



## ACM-ICPC Thailand Southern Programming Contest 2013

Hosted by  
Department of Computer Engineering  
Prince of Songkla University Hatyai Campus

10 August 2013

### Contest Problems

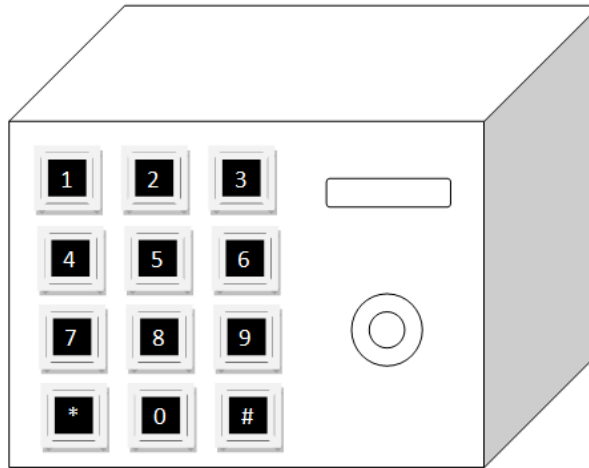
- There are **8** problems (A-H) to solve within 3 hours 30 minutes.
- Solve as many problems as you can, in an order of your choice.
- Use C or C++ or Java to program at your convenience for any problems.
- Input and output of each program are **standard input** and **output**.

Problem A	Unlock My Safe
Problem B	Two Mysterious Alphabets from a Tree
Problem C	Max Volume
Problem D	Birthday Statistics
Problem E	Nonogram
Problem F	Jane's First Words
Problem G	Range Sum Query
Problem H	Sum of Distinct Numbers ผลรวมเลขไม่ซ้ำ

## Problem A. Unlock My Safe

Time Limit: 2s

I forgot the password to my safe. There is a lot of money in it! Please help me unlock the safe. The keypad looks like this.



I do not remember how long my password is. Hence, you need to try a different length of the password. However, there are some hints that I can recall.

- I never use characters \*, #, 0 and 9 in my password.
- Each digit in the password is distinct. That is, they never appear more than once.
- My password is at most 8 digits ( $1 \leq N \leq 8$ , where  $N$  is a number of digits in the password).
- Each digit  $i$  in the password always has the value less than or equal to  $N$  (that is, a password 132 is valid for  $N = 3$  but a password such as 124 is invalid because the 3<sup>rd</sup> digit exceeds 3).

Use the information above and generate all possible permutations. One permutation corresponds to one guess of a password to unlock my safe. Importantly, the correct password is deliberately fixed at position  $L \backslash 3$  in the sorted array of permutations, where  $L$  is a number of all possible permutations and ' $\backslash$ ' is an *integer division*. The sorted array of permutations is in ascending order and the starting index in the sorted array begins at 0 (not 1).

Write a program to find a correct password for a given length (a number of digits in the password).

### Input

The first line of the input contains an integer  $T$  ( $1 \leq T \leq 6$ ) denoting the number of test cases. After that  $T$  test cases follow. Each test case contains an integer  $N$  ( $1 \leq N \leq 8$ ) denoting a number of digits in a password.

### Output

Your program should output the  $N$ -digit password for each corresponding test case, one password per line.

Sample input	Sample output
3	12
2	213
3	1
1	

### Explanation

There are 3 test cases above. In the second case, for example, the sorted permutations are {123, 132, 213, 231, 312, 321}. Password is located at the position  $6 \setminus 3 = 2$  (integer division). When the starting index begins at 0, the password is, therefore, 213.