



## سوال A: 808A

اعداد lucky عبارتند از :

1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 30, 40....

بنابراین میتوان نتیجه گرفت هر عدد lucky به فرم  $a * 10^b$  است ( برای  $a$  های تک رقمی) است و عدد lucky بعد از هر عدد، به فرم  $(\text{first digit of the current} + 1) * 10^{(\text{number of digits of the current} - 1)}$  است، بنابراین داریم:

C++:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s;
    cin>>s;
    int n = s.size();
    long long a = stoll(s);
    long long p = pow(10, n-1);
    cout<< p - (a % p);
    return 0;
}
```

Python:

```
n=input()
i=len(n)-1
t=int(n[0])+1
print(10**i*t-int(n))
```



برای حل این سوال کافی است ما، ماکسیمم عددی را که  $k$  بر آن بخش پذیر است، پیدا کنیم.  
در این صورت آب دادن باغ، به اندازه  $k / \text{max divisor of } k \text{ in array}$  خواهد بود.

C++:

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n, k;
    cin >> n >> k;
    int best = INT_MIN;
    for (int i = 0; i < n; i++) {
        int a;
        cin >> a;
        if (!(k % a) && a > best)
            best = a;
    }
    cout << k / best;
    return 0;
}
```

Python:

```
n,k=map(int,input().split())
l=list(map(int,input().split()))
print(min([k//i for i in l if k%i==0]))
```



باید همه‌ی حالات ممکن را بررسی کنیم، و همه‌ی حالات ممکن به اندازه‌ی انتخاب 3 از 6 است.

C++:

```
#include <bits/stdc++.h>
using namespace std;
int main() {

    int a[6];
    int sum = 0;
    for(int i = 0; i < 6; i++){
        cin>>a[i];
        sum +=a[i];
    }
    bool find = false;
    for(int i = 0; i < 6 && !find; i++)
        for(int j = 0; j < 6 && !find; j++)
            for (int k = 0; k < 6 && !find; k++)
                if(i !=j && j != k && i !=k && a[i]+a[j]+a[k] == sum -(a[i]+a[j]+a[k]))
                    find = true;
    if(find) cout<<"YES";
    else cout<<"NO";

}
```

Python:

```
import itertools

a = map(int, input().split())

perms = list(itertools.permutations(a))

for i in perms:
    if sum(i[:3]) == sum(i[3:]):
        print("YES")
        exit()
print("NO")
```



برای حل این سوال اعداد داده شده را بررسی میکنیم و تعداد اعداد در حالت صعودی و یکسان و نزولی رو حساب میکنیم و بعد بررسی میکنیم که آیا این مجموع این سه عدد با تعداد کل اعداد برابر است یا خیر.

C++:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin >> n;
    int a[n];
    for (int i = 0; i < n; i++)
        cin >> a[i];
    int y = 0, d = 0, s = 0;
    int index = 0;
    while ((a[index] < a[index + 1]) && (index < n - 1))
    {
        y++;
        index++;
    }
    while ((a[index] == a[index + 1]) && (index < n - 1))
    {
        d++;
        index++;
    }
    while ((a[index] > a[index + 1]) && (index < n - 1))
    {
        s++;
        index++;
    }
    if ((y + d + s) == n - 1)
        cout << "YES";
    else
        cout << "NO";
}
```

Python:

```
n = int(input())
num = [int(x) for x in input().split()]
i=1
while(i<n and num[i]>num[i-1]):
    i+=1
while(i<n and num[i]==num[i-1]):
    i+=1
while(i<n and num[i]<num[i-1] and i<n):
    i+=1
print("YES" if i==n else "NO")
```



ایده ی دیگر این است که بازه ی صعودی از اول ارایه و بازه ی نزولی از آخر ان را رد کرده و بررسی کنیم مقادیری که وسط باقی میمانند مشابه باشند.

در این صورت داریم :

```
n=int(input())
l=list(map(int, input().split()))
a=0;b=n-1
while a<n-1 and l[a]<l[a+1]:a+=1
while b>0 and l[b]<l[b-1]: b-=1
if len(set(l[a:b+1]))==1: print("YES")
else: print("NO")
```



بهترین ترتیبی که یک رشته میتواند در آن باشد آن است که همه حروف آن به ترتیب حروف الفبا باشند.

پس بهترین کار ممکن با توجه به محدودیت تنها یک حرکت آن است که اگر حرفی در رشته اصلی هست که در رشته مرتب شده در جایگاهش حرف دیگری نشسته آن حرف را با دور ترین حرف بهتر در رشته اصلی عوض کنیم تا به بهترین ترتیب ممکن برسیم.

C++:

```
#include <bits/stdc++.h>
using namespace std;
void solve()
{
    string a, b, ah;
    cin >> a >> b;
    ah = a;

    sort(ah.begin(), ah.end());

    for (int i = 0; i < a.length(); ++i)
    {
        if (a[i] != ah[i])
        {
            int mi = i;
            for (int j = i; j < a.length(); ++j)
                if (a[j] <= a[mi])
                    mi = j;
            if (i != mi)
            {
                swap(a[i], a[mi]);
                break;
            }
        }
    }

    if (a < b)
        cout << a << endl;
    else
        cout << "---" << endl;
}

int main()
{
    int tc;
    cin >> tc;
    while (tc--)
        solve();
}
```



## Python:

```
for _ in [0]*int(input()):  
    s,c=input().split();t=sorted(s);i=0;l=len(s)  
    while i<l and s[i]==t[i]:i+=1  
    if i<l:j=s.rfind(t[i]);s=s[:i]+s[j]+s[i+1:j]+s[i]+s[j+1:]  
    print(('---',s)[s<c])
```

