI.

A12: il server manda i risultati ai client che devono essere in grado di riceverli. Questo ovviamente introduce una certa complessità, assolutamente voluta. Successivamente alla comunicazione, fintanto che il server è in pausa, i client possono mandare ulteriori richieste di MSG_PUNTI_FINALI, ma la prima volta vengono inviati dal server ai client senza necessità che questi ultimi lo richiedano.

Q13: Discutendo con i miei colleghi riguardo al carattere speciale "Qu" e dopo aver letto le FAQ siamo ancora in dubbio su come approcciarci. La lettera Q esiste? Altrimenti parole come Soqqadro potrebbero non esistere. Dobbiamo considerare "Qu" come intercambiabile o semplicemente diamo alcune parole come inutilizzabili?

A13: Come già spiegato nella A1, il carattere Q da solo non esiste. Non è possibile avere Soqqadro? Vero! Ce ne faremo una ragione. Quindi, la Q esiste SOLO associata alla U, nella forma Qu.

Q14: Per quanto riguarda gli score finali e come vengono inviati mi sono chiesto se il client, a fine partita, debba bloccarsi ad aspettare i risultati (imponendo quindi al client, a fine partita, di ascoltare finché non ottiene il messaggio degli score e quindi una restrizione sui comandi, che non potranno essere eseguiti per un tot di tempo) oppure è meglio evitare tutto ciò e permettere sempre al client di inviare comandi in qualunque momento e di ricevere messaggi che verranno stampati quando arrivano?

A14: ti suggerisco di ragionare sulla natura bidirezionale dei socket. Una buona progettazione del client (suggerimento: multithread) permette di evitare queste problematiche.

Q15: Posso usare la libreria <stdatomic.h>? Non so se rientra nel punto 5 della domanda sulle librerie ossia nelle librerie che "non devono automatizzare eccessivamente le procedure il cui sviluppo è alla base del progetto".

A15: Non è esattamente nello spirito del progetto però credo possa essere educativo lasciarle esplorare per chi voglia farlo, quindi sì! Procedi pure. Mi raccomando motiva tutto nella relazione finale.

Q16: Nei vari punti del client c'è scritto: in altre parole fintanto che l'utente non si è correttamente registrato, l'invocazione di qualsiasi altro comando ad eccezione di fine determina un messaggio di errore. Non è più giusto poter permettere al client sia il comando fine che il comando aiuto? In questo modo un utente può capire cosa può e cosa non può fare, anche prima della registrazione del nome

A16: CORRETTO! Va benissimo permettergli anche di rispondere al comando aiuto.

Q17: Stavo pensando che per ogni client bisogna fare una ricerca nel dizionario che vorrebbe quindi dire aprire il file, leggere le parole e cercare quella che ha inserito l'utente; visto che si tratta di un'operazione molto costosa in tempo mi chiedevo se fosse ragionevole caricarsi il contenuto del file dizionario in una struttura dati che ha un tempo di ricerca elevato, ad esempio una tabella hash che anche al caso pessimo impiega circa lo stesso tempo della lettura dal file con però il problema di dover allocare una quantità enorme di byte per la tabella, che tra le altre cose non è neanche una dimensione fissa visto che potrebbe essere passato un'altro dizionario in input. Quindi la mia domanda era conviene caricarsi il dizionario in memoria per cercare più velocemente o il vantaggio che avremmo in tempo è poco utile vista la mole di spazio che dobbiamo allocare dinamicamente.

A17: Suggerisco di rappresentare il dizionario con una struttura dati chiamata TRIE (<https://en.wikipedia.org/wiki/Trie>) da caricare in memoria all'avvio del programma, che i

thread possono condividere in sola lettura, per evitare race conditions. Se non vi è chiaro come realizzarlo sentitevi liberi di contattarmi.

Q18: Se passo il file matrice al server, dove ogni riga rappresenta una matrice, per ogni gioco devo scegliere la prossima matrice in sequenza (primo gioco, prima riga; secondo gioco, seconda riga) o posso anche scegliere una riga casuale (primo gioco, riga random; secondo gioco, riga random)?

A18: Vanno scelte in sequenza.

Q19: Nel quinto punto della descrizione del server nella traccia del progetto si parla di messaggio di tipo "MSG_TEMPO_RESTANTE", anche se questo non è incluso nell'elenco di define presenti nella sezione 2.1) Protocollo di comunicazione della suddetta traccia. Provando ad inferire dal contesto quale tipo di dato fosse, ho pensato che si intendesse fare riferimento al tipo "MSG_TEMPO_ATTESA". è corretto? oppure MSG_TEMPO_RESTANTE è stato volutamente omissso perché va da noi implementato?

A19: Osservazione corretta. C'è un errore nel testo, MSG_TEMPO_RESTANTE è sbagliato, in realtà va fatto riferimento a MSG_TEMPO_ATTESA.

