


[TheKing003KS](#) | [Logout](#)

 You have [+416! Wow!](#)
[HOME](#) [TOP](#) [CONTESTS](#) [GYM](#) [PROBLEMSET](#) [GROUPS](#) [RATING](#) [EDU](#) [API](#) [CALENDAR](#) [HELP](#) [DELTIX ROUNDS 2021](#)
[PROBLEMS](#) [SUBMIT CODE](#) [MY SUBMISSIONS](#) [STATUS](#) [HACKS](#) [ROOM](#) [STANDINGS](#) [CUSTOM INVOCATION](#)

## General



#	Author	Problem	Lang	Verdict	Time	Memory	Sent	Judged	
124016190	Virtual: TheKing003KS	<a href="#">1465B</a> - 33	GNU C++14	Accepted	1450 ms	3800 KB	2021-07-28 12:39:01	2021-07-28 12:39:01	<a href="#">Compare</a>

[→ Source](#)[Copy](#)

```
#include<iostream>
#include<limits.h>
#include<math.h>
#include<vector>
#include<string>
#include<queue>
#include<stack>
#include<set>
#include<map>
#include<unordered_set>
#include<unordered_map>
#include<algorithm>
using namespace std;

#define ll long long int
#define ull unsigned long long int
#define modulo 1000000007
#define mp make_pair
#define pb push_back

bool check(multiset<int>& st, ull n)
{
    for(auto it = st.begin(); it != st.end(); it++)
    {
        if(n%(*it) != 0) {return false;}
    }
    return true;
}
```

```
void update(multiset<int>& st, ull& n)
{
    if(n%10 != 9)
    {
        int count = st.count(n%10);
        st.erase(n%10);
        if(count > 1)
        {
            --count;
            while(count > 0) {st.insert(n%10); count--;}
        }
        n++;
        st.insert(n%10);
    }
    else
    {
        ull temp = n;
        n++;

        int allcount = st.count(9);
        int count = 0;
        while(temp > 0 && temp%10 == 9)
        {
            count++;
            temp /= 10;
        }
        st.erase(9);
        if(temp == 0) {st.insert(1);}
        else
        {
            allcount -= count;
            while(allcount > 0)
            {
                st.insert(9);
                allcount--;
            }

            st.insert((temp%10)+1);
            allcount = st.count(temp%10)-1;
            st.erase(temp%10);
            while(allcount > 0)
            {
                st.insert(temp%10);
                allcount--;
            }
        }
    }
}

int main()
{
    ll tests;
```

```
cin >> tests;
while(tests--)
{
    ull n;
    cin >> n;
    multiset<int> st {};

    ull temp = n;
    while(temp > 0)
    {
        if(temp%10 != 0) {st.insert(temp%10);}
        temp /= 10;
    }

    bool flag = false;
    while(!flag)
    {
        if(check(st,n)) {flag = true; break;}
        update(st,n);
    }
    cout << n;

    cout << "\n";
}

return 0;
}

// LL n;
// cin >> n;
// vector<LL> arr(n);
// for(LL i = 0; i < n; i++) {cin >> arr[i];}

// for(LL i = 0; i < n; i++) {cout << arr[i] << " ";}
// sort(begin(arr),end(arr));
```

**1****Time:** 15 ms, **memory:** 3632 KB**Verdict:** OK**Input**

```
4
1
282
1234567890
10000000000000000000
```

**Participant's output**

```
1
288
1234568040
10000000000000000000
```

**Jury's answer**

```
1
288
1234568040
10000000000000000000
```

**Checker comment**

ok 4 number(s): "1 288 1234568040 10000000000000000000"

**2**

**Time:** 0 ms, **memory:** 3644 KB

**Verdict:** OK

**Input**

```
100
95
94
31
65
35
95
70
78
81
36
69
97
39
28
89
62
36
23
35
21
36
11
65
39
13
34
79
87
91
70
43
82
```

```
24
97
6
87
49
81
60
92
63
9
16
6
31
90
6
5
70
48
27
86
65
66
45
8
67
17
24
44
34
93
26
32
82
54
72
83
94
81
79
79
9
9
4
59
42
83
80
67
28
62
56
58
```

```
87
67
34
3
61
14
62
19
94
31
74
97
20
70
7
98
```

