BAPS-BUBT
National Programming Camp 2013

Selection Test

Name: ________________________________
Institution: ________________________________
Semester/Class: ________________________________
Phone: ________________________________
E-mail: ________________________________

Instructions:

1. The answers should be as **brief** as possible. If you are asked to find minimum time, just write down the minimum time, no need to explain how.

2. For the last part of all the problems (except 9) you are asked to write down code. You are encouraged to use C/C++/Java. The codes need not be pin-point perfect but it should be as clear as possible.

3. Marks for all the parts (except code part) are given in the corresponding questions.

4. For code part, the full marks is equal to the full marks of the method you described in your question. For example, for Question 1, suppose you solved your problem for the second constraint. Then full marks for your code will be 6.

5. For answering question you may divide the page into multiple columns. It will help you to complete your writing within given limited space.

6. We encourage you to answer in pencil. It will help you to erase them if you make mistakes.

Question 1. **N** Macaw birds are placed inside a **N*N** grid so that no two birds are in same cell. Now if two Macaw birds are in same row or same column then they start fighting. To avoid this fighting you have to place these **N** Macaw birds in the grid so that no two birds are in same row or same column.

a. If **N = 4**, how many ways are there to place the Macaw birds? **(Marks: 3)**

b. Find out a general formula for number of placements for **N** Macaw Birds.

Constraints: For **N<=8**: **(Marks: 4)**

For **N<=100,000**: **(Marks: 6)**

c. Write down code to solve (b).

Question 2. Suppose **N** Macaw birds are already placed inside **N*N** grid. There can be two Macaw Birds in same row or same column. So before it is too late, you want to move the birds so that there are no two birds in same row or same column. You want to do this as soon as possible. To move a bird from **(r1, c1)** to **(r2, c2)** it requires: **|r1-r2| + |c1-c2|** seconds. So move the Macaws as early as possible. Please note you must not move a Macaw through a cell that already contains another Macaw.

	1	2	3
1	M1	M2	
2		M3	
3			

For example, there are three Macaws in above diagram. Here we can solve within **3 seconds**. We can move **M2** right, then down twice. However, we could not do: **down, right, down** OR **down, down, right**. Because in such case **M2** would have gone through another Macaw **M3**. But, You could move **M3** to right and down, then move **M2** down. So this is another solution within **3 seconds** time.

a. Find out the minimum time to solve for following grid: **(Marks: 2)**

	M1		
M2			
	M3	M4	

b. Find out the minimum time to solve for following grid: **(Marks: 4)**

		M		M	M	M		
			M					M
	M					M		
				M				

c. Find out general solution.

Constraint: For **N <= 8:** **(Marks: 3)**

For **N <= 1,00,000** **(Marks: 6)**

d. Write down code to solve (c).

Question 3. There are **N** towers in a line. From the topmost floor of **i**th building one can see another person on the topmost floor of **j**th building if no tower between these two buildings are strictly higher than them. We call **(i, j)** a valid pair [Here **i** < **j**]. For example: suppose the height of the buildings are: 10, 5, 8, 11, 4. So we have six valid pairs: (1st, 2nd), (1st, 3rd), (1st, 4th), (2nd, 3rd), (3rd, 4th), (4th, 5th). [Buildings are considered **1-indexed**] 2nd Building and 4th Building can not be valid pair because the building between them is 8, which is larger than 2nd one.

a. If heights of the buildings are: 168, 92, 120, 147, 73, 160, 156; how many valid pairs exist? **(Marks: 2)**

b. Given height of N buildings. Give a general solution.

Constraint:	For **N <= 50:**	**(Marks: 3)**
	For **N <= 1,000:**	**(Marks: 4)**
	For **N <= 1,00,000:**	**(Marks: 5)**

c. Write code for (b).

Question 4. There are **N** mosquitoes and **M** bombs. Each bomb has some power. Say if the power of a bomb is 10, then it can kill 10 mosquitoes, if 5 then 5 mosquitoes, if 0 then no mosquito. You are given power of **M** bombs. Find out minimum number of bombs you need to use to kill all mosquitoes. If it is not possible to kill all, say IMPOSSIBLE.

For example, if there are 12 mosquitoes and 3 bombs of power: 2, 9, 4 then two bombs are sufficient. If we use 9 and 4 all 12 mosquitoes will die. But if we had used 9 and 2, then 1 mosquito would be alive.

Another example, if there are 10 mosquitoes and 3 bombs of: 0, 1, 5 then we can not kill all of them.

a. Find out the answer for: **N = 100**, **M = 10**. Powers are: 5, 22, 3, 37, 13, 29, 6, 24, 16, 11 (Marks: 3)

b. Given **N, M** and **M** powers. Find a general solution.

