

1. Realizzate il crivello di Eratostene, un metodo per calcolare i numeri primi noto agli antichi greci. Scegliete un numero n : questo metodo calcolerà tutti i numeri primi fino a n . Come prima cosa inserite in un **Set** tutti i numeri da 2 a n . Poi, cancellate tutti i multipli di 2 (eccetto 2); vale a dire 4, 6, 8, ... Dopodiché cancellate tutti i multipli di 3 (eccetto 3), cioè 6, 9, 12, ... Arrivate fino a $n/2$, quindi visualizzate il **Set**.
2. Usate una pila per invertire le parole di una frase. Continuate a leggere parole, aggiungendole alla pila, fin quando non trovate una parola che termina con un punto. A questo punto estraete tutte le parole dalla pila e visualizzatele. Realizzate la pila tramite una **Deque**.
3. Dovete realizzare un elenco di cose da fare (to do list). A ciascun compito viene assegnata una priorità, un numero intero da 1 a 9, e una descrizione. Quando l'utente digita il comando *add priorità descrizione* il programma aggiunge una cosa da fare, mentre quando l'utente digita *next* il programma elimina e visualizza la cosa da fare più urgentemente. Il comando *quit* termina il programma. Risolvete il problema usando una **PriorityQueue**.

SIMULAZIONE PARTITA DI TENNIS (prova d'esame di qualche anno fa)

Il tennis è uno sport che vede opposti due giocatori (incontro singolare). Gli incontri hanno dei punteggi che si suddividono in partite, in inglese set; in base ai tipi di torneo gli incontri sono al meglio dei 3 o 5 set; una partita è costituita da 6 o più giochi, (games). Se i giocatori sono sul punteggio di 6-6 si continua finché uno dei due giocatori vince con due giochi di scarto.

Scrivete le classi necessarie per la simulazione di una partita di singolare di tennis al meglio dei 3 set (la partita viene giocata da due giocatori e vince chi conquista 2 set su 3).

La classe **Giocatore** deve contenere la posizione in classifica del giocatore (int), nome e cognome (String). La classe **Incontro** al suo interno contiene due oggetti della classe **Giocatore**, 3 oggetti della classe **Set** ed il metodo per effettuare la simulazione (quindi per giocare i set finché uno dei due giocatori ne ha vinti 3). La classe **Set** conterrà la sequenza dei risultati parziali del set (ad esempio: 0-0, 1-0, 1-1, 2-1, 3-1, 3-2, 4-2, 4-3, 5-3, 6-3) ed il metodo per effettuare la simulazione. La simulazione di un set avviene mediante la generazione di numeri casuali che stabiliscono se un gioco è stato vinto dal primo o dal secondo giocatore e dalla verifica dell'avvenuta vittoria da parte di un giocatore.

Il programma in particolare deve avere le seguenti funzionalità:

- 1) creare due oggetti **Giocatore**
- 2) creare un oggetto **Incontro** con i due giocatori appena creati
- 3) avviare la simulazione dell'incontro
 - a. avviare la simulazione del primo set
 - b. avviare la simulazione del secondo set
 - c. se necessario avviare la simulazione del terzo set
- 4) stampare il risultato dettagliato di tutti i set giocati
- 5) stampare i dati del vincitore

Per esempio, una esecuzione del programma potrebbe essere la seguente:

```
Inserisci dati giocatore 1: Mario Rossi 25
Inserisci dati giocatore 2: Luca Neri 32
Inizia la partita!
Simulazione set 1: vince Mario Rossi
Simulazione set 2: vince Luca Neri
Simulazione set 3: vince Luca Neri
Ha vinto la partita Luca Neri
Set 1: 0-0, 1-0, 2-0, 2-1, 3-1, 4-1, 5-1, 6-1
Set 2: 0-0, 1-0, 1-1, 1-2, 1-3, 2-3, 3-3, 3-4, 4-4, 5-4, 5-5, 5-6, 6-6, 6-7, 6-8
Set 3: 0-0, 0-1, 0-2, 0-3, 0-4, 1-4, 2-4, 2-5, 3-5, 3-6
```

NOTE PER COMPILAZIONE E TEST A RIGA DI COMANDO IN AMBIENTE LINUX:

```
javac -d ../classes -cp ../classes nomeClasse.java compila e genera il bytecode
java -cp ../classes nomePackage.nomeClasse esegue il bytecode sulla JVM
```