

Esercizio 1 (10 punti)

In un sistema sono presenti quattro processi P_1, P_2, P_3, P_4 che usano quattro tipi di risorsa A, B, C, D di cui sono presenti rispettivamente 6, 1, 3 e 1 unità. I processi hanno le seguenti matrici di allocazione delle risorse e di richiesta di risorse al tempo T_0 :

	Allocazione				Req			
	A	B	C	D	A	B	C	D
P_1	1	0	1	0	0	0	0	0
P_2	2	0	0	1	X	1	0	0
P_3	1	0	2	0	0	0	0	1
P_4	2	1	0	0	0	0	Y	0

Dove X e Y sono valori interi maggiori o uguali a zero (compatibili con risorse totali presenti).

Usando l'**algoritmo di rilevazione dello stallo** stabilire per quali valori di X e Y il sistema non si trova in stallo, giustificare il risultato ottenuto.

Esercizio 2 (20 punti)

Si vuole simulare un sistema di web crawling, nel quale sono presenti N thread *Crawler*, dove ognuno preleva una URL da un oggetto *UrlStore* che contiene le Url che devono essere analizzate (attende se vuoto), effettua il download della pagina HTML (si simuli questa attività con una attesa), la salva in un oggetto *DocStore*, incrementa una variabile globale che indica il numero di download effettuati da tutti gli N *Crawler* e quindi preleva una nuova URL e procede. Sono presenti anche M thread *Parser* dove ognuno preleva dal *DocStore* una pagina HTML (attende se vuoto), effettua il parsing della pagina HTML (si simuli questa attività con una attesa), produce l'insieme dei link che sono presenti nella pagina aggiungendoli nell'oggetto *UrlStore* ed infine incrementa variabile globale che indica il numero di pagine analizzate dagli M *Parser*. Ai fini della simulazione si ipotizza che da ogni pagina HTML vengano estratte un numero casuale di nuove URL comprese tra 0 e 10. Per rappresentare le URL e le pagine HTML si usino oggetti *String* e per memorizzare le Url e pagine HTML si usi la classe *ArrayList* (che NON è thread-safe) che ha metodo *add* per inserire in coda e metodo *remove(0)* per estrarre dalla testa.

Il programma principale deve inserire nel *UrlStore* una URL iniziale, avviare i thread e terminare l'esecuzione dei thread dopo 60 secondi e ogni secondo scriva quanti download e parsing sono stati fatti.

Implementare il sistema descritto in Java usando i semafori per la sincronizzazione tra thread.