**Participant's output**

```
99
99
33
66
36
99
70
80
88
36
70
99
40
30
90
66
36
24
36
22
36
11
66
40
15
36
80
88
99
70
44
88
24
99
6
```

```
88
50
88
60
99
66
9
20
6
33
90
6
5
70
48
30
88
66
66
48
8
70
20
24
44
36
99
30
33
88
55
77
88
99
88
80
80
9
9
4
60
44
88
80
70
30
66
60
60
88
70
36
```

```
3  
66  
15  
66  
20  
99  
33  
77  
99  
20  
70  
7  
99
```

**Jury's answer**

```
99  
99  
33  
66  
36  
99  
70  
80  
88  
36  
70  
99  
40  
30  
90  
66  
36  
24  
36  
22  
36  
11  
66  
40  
15  
36  
80  
88  
99  
70  
44  
88  
24  
99  
6  
88  
50  
88
```

```
60
99
66
9
20
6
33
90
6
5
70
48
30
88
66
66
48
8
70
20
24
44
36
99
30
33
88
55
77
88
99
88
80
80
9
9
4
60
44
88
80
70
30
66
60
60
88
70
36
3
66
15
```

```
66  
20  
99  
33  
77  
99  
20  
70  
7  
99
```

**Checker comment**

ok 100 numbers

**3****Time:** 78 ms, **memory:** 3788 KB**Verdict:** OK**Input**

```
500  
140183543337950  
168603966874180  
78351473674564  
200536681738072  
77808397578505  
198657155536461  
64667077838113  
108830432159844  
209565500315551  
87474177825462  
86580595965887  
144490934818354  
195277503126334  
81302856621931  
72331495706596  
96854512117008  
169998132976209  
169696018293696  
195397880153878  
183161433320478  
116408689974678  
87474669378969  
105688335541377  
50139619003631  
37243552439852  
25270137623188  
13340436081126  
168125576660804  
211005144316370  
16173812965149  
1832922904...
```

**Participant's output**

```
140183543338080
168603966874224
78351473675160
200536681738680
77808397579920
198657155536560
64667077838160
108830432160000
209565500315940
87474177825720
86580595965960
144490934818368
195277503126510
81302856622080
72331495707240
96854512118040
169998132976560
169696018293696
195397880154120
183161433320496
116408689974864
87474669379272
105688335541440
50139619003800
37243552440360
25270137623220
13340436081144
168125576661000
211005144316440
16173812966472
183292290420624...
```

**Jury's answer**

```
140183543338080
168603966874224
78351473675160
200536681738680
77808397579920
198657155536560
64667077838160
108830432160000
209565500315940
87474177825720
86580595965960
144490934818368
195277503126510
81302856622080
72331495707240
96854512118040
169998132976560
169696018293696
195397880154120
```

```
183161433320496
116408689974864
87474669379272
105688335541440
50139619003800
37243552440360
25270137623220
13340436081144
168125576661000
211005144316440
16173812966472
183292290420624...
```

**Checker comment**

ok 500 numbers

**4**

**Time:** 358 ms, **memory:** 3776 KB

**Verdict: OK**

**Input**

```
1000
847043333344212536
255711414640467215
359585426230670768
901114206246591135
816368992977850831
614043958546617363
986187333189195992
169716380709625331
673684382341944346
728995382044081487
263528322955836327
770679321638129300
727806023459228375
870091250216349348
732327799524189381
850307540730080243
539948832852680157
666815481378279537
956219684847752988
939662552121624402
510471857531675919
591870557456397874
430371779479282132
286971322007582427
993026004339608467
30549...
```

**Participant's output**

```
847043333344212648
255711414640467240
```

```
359585426230672320
901114206246591300
816368992977852720
614043958546617480
986187333189197160
169716380709627360
673684382341944792
728995382044083000
263528322955836480
770679321638130072
727806023459229960
870091250216350680
732327799524191640
850307540730081000
539948832852680280
666815481378280080
956219684847754320
939662552121624540
510471857531676600
591870557456398080
430371779479282368
286971322007583360
993026004339608496
30549997358...
```

