

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 indicati anche i tempi di CPU burst:

<i>Processo</i>	<i>Tempo di Arrivo</i>	<i>CPU Burst</i>
<i>P1</i>	0	11
<i>P2</i>	4	5
<i>P3</i>	7	6
<i>P4</i>	14	4
<i>P5</i>	27	11

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

- **FCFS**
- **SJF con prelazione**

**Prova 2**

Si descriva, anche con l'aiuto di schemi e/o grafici opportunamente commentati, come funziona la memoria virtuale.

**Prova 3**

Una banca è dotata di **30 sportelli** bancomat distribuiti in un'area urbana. Ogni sportello ha una capacità di denaro pari a 10.000 ed è connesso con **una sede centrale**, la quale conosce l'ammontare di denaro presente in ogni momento, in ogni sportello. **Un agente** della banca gira per la città con un furgoncino, dotato di una quantità infinita di denaro, effettuando ciclicamente le seguenti operazioni:

- chiede alla sede centrale di indicargli quale sportello contiene la minore quantità di denaro;
- impiega 10ms per raggiungere quello sportello;
- rifornisce lo sportello, riportandone la disponibilità a 10.000, impiegando per questa operazione 50ms.

La banca ha **100 clienti**, ognuno dei quali effettua ciclicamente le seguenti operazioni:

- sceglie a caso uno sportello bancomat;
- impiega 10ms per raggiungere quello sportello;
- cerca di prelevare da quello sportello una quantità di denaro compresa casualmente tra 50 e 250, aspettando in ordine FIFO qualora lo sportello sia impegnato con un'altra operazione; se c'è denaro sufficiente effettua il prelievo impiegando 5 ms per tale operazione e poi abbandona lo sportello; se il denaro non è sufficiente, abbandona lo sportello senza effettuare il prelievo;
- se è riuscito a prelevare, aspetta per un numero di ms pari alla quantità di denaro prelevata.

Si modellino in Java i clienti e l'agente attraverso dei Thread e si implementino due soluzioni che riproducano il funzionamento del problema sopra descritto utilizzando:

1. la classe *Semaphore* del package *java.util.concurrent*
2. gli strumenti di mutua esclusione e sincronizzazione del package *java.util.concurrent.locks*.

Si scriva infine un *main* d'esempio che faccia uso di una delle due soluzioni precedenti.