

1. Een programma bestaat uit 5 threads T1 t/m T5. Deze threads gebruiken 4 gedeelde resources R1 t/m R4. Elke resource is mutual exclusive beveiligd met een mutex M1 t/m M4. Deze mutexen gebruiken het **priority inheritance protocol**. Als een thread een resource toegewezen heeft gekregen dan kan deze thread de resource voor een bepaalde maximale tijd bezetten. In de onderstaande tabel staat k voor het nummer van de resource en $C(k)$ voor de maximale tijd dat resource k bezet kan worden.

K	$C(k)$
1	20
2	15
3	10
4	5

In de volgende tabel staat aangegeven welke resources elke thread gebruikt. In de onderstaande tabel staat i voor het nummer van de thread, $T(i)$ voor de periodetijd van thread i en $C(i)$ voor de maximale executietijd van thread i . Gegeven is dat de deadline van elke taak gelijk is aan zijn periodetijd.

i	$T(i)$	$C(i)$	Gebruikt
1	100	50	R1, R3
2	200	45	R2, R3, R4
3	350	20	R2
4	300	15	-
5	400	25	R1

Alle gegeven tijden zijn in ms.

- A. (2) De bekende “utilisation based schedulability test” is hier **niet** toe te passen. Noem **de** reden waarom deze test in dit geval **niet** gebruikt kan worden.
- B. (4) Bepaal de prioriteit van de 5 processen als RMPA (**Rate** Monotonic Priority Assignment) wordt gebruikt. De hoogste prioriteit is 5 en de laagste prioriteit is 1.
- C. (10) Bereken voor elke thread i de blocking time $B(i)$.
- D. (14) Bereken voor thread 1, 3 en 5 of de deadline wordt gehaald en geef indien de deadline wordt gehaald de response tijd R_i .