**Jury's answer**

```
847043333344212648
255711414640467240
359585426230672320
901114206246591300
816368992977852720
614043958546617480
986187333189197160
169716380709627360
673684382341944792
728995382044083000
263528322955836480
770679321638130072
727806023459229960
870091250216350680
732327799524191640
850307540730081000
539948832852680280
666815481378280080
956219684847754320
939662552121624540
510471857531676600
591870557456398080
430371779479282368
286971322007583360
993026004339608496
30549997358...
```

**Checker comment**

```
ok 1000 numbers
```

**5****Time:** 295 ms, **memory:** 3788 KB**Verdict:** OK**Input**

```
1000
624037064446665658
11551354003426892
344719137001384348
69940153225211284
539745875227813278
645852903622698659
374152487627950830
290167635117949847
568015937529312416
392385846934427640
505525651762264592
430979836740416887
252195672611250254
21156867317827548
152140808948981140
515121538287374980
611307840308581734
56510051049685818
342610535005610572
203486778843339243
37164151431864898
608671855667679978
210766843536131373
134521051551561523
749229626614494682
9449738414...
```

**Participant's output**

```
624037064446665660
11551354003427280
344719137001385160
69940153225211400
539745875227815480
645852903622698960
374152487627950920
290167635117950100
568015937529314400
392385846934428120
505525651762265190
430979836740417216
252195672611250270
21156867317828760
152140808948981400
```

```
515121538287375600
611307840308582760
56510051049686160
342610535005610640
203486778843339600
37164151431865080
608671855667680560
210766843536131400
134521051551561540
749229626614494816
9449738414003865...
```

**Jury's answer**

```
624037064446665660
11551354003427280
344719137001385160
69940153225211400
539745875227815480
645852903622698960
374152487627950920
290167635117950100
568015937529314400
392385846934428120
505525651762265190
430979836740417216
252195672611250270
21156867317828760
152140808948981400
515121538287375600
611307840308582760
56510051049686160
342610535005610640
203486778843339600
37164151431865080
608671855667680560
210766843536131400
134521051551561540
749229626614494816
9449738414003865...
```

**Checker comment**

ok 1000 numbers

**6**

**Time:** 296 ms, **memory:** 3800 KB

**Verdict:** OK

**Input**

```
1000
631622285725806863
187442345210855180
386143350611891104
```

```
247599208614837737
550803700396545707
31901532839071033
236979039371594960
893319205139158355
443672627051883972
228249824298677237
411672688625388609
630465763785057620
346223175956428606
216358969491794547
419911863342430816
345615480229773495
572991835058696623
259562585716948667
428367276170650764
942259857750813281
758859354253482978
87294868760992809
476471334105556892
833615340416254917
51616636026849110
85538797...
```

**Participant's output**

```
631622285725807320
187442345210855280
386143350611891400
247599208614839760
550803700396547640
31901532839071560
236979039371595570
893319205139158680
443672627051884440
228249824298677496
411672688625388960
630465763785058320
346223175956428680
216358969491796320
419911863342430824
345615480229773600
572991835058697600
259562585716950600
428367276170651160
942259857750814680
758859354253483560
87294868760993040
476471334105557280
833615340416255040
51616636026849360
85538797372718...
```

**Jury's answer**

```
631622285725807320
187442345210855280
386143350611891400
247599208614839760
550803700396547640
31901532839071560
236979039371595570
893319205139158680
443672627051884440
228249824298677496
411672688625388960
630465763785058320
346223175956428680
216358969491796320
419911863342430824
345615480229773600
572991835058697600
259562585716950600
428367276170651160
942259857750814680
758859354253483560
87294868760993040
476471334105557280
833615340416255040
51616636026849360
85538797372718...
```

**Checker comment**  
ok 1000 numbers

## 7

**Time:** 264 ms, **memory:** 3792 KB

**Verdict:** OK

### Input

```
1000
33421481968028398
95653271847299321
7611690845686283
156354741649171705
43696465709526348
36332373508937433
102453327790185621
1589987815775090
58246804084363849
103215679527146719
166048441199548275
44830755195272318
47355068131500373
228747751792865677
67866492032462256
```

```
48655913753676936
197951538014280862
57956577080517877
52228384353660411
39207405442986405
67636344578644238
110057949702174353
68620659817553044
8254587852331102
114181150634012143
211530230394118081
13724...
```

