

Cognome e Nome		Matricola	
----------------	--	-----------	--

Prova 1

Si supponga di avere cinque processi che arrivano nel sistema al tempo di arrivo specificato nella seguente tabella, dove sono indicate anche le durate dei CPU burst:

<i>Processo</i>	<i>Tempo di Arrivo</i>	<i>CPU Burst</i>
<i>P1</i>	0	13
<i>P2</i>	3	9
<i>P3</i>	8	2
<i>P4</i>	12	8
<i>P5</i>	16	11

Si mostri la **sequenza di esecuzione** e si calcolino il **tempo di risposta** ed il **tempo di completamento** di ciascun processo, considerando i seguenti algoritmi di scheduling:

1. FCFS
2. SJF con prelazione

Prova 2

Si descriva, anche con l'aiuto di schemi e/o grafici opportunamente commentati, l'architettura di paginazione con TLB (transaction look-aside buffer).

Prova 3

Un cementificio, specializzato nella vendita di sacchi di cemento, è in grado di accogliere al massimo **N** clienti contemporaneamente, e contiene all'inizio **P** sacchi in vendita. I clienti si recano nel cementificio per acquistare un certo numero di sacchi. Ogni cliente inizialmente sceglie il numero di sacchi da acquistare (un numero casuale compreso tra 10 e 30), dopodiché entra nel cementificio, preleva tutti i sacchi di cemento, ed esce dal cementificio. I clienti in attesa di entrare nel cementificio devono essere risvegliati in ordine FIFO. Ogni cliente preleva i sacchi di cemento uno alla volta: tra un prelievo e l'altro il cliente attende 1 minuto; nel caso in cui i sacchi siano esauriti, il cliente si sospende in attesa che ritornino disponibili. Ogni volta che i sacchi si esauriscono un addetto che lavora all'interno del cementificio li riporta al valore iniziale impiegando per questa operazione 5 minuti. L'operazione di rifornimento non è bloccante per l'ingresso dei clienti nel cementificio, ovvero questi possono liberamente continuare ad entrare ed uscire (nel caso in cui abbiano terminato l'acquisto) dal cementificio, ma non possono ovviamente prelevare sacchi.

Si modelli il sistema descritto in Java, dove i **clienti** e l'**addetto** sono dei thread che interagiscono tramite un oggetto **cementificio**. Il cementificio espone tre metodi, *entra*, *esci* e *preleva* che permettono ai clienti rispettivamente di entrare e di uscire dal cementificio e di prelevare un sacco, e di due metodi *iniziaRifornimento* e *terminaRifornimento* che permettono all'addetto di iniziare e terminare il rifornimento del cementificio. Si implementino due soluzioni che riproducano il funzionamento del problema sopra descritto utilizzando:

la classe *Semaphore* del package *java.util.concurrent*

gli strumenti di mutua esclusione e sincronizzazione del package *java.util.concurrent.locks*.

Si scriva infine un *main* d'esempio che, facendo uso di una delle due soluzioni precedenti, inizializzi un **cementificio** con **N=100** e **P=1000**, inizializzi **200 clienti**, un **addetto**, e ne avvii l'esecuzione.