



## سوال A: 1512A

❖ باید اندیس عددی که فقط یکبار تکرار شده است را پیدا کنیم و برای اینکار میتوان از تابع count استفاده کرد.

C++:

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

Python:

```
for _ in range(int(input())):
    n=int(input())
    l=list(map(int,input().split()))
    for i in range(n):
        if l.count(l[i])==1:
            print(i+1)
            break
```



## سوال B: 1311A

- ❖ اگر  $b > a$  باشد، باید  $a$  را با  $(b-a)$  جمع کنیم که اگر  $b-a$  فرد باشد در یک مرحله و اگر زوج باشد در دو مرحله صورت میگیرد.
- ❖ اگر  $a > b$  باشد، باید مقدار  $(a-b)$  را از  $a$  کم کنیم که اگر  $a-b$  زوج باشد در یک مرحله و اگر فرد باشد در دو مرحله صورت میگیرد.
- ❖ اگر  $a = b$  باشد، جواب صفر است.

C++:

```
#include<bits/stdc++.h>
using namespace std;
int main(){
    int t;
    cin>>t;
    while (t--){
        int a,b;
        cin>>a>>b;
        if(b>a){
            if((b-a)%2)
                cout<<1<<endl;
            else
                cout<<2<<endl;
        }
        else if(a>b){
            if(!((a-b)%2))
                cout<<1<<endl;
            else
                cout<<2<<endl;
        }
        else
            cout<<0<<endl;
    }
    return 0;
}
```

Python:

```
for _ in range(int(input())):
    a,b=map(int,input().split())
    print(0 if a == b else 1 + ((a+b)%2^(a<b)))
```



- ❖ خانه های ماتریس را بر اساس فاصله ی آنها از نقطه ی شروع، گروه بندی میکنیم. در گروه صفر، خانه ی (۱،۱) است؛ گروه یک شامل (۱،۲)، (۲،۱) است و به همین ترتیب  $n+m-1$  گروه خواهیم داشت.
- ❖ حال گروه های  $k$  و  $n+m-2-k$  را در نظر بگیرید. اگر  $k = 0$  یا  $k = n+m-2$ ، نقاط شروع و پایان را داریم که باید مقادیر آنها برابر باشند؛ در غیر این صورت، فرض کنید دو سلول  $(x,y)$  و  $(x+1,y-1)$  به یک گروه تعلق دارند. ما به راحتی و با در نظر گرفتن مسیر های مختلف، می توانیم اثبات کنیم که محتوای این دو سلول باید برابر باشند و با استفاده از استقرا، میتوان ثابت کرد که محتوای تمام خانه های موجود در یک گروه باید برابر باشند و از طرفی چون ماتریس باید palindromic باشد پس باید اعضای گروه  $k$  با گروه  $n+m-2-k$  برابر باشند و با این حساب، باید حداقل تغییرات را حساب کنیم تا همه ی خانه های گروه های  $k$  و  $n+m-2-k$  را مشابه کنیم.
- ❖ توجه داشته باشید که اگر  $n+m$  فرد باشد، گروه وسطی، گروهی برای اینکه با آن برابر باشد ندارد و نیازی به تغییر در آن نیست.

Python:

```

for o in range(int(input())):
    a,b=map(int,input().split())
    s=[list(map(int,input().split())) for i in range(a)]
    d=a+b-1
    S=[[ ] for i in range(d)]
    res=0
    for i in range(a):
        for j in range(b):
            S[i+j].append(s[i][j])
    for i in range(d//2):
        v=S[i]+S[-1-i]
        res+=min(v.count(0),v.count(1))
    print(res)

```



C++:

```
#include <bits/stdc++.h>
using namespace std;
void solve()
{
    int n, m;
    cin >> n >> m;
    vector<vector<int>> > a(n, vector<int>(m));
    for(int i = 0; i < n; i++)
        for(int j = 0; j < m; j++)
            cin >> a[i][j];
    vector<vector<int>> > cnt(n + m - 1, vector<int>(2));
    for(int i = 0; i < n; i++)
        for(int j = 0; j < m; j++)
            cnt[i + j][a[i][j]]++;
    int ans = 0;
    for(int i = 0; i <= n + m - 2; i++)
    {
        int j = n + m - 2 - i;
        if(i <= j) continue;
        ans += min(cnt[i][0] + cnt[j][0], cnt[i][1] + cnt[j][1]);
    }
    cout << ans << endl;
}

int main() {
    int t;
    cin >> t;
    for(int i = 0; i < t; i++)
        solve();
}
```



