

A 1, 10, 2, 3, 4, ...

时间限制: 1000ms 内存限制: 65536kb

通过率: 187/190 (98.42%) 正确率: 187/590 (31.69%)

## 题目描述

Python 是一门神奇的语言。

在 Python 中, `input()` 函数会读入输入中完整的一行, 并以字符串类型返回。

我们初学 Python 的可爱小萌新并不知道这一点。小萌新建立了一个有  $n$  行, 第  $i$  行是一个正整数  $i$  的文件, 并照着教程编写了一个 Python 程序——从输入中读入  $n$  个数字, 排序后输出这些数字的**最大值**。当然, 小萌新直接以 `input()` 读入, 并没有将这  $n$  个数字转换为整数类型, 所以对这  $n$  个数字排序是作为**字符串按照字典序**排序的。

初学 Python 的可爱小萌新自然也不知道时间复杂度的概念。小萌新所建立的文件行数  $n$  可能非常大, 程序执行结束遥遥无期。小萌新带着文件找到了你, 想让你用超级计算机求出程序的输出。你能帮帮他吗?

对于两个字符串  $S, T$ , 其字典序定义为:  $S < T$  当且仅当  $S$  为  $T$  的前缀, 或是存在  $k$  使得  $S_1S_2 \dots S_k = T_1T_2 \dots T_k$  且  $S_{k+1} < T_{k+1}$ 。例如, 字符串 `132`  $<$  `24`, 字符串 `23`  $<$  `230`。

你当然没有超级计算机, 但你有算法赋予你的力量。

题目大意: 给你一个数 $n$ , 找从1到 $n$ 这些数里, 看作字符串时, 字典序最大的那个

## 输入

本题测试点包含多组数据。

第一行, 一个正整数  $T$  ( $1 \leq T \leq 10^3$ ) , 表示数据组数。

对于每组数据:

一行, 一个正整数  $n$  ( $1 \leq n \leq 10^{10^3}$ ) , 表示小萌新的文件行数。

## 输出

对于每组数据:

输出一行, 一个整数, 表示答案。

## 输入样例

```
3
6
61
616
```

## 输出样例

```
6
9
99
```

### 提示

对于第一组数据, 1, 2, 3, ..., 6 排序后的结果为 1, 2, 3, 4, 5, 6。

对于第二组数据, 1, 2, 3, ..., 61 排序后的结果为 1, 10, 11, 12, ..., 2, 20, 21, ..., 3, 30, 31, ..., 4, 40, 41, ..., 5, 50, 51, ..., 6, 60, 61, 7, 8, 9, 因此排序后得到的最大值为 9。

61----9

616----99

1234---999

87087----9999

90----9不行, 需90

992----99不行, 需992

9008----999和9008? 选999

956----99和956? 选99

假设输入字符串s

长度为n

那么只需比较(n-1)个9

和字符串s谁大

谁大输出谁

特殊情况: 一位数

(特殊情况其实可以合并)

```
01  #include <bits/stdc++.h>
02  using namespace std;
03
04  int T;
05  char s[1002];
06  char ans[1002];
07  int main(){
08      ios::sync_with_stdio(false);
09      cin.tie(nullptr);
10      cin>>T;
11      while(T--){
12          cin>>s;
13          int len=strlen(s)-1;
14          for(int j=0;j<len;j++){
15              ans[j]='9';
16              if(strcmp(s,ans)<=0)
17                  cout<<ans<<endl;
18              else
19                  cout<<s<<endl;
20              memset(ans,0,sizeof(ans));
21          }
22      }
23      return 0;
```

Accepted

1

C++

380

5

3284

26 分钟 26 秒前

# Thank you!