سوال A: 1209A

https://codeforces.com/problemset/problem/1209/A

کوچکترین عضو آرایه باید به یک طریق رنگ شود. بنابراین کوچکترین عدد و مضرب هایش در آرایه را یک رنگ می کنیم و اعداد رنگ شده را حذف میکنیم. پس تا آرایه خالی نشده است باید روند زیر را پیش بگیریم:

- پیدا کردن کوچکترین عدد
- رنگ جدیدی را استفاده کنیم و تمام مضرب های کوچکترین عدد را از آرایه حذف کنیم.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int n;
    cin>>n;
    int a[101]={0};
    int tt=n,ans=0;
    for (int i = 0,t; i < n; ++i){
        cin>>t;
        a[t]=1;
    }
    for (int i = 1; i < 101; ++i){
        if (a[i]){
            for (int j = i; j < 101; j+=i){
                a[j]=0;
            }
            ans++;
        }
    }
    cout<<ans<<endl;
}
```

Python:

```python
n=int(input())
ls=list(map(int,input().split()))
l=[]
c=0
cnt=0
while ls:
    j=min(ls)
    ls=[x for x in ls if x%j!=0]
    c+=1
print(c)
```

سوال B: 244A

https://codeforces.com/problemset/problem/244/A

به نفر i ام عدد a[i] را میدهیم. و دیگر نباید از این عدد استفاده کنیم. حالا روی اعداد 1 تا n*k حلقه میزنیم. اگر تا الان این عدد به شخصی تعلق ندارد، به شخص i ام میدهیم. این روند را برای شخص i ادامه میدهیم تا n عدد به آن شخص داده شود. سپس سراغ فرد i+1 میرویم. ...

C++:

```cpp
#include<bits/stdc++.h>
using namespace std;
int n,k,d,c,a[31],b[901];
main(){
    cin>>n>>k;
    for(int i=1;i<=k;i++){
        cin>>a[i];
        b[a[i]]=1;
    }
    for(int i=1;i<=k;i++){
    cout<<a[i]<<" ";
    d=n;c=1;
    while(d>1){
        if(b[c]==0){
            cout<<c<<" "<<endl;
            b[c]=1,d--;
        }
        c++;
    }
    }
}
```

Python:

```python
n,k=map(int,input().split())
a=[int(x) for x in input().split()]
j=1
for i in range(k):
    l=[a[i]]
    while len(l)!=n:
        if j not in a:
            l.append(j)
        j+=1
    print(*l)
```

سوال C: 1077B

https://codeforces.com/contest/1077/problem/B

اولین نکته این است که ما فقط به الگوهای نوع "101" علاقه مندیم. همه الگوهای دیگر اصلاً معنی ندارند.
بنابراین، یک رویکرد حریصانه پیش میگیریم. بیایید روی آرایه داده شده از چپ به راست تکرار کنیم و فرض کنیم که پیشوند پاسخ داده شده از قبل درست است. اگر اکنون در موقعیت i هستیم، ai – 1 = ai + 1 = 1 و ai = 0 (و برای پیشوند 1 تا i-2 پاسخ درست را داریم) پس کدام ۱ را باید جایگزین کنیم؟ وقتی سمت چپ را جایگزین می کنیم، در آینده نمی توانیم بهتر عمل کنیم، اما وقتی سمت راست را جایگزین می کنیم، می توانیم به جواب بهتری برسیم.

C++:

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

int main(){
    ll n,ans=0;
    cin >> n;
    std::vector<ll> flats(n);
    for(ll i=0;i<n;i++)cin >> flats[i];
    for(ll i=1;i<n-1;i++){
        if(flats[i]==0&&flats[i-1]==1&&flats[i+1]==1){
            flats[i+1] = 0;
            ans++;
        }
    }
    cout<<ans<<endl;
    return 0;
}
```

Python:

```python
input()
print(input().count('1 0 1'))
```

how dose it feel to see the python version? :)

سوال D: 483A

https://codeforces.com/problemset/problem/483/A

ساده ترین راه حل این است که 3 عدد متوالی را چاپ کنیم که عدد اول و آخر زوج هستند. زیرا هر دو عدد متوالی نسبت به هم اول اند و عدد اول و آخر هم زوج هستند پس نسبت به هم اول نیستند.
در صورتی که طول بازه کوچک باشد، جوابی نخواهیم داشت.

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long int lli;
int main(){
    lli l, r;
    cin>>l>>r;
    if(l%2 !=0) l++;
    if(l+2>r) cout<<-1;
    else{
        cout<<l<<" "<<l+1<<" "<<l+2;
    }
}
```

Python:

```python
l,r=map(int,input().split())
if l%2: l+=1
if l+2>r: print(-1)
else: print(l,l+1,l+2)
```

سوال E: 356A

https://codeforces.com/contest/356/problem/A

اجازه دهید مبارزه فعلی (l, r, x) از مبارزه K شوالیه تشکیل شود. سپس تنها کاری که باید انجام دهیم این است که همه این شوالیه ها را در زمان O(K) یا O(KlogN) پیدا کنیم. چندین راه برای انجام این کار وجود دارد.
یکی از آن ها استفاده از ساختمان داده set در cpp است.(tree set در جاوا)

C++:

```cpp
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const ll mx = 1e6;
ll ans[mx];
set<ll> w;

int main(){
    ll n,m;
    cin >> n >> m;
    for(ll i=0;i<n;i++)w.insert(i);
    while(m--){
        ll l,r,x;
        cin >> l >> r >> x;
        l--;
        for(auto it=w.lower_bound(l);it!=w.end();it++){
            if(*it>=r) break;
            if(*it!=x-1) ans[*it]=x;
        }
        w.erase(w.lower_bound(l),w.lower_bound(r));
        w.insert(--x);

    }
    for(ll i=0;i<n;i++){
        cout<<ans[i]<<' ';
    }
    cout<<endl;
    return 0;
}
```

Python:

```python
n, m = map(int, input().split())
p, d = [0] * (n + 2), list(range(1, n + 3))
for i in range(m):
    l, r, x = map(int, input().split())
    while l < x:
        if p[l]:
            k = d[l]
            d[l] = x
            l = k
        else:
            d[l], p[l] = x, x
            l += 1
    l += 1
    r += 1
    while p[r]: r = d[r]
    while l < r:
        if p[l]:
            k = d[l]
            d[l] = r
            l = k
        else:
            d[l], p[l] = r, x
            l += 1
print(' '.join(map(str, p[1: -1])))
```