Q20: Quando l'utente si registra finisce lì e riceve solo il messaggio "Registrazione avvenuta con successo" o riceve anche la matrice e il tempo mancante insieme al messaggio di successo della registrazione e poi può riutilizzare il comando per riottenere? Oppure per ottenere la matrice ed il risultato deve fare necessariamente il comando matrice?

A20: Dopo che la registrazione è avvenuta con successo, il server comunica matrice e tempo mancante al client, senza attendere che quest'ultimo invochi ulteriori comandi.

Q21: Quando il gioco è in corso, il client può cominciare a proporre le parole. Ogni parola è proposta utilizzando il messaggio MSG_PAROLA. Se la parola è corretta, il server risponde con un messaggio di tipo MSG_PUNTI_PAROLA, se non lo è risponde con un messaggio MSG_ERR. Per "Correttezza della parola" cosa intendiamo? Cioè io devo restituire MSG_ERR nei casi in cui:

- non trovo la parola nel dizionario
- non trovo la parola nella matrice
- ho già utilizzato quella parola

O per quanto riguarda l'ultimo punto va bene anche inviare MSG_PUNTI_PAROLA con risultato 0?

A21: se la parola non è nel dizionario o non è nella matrice va restituito MSG_ERR, se ho già utilizzato quella parola mi deve essere restituito un messaggio MSG_PUNTI_PAROLA con valore pari a 0.

Q22: nella sezione 2.3 della traccia, dedicata al Client, sotto il punto di numero 2, quando si dice "l'invocazione di qualsiasi altro comando, ad eccezione di fine, determina un messaggio di errore." il messaggio di errore inteso è MSG_ERR oppure possiamo evitare di mandare richieste al server, e restituire un messaggio di errore dal lato client?

A22: Scegli la strategia che ritieni più opportuna, motivando la scelta nella relazione.

Q23: nel testo viene detto che il Server durante la pausa accetta solo messaggi del tipo: MSG_REGISTRA_UTENTE, MSG_MATRICE ed MSG_PUNTI_FINALI però quest'ultimo tipo di messaggio non viene mai specificato se il client possa mandarlo mediante un comando apposito, ad esempio punteggio, che manda MSG_PUNTI_FINALE.

A23: Corretto. Il che significa che è lasciato allo studente decidere se trattare il caso oppure no.

Q24: Quando viene detto che in caso di SIGINT il server deve terminare, la terminazione deve comportarsi come se la partita si fosse chiusa normalmente, oppure è sufficiente chiudere il socket creato dal server ed ovviamente i file descriptor con cui comunica ai client.

A24: Anche in questo caso la scelta è lasciata allo studente, che dovrà motivare nella relazione finale.

Q25: Dopo che un Client si disconnette dobbiamo rendere disponibile un'altro "slot" per la connessione oppure non ci interessa; più che altro questo implicherebbe che dopo che il numero massimo di client si sono connessi e disconnessi il server si debba chiudere

A25: Sì, va reso disponibile nuovamente lo slot precedentemente occupato.

Q26: Se ci viene passato il file delle matrici come parametro opzionale come dobbiamo comportarci in caso si arrivasse alla fine del file, ricominciamo a leggere dall'inizio oppure generiamo le matrici casualmente.

A26: Arrivato alla fine del file puoi ricominciare da capo oppure terminare il server.

Q27: Nell sezione 4 per la relazione viene scritto : "la struttura e la logica dei programmi di test" cosa si intende ?

A27: Si intende descrivere come sono fatti e perchè i programmi di test utilizzati per convalidare il progetto.

Q28: Il messaggio MSG_PUNTI_FINALI viene impacchettato ed inviato in formato CSV. Il client lo riceve. Basta stamparlo così come è oppure sistemarlo? Inoltre, sempre su questa linea volevo sapere se bisogna fare un sorting.

A28: La decisione è vostra. Fate quello che ritenete più opportuno e motivatelo nella relazione.

Q29: Viene detto questo nelle specifiche: "..dopo che il vincitore è stato nominato, comunicano ai rispettivi client il risultato finale". In che senso viene decretato un vincitore, devo inviare al client il nome del vincitore o, di nuovo, basta il messaggio in CSV con il nome del vincitore come primo? Anche perché viene detto di usare un messaggio ("La comunicazione ai client del risultato finale avviene tramite il messaggio MSG_PUNTI_FINALI").

A29: Trovate un modo ragionevole, implementatelo e descrivetelo nella relazione.

Q30: E' possibile usare le funzioni della libreria sys/socket.h send() e recv() per comunicare tra server e client, oppure dobbiamo inviare e leggere usando read() e write()?

A30: Decisamente preferibile usare la read e la write a meno di necessità specifiche.

Q31: Non mi torna la risposta alla Q10. Se il parametro seed non serve solamente per generare la matrice casuale, allora perché il programma dovrebbe dare errore nel caso passassi sia --seed che --matrici? Non dovrei poter passare seed comunque, senza usarlo per generare la matrice?

A31: Se lo usi anche per altro sei libero di non dare errore e motivare la scelta nella relazione.