1. (a) The minimum value of K that may cause the system deadlock is 4. Since we have 8 printers and each of the processes can take no more than 3 printers, if $K = 3$, then even the 3 processses require 3 printer at the same time, we can always satisfy two of them first, then the two process will finish after a certain time and give back their resources and then we can satisfy the third one. However, if $K = 4$, when all 4 processes have taken 2 printers and are requiring for one more printer, then the four conditions that will cause a deadlock are all satisfied, and there will be a deadlock.

   (b) No. The reason is that, no matter how big K is, there is a possibility that all the processes just need one or two printers, and in such a case, we can find a safe sequence, thus the system is in safe sate and there will be no deadlock.

2. (a) From the table we know that

$$\mathbf{MAX} = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 2 & 7 & 5 & 0 \\ 6 & 6 & 5 & 6 \\ 4 & 3 & 5 & 6 \\ 0 & 6 & 5 & 2 \end{pmatrix} \qquad \mathbf{Allocation} = \begin{pmatrix} 0 & 0 & 1 & 2 \\ 2 & 0 & 0 & 0 \\ 0 & 0 & 3 & 4 \\ 2 & 3 & 5 & 4 \\ 0 & 3 & 3 & 2 \end{pmatrix}$$

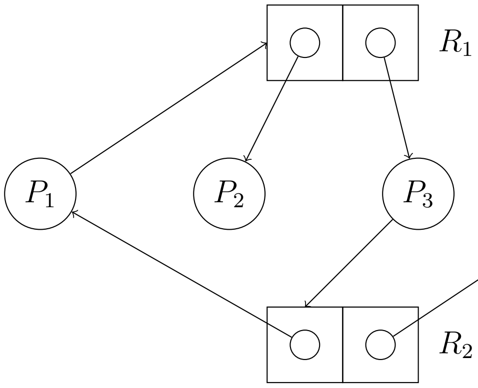
   Since $\mathbf{Need} = \mathbf{MAX} - \mathbf{Allocation}$, we have

$$\mathbf{Need} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 7 & 5 & 0 \\ 6 & 6 & 2 & 2 \\ 2 & 0 & 0 & 2 \\ 0 & 3 & 2 & 0 \end{pmatrix}$$

   (b) Yes. Because we can find a safe sequence by banker's algorithm, that is, P_0, P_3, P_4, P_1, P_2.

   (c) We know that $\mathbf{Request_2} = (0, 2, 0, 0)$ and $\mathbf{Available} = (2, 1, 0, 0)$.Since $\mathbf{Request_2} \leq \mathbf{Need_2}$, but $\mathbf{Request_2} \not\leq \mathbf{Available}$, thus P_2 has to wait until the resources are available.

3. (a)



   (b) Yes. The cycle is $P_1 \to R_1 \to P_3 \to R_2 \to P_1$. We can name it C_{P_1,P_3}.

   (c) No. A possible sequence of executions is P_2, P_4, P_1, P_3.

4. From the table we know that

$$\mathbf{MAX} = \begin{pmatrix} 1 & 1 & 2 & 1 & 3 \\ 2 & 2 & 2 & 1 & 0 \\ 2 & 1 & 3 & 1 & 0 \\ 1 & 1 & 2 & 2 & 1 \end{pmatrix} \qquad \mathbf{Allocation} = \begin{pmatrix} 1 & 0 & 2 & 1 & 1 \\ 2 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

   Since $\mathbf{Need} = \mathbf{MAX} - \mathbf{Allocation}$, we have

$$\mathbf{Need} = \begin{pmatrix} 0 & 1 & 0 & 0 & 2 \\ 0 & 2 & 1 & 0 & 0 \\ 1 & 0 & 3 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{pmatrix}$$

   By comparing $\mathbf{Need_i}$, where $i \in \{1, 2, 3, 4\}$ with **Available**, we know that if we want the system in a safe state, then $X \geq 1$. Let $X = 1$, we can find a safe sequence, that is, D, A, C, B. Thus, the smallest value of X for which the system is in a safe state is 1.