Constraint:	For **N, M <= 100:**	**(Marks: 4)**
	For **N, M <= 100,000:**	**(Marks: 6)**

c. Write code for (b).

5. Given a non-decreasingly sorted array. Find the rightmost occurrence of a given number X. Note that, this query will be given repeatedly.

a. Find the general solution:

Constraint: **Length of array <= 100:** **(Marks: 4)**

Length of array <= 1,000,000: **(Marks: 7)**

Note that, the numbers in the array will have absolute value <= 10^9.

b. Write code for (a).

Question 6. Suppose you want to unlock a lock. The key of the lock is a 4 digit number. Say, initially the lock shows “1234” and the key is “9876”. Now in each move you can change one of the digits of the number shown in the lock. But there are some forbidden numbers. If after any moves one of the forbidden number is displayed in the lock the lock will never unlock. Can you unlock the lock? If so give minimum number of moves.

For example, suppose: initially the lock displays: “1234” and the key is: “1243”. Forbidden numbers are: 1233 and 1244. One of the solutions is: 1234 -> 1254 -> 1253 -> 1243. Another solution can be: 1234 -> 2234 -> 2244 -> 2243. So you just need to output 3 because you needed 3 moves to unlock the lock.

Another example, say initally lock displays: “0000” and the key is “9999”. Forbidden numbers are: all except “0000” and “9999”. In this case it is impossible to solve.

a. Initial lock displays: “1234” and the key is: “5434”. Forbidden numbers are: “5234”, “1434”, “2234”, “3234”, “4234”, “5234”, “6234”, “7234”, “8234”, “9234”, “0234”, “1034”, “1134”, “1234”, “1334”, “1534”, “1634”, “1734”, “1834”, “1934”, “1254” and “1235”. Find out minimum number of moves. (Marks: 4)

b. Give a general solution. **(Marks: 8)**

c. Write code for (b).

Question 7. Given a convex polygon of **N** vertices. You have to select 3 vertices of convex polygon so that the area of the triangle formed by these three vertices is maximum.

a. Give general solution.

Constraint: **For N <= 100:** **(Marks: 4)**
For N <= 1,000: **(Marks: 6)**

b. Write code for (b)

Question 8. There are **N** lockers in a hallway numbered from **1** to **N**. Initially all the lockers are open. Now, **N** person goes through this hall way. **1st** person changes the status of all lockers. **2nd** person changes the status of all the lockers that are divisible by **2**. Similarly, **3rd** person changes the status of all the lockers that are divisible by **3**. To sum up, **i**th person changes the status of the lockers divisible by **i**. For each locker print number of times it is opened, and closed.

For example, if **N = 5**:

Locker 1 - Opened 0 - Closed 1
Locker 2 - Opened 1 - Closed 1
Locker 3 - Opened 1 - Closed 1
Locker 4 - Opened 2 - Closed 1
Locker 5 - Opened 1 - Closed 1

a. Find out the Open-Close status for N = 10. **(Marks: 3)**

b. Give general solution.

Constraint:	**For N <= 100:**	**(Marks: 4)**
	For N <= 100,000:	**(Marks: 6)**

c. Write code for (b)

Question 9. Given a bipartite graph. Let **M** be maximum matching of the graph. Count the number of edges **u-v** for which, if we match u with v, then the total number of matching (in any way) will be less than **M**.

For example, Say there are 3 nodes in left(1, 2 and 3) and 3 nodes in right (A, B and C). There are 5 edges. 1-A, 1-B, 2-B, 2-C and 3-C

```
1 o---o A
   \
    \
     \
2 o---o B
   \
    \
     \
3 o---o C
```

Here maximum matching is 3. (1-A, 2-B, 3-C). However, if we take 1-B in matching, then maximal matching can be 2 (1-B, 2-C or 3-C). So 1-B can never be part of a maximum matching. Similarly 2-C. So here there are 2 such edges. Answer = 2.

a. Find out the general solution.

Constraint:	For **V, E <= 100:**	**(Marks: 8)**
	For **V <= 1000, E <= 10,000**	**(Marks: 12)**

Here V = number of vertices, E = number of Edges

Question 10. Given two strings **S** and **T**. You can delete a character from **S** with cost **15** and a character from **T** with cost **30**. Your goal is to make the strings equal (same). It is not mandatory to delete a character.

For example: **S = AXB** and **T = YAB**. Now, if we delete X from **S** and Y from **T**, then total cost = 15 + 30 = **45**. And **S** and **T** will become **AB**.

Another example: **S = AB**, **T = CD**. Now total cost = 15 + 15 + 30 + 30 = **90**.

a. **S = MZJAWXU** and **T = XMJYAUZ**. Find out the minimum cost. **(Marks: 4)**

b. Find out the general solution.

Constraint: **Length of S or T <= 1,000.** **(Marks: 8)**

c. Write down code of (b).