Indian National Olympiad in Informatics, 2002
Suggested solutions to problems

1. There are at least two straightforward solutions.

   Non-recursive solution Let the original contents be (in sequence) $a_1a_2\dots a_{p-1}a_pa_{p+1}\dots a_n$. First, let a_p "bubble" forward to the first box as follows.

   - Swap contents of box p with box $p-1$. Contents are now $a_1a_2\dots a_pa_{p-1}a_{p+1}\dots a_n$.
   - Swap contents of box $p-1$ with box $p-2$. Contents are now $a_1a_2\dots a_pa_{p-2}a_{p-1}a_{p+1}\dots a_n$.
   - $\cdots$
   - Swap contents of box 2 with box 1. Contents are now $a_pa_1a_2\dots a_{p-2}a_{p-1}a_{p+1}\dots a_n$.

   Similarly, move a_{p+1} to box 2— repeat the earlier procedure starting with box $p+1$ and stopping with box 2 to get $a_pa_{p+1}a_1a_2\dots a_{p-2}a_{p-1}a_{p+2}\dots a_n$.

   Do the same for $p+2$ to 3, $p+3$ to 4, …, n to $(n-p)+1$ so that eventually we have $a_pa_{p+1}\dots a_na_1a_2\dots a_{p-1}$.

   Recursive solution

   Case 1 Let $p \geq n/2$. Swap the contents of box 1 with box p, box 2 with box $p+1$, …, box $n-p+1$ with box n to get a new arrangement $a_pa_{p+1}\dots a_na_{n-p+2}\dots a_{p-1}a_1a_2\dots a_{n-p+1}$.

   Case 2 If $p < n/2$, instead swap the contents of box 1 with box $n-p+2$, box 2 with box $n-p+3$, …, box $p-1$ with box n to get a new arrangement $a_{n-p+2}\dots a_na_{p+1}\dots a_{n-p}a_1a_2\dots a_{p-1}$.

   After the first step, in case 1 ($p \geq n/2$), recursively apply the same procedure for boxes $n-p+2,\dots,n$ with appropriate new values $n' = n-p+1$ and $p' = n-2p+3$ so that contents $a_{n-p+2}\dots a_{p-1}$ of the $n-2p+2$ boxes numbered $n-p+2,\dots,p-1$ is interchanged with contents $a_1a_2\dots a_{n-p+1}$ of the $p-1$ boxes numbered $n-p+1,\dots,n$.

   Similarly, in the second case ($p < n/2$), recursively apply the procedure for boxes $1,\dots,n-p$ with $n' = n-p+1$ and $p' = n-2p+3$ so that contents $a_{n-p+2}\dots a_{n-2p}$ of the $n-2p+2$ boxes numbered $n-p+2,\dots,p-1$ is interchanged with contents $a_1a_2\dots a_{n-p+1}$ of the $p-1$ boxes numbered $n-p+1,\dots,n$.

2. (a) If a player has to play when the stack has 3 coins, he loses—whether he takes 1 or 2 coins, the opponent can empty the stack on the next move and thus win the game.

   If a player plays when the stack has 6 coins, the opponent can leave the stack with 3 coins after his next move, leading to a losing position for the first player.

In general, it follows that any player who moves when the number of coins in the stack is a multiple of 3 loses— the opponent can reduce it each time to the next lower multiple of 3 until eventually the stack reaches exactly 3 coins, which is a losing position.

Conversely, if the number of coins is not a multiple of 3, then the player who moves can reduce it to a multiple of 3 and this is a losing position for the opponent.

Thus, if the number of coins in initial stack is *not* a multiple of 3, Player A has a winning strategy and if the number of coins *is* a multiple of 3, Player B has a winning strategy. The winning strategy in either case is to reduce the number of coins to the next lower multiple of 3.

(b) If there are two stacks, a losing position is one in which the *difference* between the two stacks is a multiple of 3. If a player plays when the difference is a multiple of 3, then the opponent can always restore the difference to be a multiple of 3. Eventually, when one of the two stacks becomes empty, the invariant guarantees that the opponent can make the number of coins in the nonempty stack a multiple of 3, which is a losing position for first player in the single stack game (by the first part of this question).

Therefore, Player A can always win if, in the initial arrangement, the difference between the number of coins in the two stacks is *not* a multiple of 3, and Player B can always win if the initial difference *is* a multiple of 3. The winning strategy in both cases is to restore the difference to a multiple of 3 so long as both stacks are nonempty and then switch to the winning strategy for a single stack after one of the stacks becomes empty. (Actually, the strategy does not really change after one stack becomes empty—when one of the stacks is empty, if the difference between the stacks is a multiple of 3 then the size of the nonempty stack must be a multiple of 3.)

3. (a) Move the first 8 carriages to the rearranging yard and rearrange them in the *reverse* order to which they would appear in the final arrangement. After rearranging them, transfer these 8 carriages into siding 1. Do the same for the next 8 carriages and move them into siding 2. Now, reassemble the final train into the correct order by *merging* the carriages from the two sidings, one at a time—that is, compare the next available carriage in both sidings and pull out the one that is to be placed next in the final arrangement.

For instance, in the example given in the question, the incoming train is arranged as

$$c_1c_2c_3c_4c_5c_6c_7c_8c_9c_{10}c_{11}c_{12}c_{13}c_{14}c_{15}c_{16}$$

and the outgoing train is required to be

$$c_1c_3c_5c_7c_9c_{11}c_{13}c_{15}c_2c_4c_6c_8c_{10}c_{12}c_{14}c_{16}.$$

The correct order for the first 8 carriages $c_1c_2c_3c_4c_5c_6c_7c_8$ in the final train is $c_1c_3c_5c_7c_2c_4c_6c_8$. We thus rearrange the carriages as $c_8c_6c_4c_2c_7c_5c_3c_1$ and push them into siding 1, with c_1 as the outermost carriage. Likewise, we rearrange the next 8 carriages as $c_{16}c_{14}c_{12}c_{10}c_{15}c_{13}c_{11}c_9$ and push them into siding 2 with c_9 as the outermost carriage. We now merge these two 8 carriage trains by pulling c_1, c_3, c_5 and c_7 from siding 1. At this point, the next available carriage in siding 1 is c_2 while the next available carriage in siding 2 is c_9. Since we need c_9 before c_2 in the final train, we switch to siding 2, pull out $c_9, \ldots, c_{15}$, switch back to siding 1 and pull out $c_2, \ldots, c_8$ and finally go back to siding 2 and pull out $c_{10}, \ldots, c_{16}$.

(b) For $n = 1$, we can rearrange any train with $2^1 = 2$ carriages with 1 siding—if the carriages have to be reversed, push the first carriage into the siding, move the second carriage out to the platform and hook up the first carriage before it.

Inductively, assume that with $n{-}1$ sidings, we can rearrange 2^{n-1} carriages in any order we want. We have to show that with n sidings at our disposal, we can rearrange a train with 2^n carriages. Observe that a train with 2^n carriages can be broken up into 2 trains with 2^{n-1} carriages each. By our inductive assumption for $n{-}1$, we can use sidings $1, \ldots, n{-}1$ to rearrange the first half of the train in the the final order that we want and then reverse the train into siding n, thus leaving sidings $1, \ldots, n{-}1$ free again. We use these $n{-}1$ sidings to generate (the reverse of) the order that we want for the second half of the train and put it into siding $n{-}1$. We then merge the two trains of length 2^{n-1} from sidings $n{-}1$ and n as described in the first part to obtain the rearrangement that we want for the overall train of size 2^n.