Zonal Informatics Olympiad, 2013

Instructions to candidates

1. The duration of the examination is 3 hours.
2. Calculators, log tables and other aids are not permitted.
3. The question paper carries 80 marks, broken up into four questions of 20 marks each. Each question has three parts. *If you solve all three parts correctly, you get 20 marks for that question.* Otherwise, you get 5 marks for each part that you solve correctly.
4. Attempt all questions. There are no optional questions.
5. There is a separate *Answer Sheet*. To get full credit, you *must* write the final answer in the space provided on the Answer Sheet.
6. Write *only* your final answers on the Answer Sheet. Do *not* use the Answer Sheet for rough work. Submit all rough work on separate sheets.
7. Make sure you fill out your contact details on the Answer Sheet as completely and accurately as possible. We will use this information to contact you in case you qualify for the second round.

This page intentionally blank

Zonal Informatics Olympiad, 2013

Questions

1. You have N registers, numbered 1 to N, each of which can hold an integer value. You are given the initial values of the registers, which have the property that every number from 1 to N occurs exactly once among the N registers.

   Each register has a "reset button": pressing the reset button on register i changes its value to i. In one *move* you can pick any subset of the registers (say, registers 3, 5, and 9) and simultaneously press all their reset buttons. However you must ensure that every number from 1 to N continues to occur exactly once amongst the N registers.

   The cost of a move that resets m registers simultaneously is m^2. You can perform a sequence of such moves one after the other, and the total cost is the sum of the costs of the individual moves.

   Register i is said to be *stable* if it contains the value i. Given a target K, where $K \leq N$, the goal is to perform a sequence of moves at the end of which at least K registers are stable. Find the minimum possible cost for achieving this.

   For example, suppose $N = 6$ and $K = 3$ and the initial values of registers 1 to 6 are, in order, $1, 5, 6, 3, 2, 4$. In one move, you can reset registers 2 and 5. At this point registers 1, 2, and 5 are stable, so we can stop. The cost of this sequence of moves is $2^2 = 4$, and you can check that this is the minimum possible cost.

   In each of the cases below, you are given the initial contents of the N registers and the value of K. You have to compute the minimum cost of a sequence of moves that results in at least K registers becoming stable.

   (a) $K = 7$,

Register	1	2	3	4	5	6	7	8	9	10	11
Initial value	11	3	6	9	8	4	1	5	10	2	7

   (b) $K = 7$,

Register	1	2	3	4	5	6	7	8	9	10	11	12	13
Initial value	13	11	4	8	1	12	9	3	5	6	2	10	7

   (c) $K = 6$,

Register	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Initial value	13	2	4	12	8	11	14	9	5	6	10	3	7	1

2. You are given a sequence of integers. You are allowed to pick some contiguous segment of this sequence (that is, a segment without any gaps), and multiply the length of this segment by the minimum number in the segment. What is the maximum value that you can generate in this manner?

   For example, suppose the given sequence is $7, 2, 8, 10$. If you pick the entire sequence, the length is 4 and the minimum number is 2, so the product is 8. If you pick the segment $8, 10$, the length is 2 and the minimum number is 8, so the product is 16. This is the maximum value you can generate for this sequence. Note that you cannot pick $7, 8, 10$, since it is not a contiguous segment.

In each of the cases below, you are given a sequence of values and you have to determine the maximum value that you can generate by picking a contiguous segment and multiplying its length by the minimum number in the segment.

(a) 5, 14, 8, 7, 6, 10, 10, 4, 2, 5, 30

(b) 24, 22, 16, 18, 6, 7, 17, 16, 5, 20, 8, 19, 16

(c) 15, 30, 23, 1, 47, 23, 3, 8, 9, 10, 19, 21, 13, 5, 4

3. Your shop sells several different types of dolls. Each doll has a suggested price, and no two types of doll have the same price. You would like to fix an actual selling price for each doll so that dolls of different types are as different in price as possible. Due to some government regulations, you can only modify the suggested price within a fixed band of $\pm K$—in other words, if the suggested price is p, you can pick any selling price in the range $\{p-K, p-K+1, \ldots, p+K-1, p+K\}$. Of course, the selling price must always be non-negative.

   For instance, suppose there are four types of dolls with suggested prices 130, 210, 70 and 90 and you are allowed to modify prices within a band of 20. Then, you can adjust the prices to 150, 210, 50 and 100, respectively, so that the minimum difference in price between any two types of dolls is 50. (For the second doll, you could have picked any price between 200 and 230.) You can check that this is the largest separation that you can achieve given the constraint.

   In each of the cases below, you are given a sequence of prices and the value of K. You have to determine the maximum separation that you can achieve between all pairs in the sequence if you are allowed to modify each price by upto $\pm K$.

   (a) $K = 13$. Sequence: 144, 152, 214, 72, 256, 3, 39, 117, 238, 280.

   (b) $K = 10$. Sequence: 10, 48, 57, 32, 61, 74, 33, 45, 99, 81, 19, 24, 101.

   (c) $K = 20$. Sequence: 10, 19, 154, 67, 83, 39, 54, 110, 124, 99, 139, 170.

4. Your weekend is ruined because your parents have reminded you of a number of chores to be completed. You can only do one chore at a time and some of them depend on others: for instance, you can make your bed only after you pick up all the books on the bed and put them away. Rather than start on the chores, you begin to calculate the number of ways you could reorder them to get them done, while ensuring that you do not violate any dependencies.

   For example, suppose you have four tasks to complete—for convenience, we assume the tasks are numbered from 1 to 4. Suppose that task 4 depends on both task 2 and task

3, and task 2 depends on task 1. One possible sequence in which we can complete the given tasks is [3,1,2,4] — in this sequence, no task is attempted before any of the other tasks that it depends on. The sequence [3,2,1,4] would not be allowed because task 2 depends on task 1 but task 2 is scheduled before task 1. In this example, you can check that there exactly three possible sequences compatible with the dependencies: [3,1,2,4], [1,2,3,4] and [1,3,2,4].

In each of the cases below, you have N tasks numbered 1 to N with some dependencies between the tasks. You have to calculate the number of ways you can reorder all N tasks into a sequence that does not violate any dependencies.

(a)

Task Number	1	2	3	4	5	6	7	8	9	10
Depends on	-	1	2	1	4	3,5	6	7	6	8,9

(b)

Task Number	1	2	3	4	5	6	7	8	9	10	11	12	13
Depends on	-	1	2	-	4	3,5	6	7	6	9	8,10	11	11

(c)

Task Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Depends on	-	-	-	1,2,3	4	4	5,6	7	8	9	7	11	12	10,13	14

This page intentionally blank

Zonal Informatics Olympiad, 2013: *Answer sheet*

Name:	**Class:**	**Sex:**
School:		
Examination Centre:		
Father or Mother's Name:		
Full home address with PIN code:		
Home phone number, with STD Code:		
Email address:		

Write only your final answers in the space provided. Write all rough work on separate sheets.

Question 1	Minimum cost of sequence of moves to make K registers stable		
	(a)	(b)	(c)

Question 2	Maximum value generated among contiguous segments		
	(a)	(b)	(c)

Question 3	Maximum separation achievable among all pairs of prices		
	(a)	(b)	(c)

Question 4	Number of ways of reordering the tasks to be completed		
	(a)	(b)	(c)

For official use only. Do not write below this line.

1.	a	b	c		2.	a	b	c			
3.	a	b	c		4.	a	b	c			**Total**