

HOMEWORK 1 (semplice)

Scrivete un programma in Go che conta il numero di volte in cui un determinato carattere "x" compare in una stringa. Il programma deve utilizzare la concorrenza, avviando una goroutine per ogni carattere nella stringa e verificando se il carattere corrisponde al carattere cercato.

Esempio: se la stringa è "aaaaaaaaaaaaabbbbbbbbbbccccdddddccccccfff" e il carattere da cercare è 'c', il programma dovrebbe avviare una goroutine per ogni carattere della stringa e utilizzare un meccanismo di sincronizzazione (come un WaitGroup) e un channel per tenere traccia del conteggio totale dei caratteri corrispondenti.

Inizializzare nel main una stringa di test e il carattere da cercare, e.g.:

```
stringa := "aaaaaaaaaaaaabbbbbbbbbbccccdddddccccccfff"
carattere := 'c'
```

Alla fine del processo, il programma deve stampare il conteggio finale dei caratteri corrispondenti. Nel nostro esempio, il conteggio finale è 11, poiché il carattere 'c' compare 11 volte nella stringa.

HOMEWORK 2

Scrivete un programma in GO che simuli un'agenzia di noleggi d'auto che deve gestire le prenotazioni di 10 clienti. Ogni cliente noleggia un veicolo tra quelli disponibili: Berlina, SUV o Station Wagon.

- Creare la struttura Cliente con il campo "nome"
- Creare la struttura Veicolo con il campo "tipo"
- Creare la function "noleggia" che prende come input un cliente e che prenota uno a caso tra i veicoli. Questa function deve anche stampare che il cliente x ha noleggiato il veicolo y.
- Creare una function "stampa" che, alla fine del processo, stampa il numero di Berline, SUV e Station Wagon noleggiati.
- Ogni cliente può noleggiare un veicolo contemporaneamente ad altri.

Si noti che si possono creare ulteriori funzioni per risolvere il problema, oltre alle due obbligatorie, descritte sopra.

HOMEWORK 3

Scrivere un programma in Go che simuli la produzione di 5 torte da parte di 3 pasticceri. La produzione di ogni torta richiede 3 fasi che devono avvenire in ordine: prima la torta viene cucinata, poi guarnita e infine decorata.

Il primo pasticcere si occupa solo di cucinare le torte e ci mette 1 secondo per ogni torta. Questo pasticcere ha a disposizione 2 spazi per appoggiare le torte una volta che ha finito di cucinarle. Se ci sono spazi liberi, può iniziare a cucinare la torta successiva

senza aspettare che il secondo pasticcere si liberi per guarnire quella appena cucinata. Il secondo pasticcere si occupa solo di guarnire le torte e ci mette 4 secondi per ogni torta. Anche questo pasticcere ha a disposizione 2 spazi per appoggiare le torte una volta che ha finito di guarnirle. Il terzo pasticcere si occupa solo di decorare le torte e ci mette 8 secondi per ogni torta. I tre pasticceri lavorano contemporaneamente.

HOMEWORK 4

Scrivere un programma in Go che simuli un'attività di trading di valute in un mercato fittizio.

Il programma deve simulare usando la concorrenza tre coppie di valute: EUR/USD, GBP/USD e JPY/USD, e simulare le operazioni di acquisto e vendita in parallelo.

Creare una funzione "simulateMarketData" che simuli il prezzo delle coppie di valute e invii i dati simulati su un canale. In particolare:

- Il prezzo della coppia EUR/USD varia casualmente tra 1.0 e 1.5.
- Il prezzo della coppia GBP/USD varia casualmente tra 1.0 e 1.5.
- Il prezzo della coppia JPY/USD varia casualmente tra 0.006 e 0.009.

I prezzi vengono generati e inviati sul canale corrispondente ad intervalli regolari, in particolare ogni secondo.

Creare una funzione "selectPair" che utilizza una "select" per gestire le operazioni di vendita e acquisto in base alle condizioni specificate. In particolare:

- Se il prezzo di EUR/USD supera 1.20, deve vendere EUR/USD. Simulare la vendita con un tempo di 4 secondi, cioè inserire un delay di 4 secondi prima di confermare la vendita.
- Se il prezzo di GBP/USD scende sotto 1.35, deve acquistare GBP/USD. Simulare l'acquisto con un tempo di 3 secondi, cioè inserire un delay di 3 secondi prima di confermare l'acquisto.
- Se il prezzo di JPY/USD scende sotto 0.0085, deve acquistare JPY/USD. Simulare l'acquisto con un tempo di 3 secondi, cioè inserire un delay di 3 secondi prima di confermare l'acquisto.

Il programma deve eseguire il ciclo di trading per un minuto e alla fine del ciclo deve terminare.