**Participant's output**

```
33421481968028400
95653271847301560
7611690845687040
156354741649171800
43696465709527260
36332373508938360
102453327790185960
1589987815776360
58246804084363920
103215679527146760
166048441199548560
44830755195274440
47355068131501200
228747751792867440
67866492032463096
48655913753679120
197951538014281800
57956577080519880
52228384353660480
39207405442988160
67636344578644800
110057949702174600
68620659817555440
8254587852331800
114181150634012160
211530230394118200
13724037292...
```

**Jury's answer**

```
33421481968028400
95653271847301560
7611690845687040
156354741649171800
43696465709527260
36332373508938360
102453327790185960
1589987815776360
58246804084363920
103215679527146760
166048441199548560
```

```
44830755195274440
47355068131501200
228747751792867440
67866492032463096
48655913753679120
197951538014281800
57956577080519880
52228384353660480
39207405442988160
67636344578644800
110057949702174600
68620659817555440
8254587852331800
114181150634012160
211530230394118200
13724037292...
```

**Checker comment**

ok 1000 numbers

**8**

**Time:** 15 ms, **memory:** 3636 KB

**Verdict:** OK

**Input**

```
1
239
```

**Participant's output**

```
240
```

**Jury's answer**

```
240
```

**Checker comment**

ok 1 number(s): "240"

**9**

**Time:** 15 ms, **memory:** 3784 KB

**Verdict:** OK

**Input**

```
1
9999999999999999
```

**Participant's output**

```
9999999999999999
```

**Jury's answer**

```
9999999999999999
```

**Checker comment**

ok 1 number(s): "9999999999999999"



1234567890361  
1234567890361  
1234567890361  
1234567890361  
1234567890... .

## **Participant's output**

## Jury's answer

## **Checker comment**

ok 1000 numbers

12

**Time:** 1404 ms, **memory:** 3640 KB

**Verdict: OK**

## Input

## **Participant's output**

1234567892880  
1...

## Jury's answer

1...

## **Checker commer**

13

**Time:** 0 ms, **memory:** 3632 KB

**Verdict: OK**

## Input

5  
54646664646677  
65455555442  
567643456561

565433456545  
12342345784556

## **Participant's output**

54646664646840  
65455555480  
567643456800  
565433456640  
12342345785280

## Jury's answer

54646664646840  
65455555480  
567643456800  
565433456640  
12342345785280

## **Checker comment**

ok 5 number(s): "54646664646840 65455555480 56...800 565433456640 12342345785280"

14

**Time:** 15 ms, **memory:** 3648 KB

**Verdict: OK**

## Input

## **Participant's output**

1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1

## Jury's answer

```
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1  
1
```

**Checker comment**

ok 12 numbers

**15**

**Time:** 15 ms, **memory:** 3648 KB

**Verdict: OK**

**Input**

```
1  
1000000002897001
```

**Participant's output**

```
1000000002898008
```

**Jury's answer**

```
1000000002898008
```

**Checker comment**

ok 1 number(s): "1000000002898008"

**16**

**Time:** 15 ms, **memory:** 3800 KB

**Verdict: OK**

**Input**

```
1  
295178723456255263
```

**Participant's output**

```
295178723456256360
```

**Jury's answer**

```
295178723456256360
```

**Checker comment**

ok 1 number(s): "295178723456256360"

**17**

**Time:** 0 ms, **memory:** 3636 KB

**Verdict:** OK

## Input

**1**  
123456780079

## **Participant's output**

123456780720

## Jury's answer

123456780720

## **Checker comment**

ok 1 number(s): "123456780720"

18

**Time:** 1045 ms, **memory:** 3792 KB

**Verdict: OK**

## Input

## **Participant's output**

123456789123459120  
123456789123459120  
123456789123459120  
123456789123459120

## Jury's answer

## **Checker comment**

ok 1000 numbers

**19****Time:** 0 ms, **memory:** 3768 KB**Verdict:** OK**Input**

```
1  
12345678912345678
```

**Participant's output**

```
12345678912346920
```

**Jury's answer**

```
12345678912346920
```

**Checker comment**

```
ok 1 number(s): "12345678912346920"
```

**20****Time:** 0 ms, **memory:** 3652 KB**Verdict:** OK**Input**

```
1  
987654321000
```

**Participant's output**

```
9876543211080
```

**Jury's answer**

```
9876543211080
```

**Checker comment**

```
ok 1 number(s): "9876543211080"
```

**21****Time:** 15 ms, **memory:** 3636 KB**Verdict:** OK**Input**

```
1  
1023456790441
```

**Participant's output**

```
1023456791700
```

**Jury's answer**

```
1023456791700
```

**Checker comment**

```
ok 1 number(s): "1023456791700"
```

**22**

**Time:** 15 ms, **memory:** 3644 KB**Verdict:** OK**Input**

1  
2317839669

**Participant's output**

2317840224

**Jury's answer**

2317840224

**Checker comment**

ok 1 number(s): "2317840224"