- ❖ هر دو عنصر متوالی که مانند هم نباشند، یک عدد به تعداد سگمنت ها اضافه میکنند. استراتژی ما برای رنگ آمیزی اعداد باید به گونه ای باشد که این سگمنت ها بیشینه شوند.
- ❖ برای این کار آرایه را از ابتدا تا انتها پیمایش میکنیم و استراتژی زیر را دنبال میکنیم:
  - فرض کنید عنصری که هم اکنون باید درباره رنگ آن تصمیم بگیریم  $z$ ، آخرین عنصر رنگ شده با سیاه  $x$  و آخرین عنصر رنگ شده با سفید  $y$  باشد. (اگر هیچ عنصری با سیاه رنگ نشده باشد  $x=0$  و همینطور اگر عنصری با سفید رنگ نشده باشد  $y=0$  خواهد بود).
  - حال به این شکل عمل میکنیم:
    - اگر  $z$  با  $x$  یکی بود، آن را به رنگ  $y$  واگر با  $y$  یکی بود، آن را به رنگ  $x$  در می آوریم (چنانچه با هردو یکی بود، دلخواه عمل میکنیم).
    - اگر  $z$  با هردوی  $x, y$  متفاوت بود،  $z$  را به رنگ آن عنصری در می آوریم که در ادامه پیمایش آرایه، زود تر به مقدار مشابه آن میرسیم.
- به عنوان مثال اگر  $x=3$  و  $y=4$  و  $z=5$  بود و دو خانه بعد از  $z$ ، دوباره عدد سه داشتیم و همینطور ۶ خانه بعد از  $y$  هم عدد ۴ داشتیم، با توجه به اینکه زود تر به مقدار 3 میرسیم،  $z$  را به رنگ  $x$  در می آوریم. (سعی کنید درست بودن استراتژی را اثبات کنید، یا اگر سعی کردید و نشد، در توتوریال اصلی سوال، اثبات را ببینید و با آن آشنا شوید).
- ❖ اشتباه متداول:
 

ممکن است شما در استراتژی تان هنگامی که  $z$  با هردوی  $x, y$  متفاوت است، تفاوتی میان آن ها نذارید و  $z$  را به طور دلخواه رنگ کنید. مثال زیر زیر نشان میدهد چرا این کار درست نیست:

1 2 2 1 2 1 3 2 2
- ❖ اگر هردو استراتژی را روی دنباله فوق اجرا کنید و نزدیک ترین مقدار مشابه را در نظر نگیرید ، ممکن است به عدد ۸ برای تعداد سگمنت ها برسید اما با استفاده از استراتژی گفته شده، به عدد ۹ برای تعداد سگمنت ها خواهید رسید.



C++:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long ll;
const ll MX = 1e5+10;
ll a[MX],nxt[MX],pos[MX];
int main(){
    ll n,x=0,y=0,ans=0;
    cin >> n;
    for(ll i=1;i<=n;i++)cin >> a[i];
    for(ll i=0;i<=n;i++)pos[i]=n+1;
    for(ll i=n; i>=0;i--){
        nxt[i] = pos[a[i]];
        pos[a[i]] = i;
    }
    for(ll i=1;i<=n;i++){
        if(a[x]==a[i]){
            ans+=(a[y]!=a[i]);
            y=i;
        }
        else if(a[y]==a[i]){
            ans+=(a[x]!=a[i]);
            x=i;
        }
        else if(nxt[x]<nxt[y]){
            ans+=(a[x]!=a[i]);
            x=i;
        }
        else{
            ans+=(a[y]!=a[i]);
            y=i;
        }
    }
    cout<<ans<<endl;
    return 0;
}
```



## Python:

```
n, x, y, ans = int(input()), 0, 0, 0
a, pos, nxt = [-1 for __ in range(n + 1)], [n + 1 for __ in range(n + 1)], [-1 for __ in range(n + 1)]
a[1:] = map(int, input().split())

for i in range(n, -1, -1):
    nxt[i], pos[a[i]] = pos[a[i]], i

for i in range(1, n + 1):
    if a[x] == a[i]:
        ans += (a[y] != a[i])
        y = i
    elif (a[y] == a[i]) or (nxt[x] < nxt[y]):
        ans += (a[x] != a[i])
        x = i
    else:
        ans += (a[y] != a[i])
        y = i
print(ans)
```

