

Programmazione I
Prova di programmazione – 23 Febbraio 2023 – 2 ore

Partendo dal frammento di codice fornito, realizzare un controllore simulato della velocità di un veicolo. Il controllore abbassa o alza la tensione applicata al motore del veicolo, con lo scopo di mantenere la velocità del veicolo più vicina possibile ad un valore desiderato. L'input al controllore è la differenza d tra la velocità attuale del veicolo e la velocità desiderata. Quindi $d < 0$, $d == 0$ e $d > 0$ e significano che il veicolo sta andando, rispettivamente, minore, uguale o maggiore di quella desiderata. Il controllore legge il valore di d ad intervalli regolari, e lascia inalterata, oppure aumenta o diminuisce di un valore costante k , la tensione v del motore, a seconda che d sia, rispettivamente, uguale a 0, minore di 0 o maggiore di 0. Il programma fornisce le seguenti funzionalità.

1. **[+2] esegui_controllo_vuoto(n , $d0$)** Esegue il controllore a vuoto, ossia senza modificare il valore di v , per n intervalli di tempo, su un veicolo che si trova a differenza di velocità $d0$ al primo intervallo. Non vi è alcun limite sul valore massimo di n . Si ottiene il punteggio aggiuntivo massimo se si evitano deallocazioni e riallocazioni di memoria non necessarie. Per il collaudo: fare leggere il valore di $d0$ anche se $n == 0$,

2. **stampa_controllo** Stampa l'ultima sequenza di valori assunti dalla differenza d , a seguito dell'ultima esecuzione del controllo di cui al punto 1 o 3. Ad esempio, se il controllo è stato eseguito con $n == 7$, $d0 == 8$ e $\delta == 6$:

8 2 -4 2 -4 2 -4

3. **[3] esegui_controllo(n , $d0$, δ)** Uguale alla funzionalità 1, ma eseguendo realmente il controllore, ed assumendo che il valore di d aumenti o diminuisca di un valore intero positivo δ se la tensione v viene aumentata o diminuita del valore k .

4. **[2, +1] salva_controllo** Salva l'ultima sequenza di valori assunti dalla differenza d in un file di testo dal nome predefinito. Punteggio aggiuntivo se si evita replicazione del codice.

5. **[2, +2] carica_controllo** Carica la sequenza dal file. L'eventuale precedente sequenza è persa. Si ottiene il punteggio aggiuntivo massimo se si evitano deallocazioni e riallocazioni di memoria non necessarie e si evita replicazione del codice.

6. **[4] esegui_controllo2($d0$, δ)** Uguale alla funzionalità 3, ma senza il passaggio del valore di n , e con una ottimizzazione per raggiungere il valore 0 più rapidamente. L'ottimizzazione è la seguente: al primo intervallo cambia il valore di v di una costante k , ma, per gli intervalli successivi, applica la seguente logica: se il cambiamento di v dell'intervallo precedente non ha fatto cambiare segno a d , allora nel prossimo intervallo aumenta o diminuisce v di una costante che fa cambiare d di una unità in più rispetto all'intervallo precedente, altrimenti aumenta o diminuisce v di una costante che fa cambiare d di una unità in meno rispetto all'intervallo precedente. Calcola automaticamente n come il valore più piccolo tale che l'ultimo valore di d è uguale a quello precedente. Ad esempio, se si esegue il controllo con $d0 == 8$ e $\delta == 6$, allora la successiva stampa produrrebbe:

8 2 -5 1 -4 0

I parametri di ingresso delle funzionalità sono solo indicativi. Gestire opportunamente le situazioni di errore, tranne l'*overflow* e l'inserimento di dati in formato errato da *stdin*.

Per il collaudo: se fate stampare messaggi per invitare l'utente ad inserire dell'input, ricordate di aggiungere la stampa di caratteri accapo. Altrimenti nel puro output del programma vi saranno delle righe fuse, e di fatto tale output non sarà quello che credete (le righe fuse non le vedete quando usate il programma da terminale, perché inserite voi l'accapo da utenti).

REGOLE

- Si può utilizzare ogni genere di manuale e di materiale didattico
- Per superare la prova, bisogna svolgere almeno i punti 1 e 2. Se si svolgono solo tali punti, il programma deve essere perfettamente funzionante. Il voto ottenuto in questo caso è 18.
- Ciascuna funzionalità DEVE essere implementata mediante almeno una funzione.
- Il voto massimo si ottiene se
 - a) il programma è perfettamente funzionante in ogni sua parte
 - b) tutti i principi di ingegneria del codice visti nel corso sono stati applicati
 - c) sono state seguite eventuali altre indicazioni presenti nella traccia in merito al voto finale