**23****Time:** 15 ms, **memory:** 3648 KB**Verdict:** OK**Input**

5  
16786871671811  
10  
20  
30  
51

**Participant's output**

16786871672088  
10  
20  
30  
55

**Jury's answer**

16786871672088  
10  
20  
30  
55

**Checker comment**

ok 5 number(s): "16786871672088 10 20 30 55"

**24****Time:** 1201 ms, **memory:** 3644 KB**Verdict:** OK**Input**

1000  
94570562000000522  
94570562000000522

## **Participant's output**

94570562000003040  
94570562000003040...

### Jury's answer

## **Checker comment**

ok 1000 numbers

25

**Time:** 1138 ms, **memory:** 3784 KB

**Verdict: OK**

## Input

## **Participant's output**

## Jury's answer

## **Checker comment**

ok 1000 numbers

26

**Time:** 1372 ms, **memory:** 3792 KB

## Verdict: OK

## Input



```
111111111157893840
111111111157893840
111111111157893840
11111111115...
```

**Checker comment**  
ok 1000 numbers

**27**

**Time:** 1247 ms, **memory:** 3792 KB

**Verdict:** OK

**Input**

```
1000
111111115789112041
111111115789114561
111111115789117081
111111115789119601
111111115789122121
111111115789124641
111111115789127161
111111115789129681
111111115789132201
111111115789134721
111111115789137241
111111115789139761
111111115789142281
111111115789144801
111111115789147321
111111115789149841
111111115789152361
111111115789154881
111111115789157401
111111115789159921
111111115789162441
111111115789164961
111111115789167481
111111115789170001
111111115789172521
11111...
```

**Participant's output**

```
111111115789114560
111111115789117080
111111115789119600
111111115789122120
111111115789124640
111111115789127160
111111115789129680
111111115789132200
111111115789134720
111111115789137240
```

```
111111115789139760
111111115789142280
111111115789144800
111111115789147320
111111115789149840
111111115789152360
111111115789154880
111111115789157400
111111115789159920
111111115789162440
111111115789164960
111111115789167480
111111115789170000
111111115789172520
111111115789175040
11111111578...
```

**Jury's answer**

```
111111115789114560
111111115789117080
111111115789119600
111111115789122120
111111115789124640
111111115789127160
111111115789129680
111111115789132200
111111115789134720
111111115789137240
111111115789139760
111111115789142280
111111115789144800
111111115789147320
111111115789149840
111111115789152360
111111115789154880
111111115789157400
111111115789159920
111111115789162440
111111115789164960
111111115789167480
111111115789170000
111111115789172520
111111115789175040
11111111578...
```

**Checker comment**

ok 1000 numbers

**28****Time:** 1294 ms, **memory:** 3788 KB**Verdict:** OK

## Input

## **Participant's output**

9875111111113120  
9875111111113120  
98751111111...

## Jury's answer

#### **Checker comment**

ok 1000 numbers

29

**Time:** 1372 ms, **memory:** 3796 KB

**Verdict: OK**

## Input

## **Participant's output**

## Jury's answer

## **Checker comment**

ok 1000 numbers

30

**Time:** 1060 ms, **memory:** 3796 KB

## Verdict: OK

## Input

59781111112219761  
59781...

## **Participant's output**

## Jury's answer

```
59781111112222280
59781111112222280
59781111112222280
5978111111...
```

**Checker comment**  
ok 1000 numbers

---

[Codeforces](#) (c) Copyright 2010-2021 Mike Mirzayanov  
The only programming contests Web 2.0 platform  
Server time: Aug/20/2021 12:00:19 (j2).  
Desktop version, switch to [mobile version](#),  
[Privacy Policy](#).

Supported by

