سوال A: 271A

❖ صرفا به سال ورودی یکی یکی اضافه کرده و چک میکنیم آیا رقم‌های تکراری دارد یا خیر.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
//solotions are same, implementations are different
bool distict(int n){
    string s = to_string(n);
    for (int i = 0; i < s.size(); i++){
        for (int j = i+1; j < s.size(); j++){
            if (s[i] == s[j])
                return false;
        }
    }
    return true;
}

int main()
{
    // solotion one
    int y;
    cin>>y;
    for (int i = y+1; i < 10000; i++)
    {
        if(distict(i)){
            cout<<i<<endl;
            break;
        }
    }

    //solotion two
    y++;
    while (!distict(y)){y++;}
    cout<<y<<endl;
}
```

Python:

```python
n = int(input())
while True:
    n += 1
    c = set(str(n))
    if len(c) == len(str(n)):
        print(n)
        break
```

سوال B: 1285A

- در ابتدا بازه‌ای برای سمت چپ(min) و سمت راست(max) پیدا میکنیم و سپس max-min+1 را چاپ میکنیم.
- راه حل خیلی خیلی راحت تر؟
  چاپ کردن عدد خط اول ورودی + 1 😊

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

int main()
{
    //find the two number wich Zoma can be anywhere between those
    int n, max = 0, min = 0;
    string s;
    cin>>n>>s;
    for (int i = 0; i < s.size(); i++){
        if (s[i] == 'L')      min--; //number of L with negitive sign
        else if (s[i] == 'R') max++; //number of R with positive sign
    }
    cout<<max-min+1; //number of Ls plus number of Rs equals n

    //based on the solotion above, we can just print n+1
    //cout<<n+1;
}
```

Python:

```python
print(int(input()) + 1)
input()
```

سوال C: 1419D1

- ابتدا باید بررسی کنیم که در چه حالاتی میتوانیم x تکه یخ خرید.
- سپس آرایه دریافتی را سورت میکنیم.
- ماکسیمم تعداد یخی که میتوانیم بخریم برابر است با: (n-1)/2 چرا؟
- ...

Ice spheres

Python:

```python
n = int(input())
l = sorted(map(int, input().split()))
print((n-1)//2)
print(*[l[n-i//2-1] if i % 2 == 0 else l[i//2] for i in range(n)])
```

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

vector<int> reorder(int n, vector<int> cheap, vector<int> exp){
    //odd = cheap
    //even exp
    vector<int> ans;
    int cheap_i = 0, exp_i = 0;

    for (int i = 0; i < n; i++){
        if(i%2 == 0){
            ans.push_back(exp[exp_i]);
            exp_i++;
        }else{
            if (cheap_i < cheap.size()){
                ans.push_back(cheap[cheap_i]);
                cheap_i++;
            }else{
                ans.push_back(exp[exp_i]);
            }
        }
    }
    return ans;
}

int main()
{
    vector<int> a, cheap, exp, ans_v;
    int n, t, ans;
    cin>>n;
    for (int i = 0; i < n; i++){
        cin>>t;
        a.push_back(t);
    }
    sort(a.begin(), a.end());

    if (n%2 == 0){

        ans = (n/2)-1;
        for (int i = 0; i < (n/2)-1; i++){
            cheap.push_back(a[i]);
        }
        for (int i = (n/2)-1; i < n; i++){
            exp.push_back(a[i]);
        }

    }else{
```

```
        ans = n/2;
        for (int i = 0; i < n/2; i++){
            cheap.push_back(a[i]);
        }
        for (int i = n/2; i < n; i++){
            exp.push_back(a[i]);
        }

    }

    ans_v = reorder(n, cheap, exp);
    cout<<ans<<endl;
    for (int i = 0; i < n; i++){
        cout<<ans_v[i]<<" ";
    }


}
```

سوال D: 1326A

- اگر n=1 باشد، آنگاه هیچ جوابی وجود ندارد چون هر عدد بر خودش بخش پذیر است.
- اگر n>=2 باشد، آنگاه عدد (n رقم) 233...3 تمامی حالت های ذکر شده را شامل میشود چون نه بر 2 و نه بر 3 بخش پذیر است.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

int main()
{
    //we simply need to find a number with 2 digit with those condition
    //then we can repeat the digits n times
    //number 23 for example

    int t, n;
    cin>>t;
    while (t--){
        cin>>n;
        if (n > 1){
            cout<<2;
            for (int i = 0; i < n-1; i++)
                cout<<3;
            cout<<endl;
        }
        else cout<<-1<<endl;
    }


}
```

Python:

```python
for __ in range(int(input())):
    n = int(input())
    print(-1 if n == 1 else "2" + "3" * (n - 1))
```

سوال E: 1365B

- ابتدا اگر همه تایپ‌ها مثل هم باشند نمیتوانیم اعداد را سورت کنیم چون در هر حرکت تنها میتوانیم دو عنصر از تایپ‌های مختلف را جابه‌جا کنیم. در این حالت صرفا چک میکنیم آرایه از اول سورت شده بوده یا نه.
- اگر دست کم یک تایپ متفاوت داشته باشیم، آنگاه میتوان آرایه را سورت کرد.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;

int main()
{
    //we should check if we have different types
    //if we do, it means we can sort it somehow
    //if all types are same, we should check if array is already sorted
    int t, n;
    cin>>t;
    while (t--)
    {
        int b_sum = 0;
        cin>>n;
        int a[n],b[n];
        for (int i = 0; i < n; i++)
            cin>>a[i];
        for (int i = 0; i < n; i++){
            cin>>b[i];
            b_sum += b[i];
        }

        //check if all types are the same
        //all types are one => sum = n
        //all types are zero => sum = 0
        if (b_sum == n || b_sum == 0){

            if(is_sorted(a, a + n))
                cout<<"Yes"<<endl;
            else cout<<"No"<<endl;

        }
        else cout<<"Yes"<<endl;

    }

}
```

Python:

```python
for __ in range(int(input())):
    n = int(input())
    list_a = list(map(int, input().split()))
    list_b = list(map(int, input().split()))
    if (sum(list_b) == 0 or sum(list_b) == n):
        if (list_a == sorted(list_a)):
            print("Yes")
        else:
            print("No")
    else:
        print("Yes")
```