



سوال A: 758A

❖ برای حل این سوال، کافی است بیشترین عدد را بین همه ی اعداد پیدا کنیم و انگاه مجموع اختلاف این عدد با هریک از اعداد دیگر حداقل مقداری است که برای برابر کردن همه ی آنها با هم، نیاز داریم.

C++:

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

    return 0 ;
}
```

Python:

```
n=int(input())
a=list(map(int,input().split()))
print(n*max(a)-sum(a))
```



❖ برای حل این سوال، با توجه با سایز ورودی، کافی است عدد داده شده در ورودی را به اعداد Lucky قبل از آن تقسیم کنید، اگر بخش پذیر بود YES و در غیر این صورت NO را چاپ کنید.

C++:

```
#include <bits/stdc++.h>
using namespace std;
int main ( )
{
    int lucky_numbers[14] = {4,7,47,74,44,77,474,477,744,747,447,444,774,777};
    int n;
    cin>>n;
    bool is_divisible = false;
    for (int i = 0; (i< 14 && !is_divisible); i++)
    {
        if(!(n % lucky_numbers[i]))
        {
            is_divisible =true;
        }
    }
    if(is_divisible) cout<<"YES";
    else cout<<"NO";

    return 0 ;
}
```

Python:

```
n=int(input())
print(['YES', 'NO'][all(n%i for i in [4,7,47,74,44,77,474,477,744,747,447,444,774,777])])
```



❖ در معادله ی داده شده در این سوال مقدار x را داریم. برای به دست آوردن مقدار a و b روی یکی از آنها پیمایش میکنیم و بر اساس آن، مقدار دومی را به دست می آوریم. اگر این مقدار عدد صحیح بود، به جواب رسیده ایم.

Python:

```
for t in range(int(input())):
    x = int(input())
    a = 1
    flag = 1
    while(a*a*a < x):
        b = (x - (a*a*a))**(1./3)
        if b - int(b) == 0.0:
            print("YES")
            flag = 0
            break
        a += 1
    if flag:
        print("NO")
```

C++:

```
#include <bits/stdc++.h>
using namespace std;
typedef long long int lli;
//brute force
void solve(lli x){
    for (lli a = 1; a*a*a < x; a++){ //a^3 >= x
        lli a3=(a*a*a);
        lli b=cbrt(x-a3);
        if((b * b * b )== x-a3){
            cout<<"YES"<<endl;
            return;
        }
    }
    cout<<"NO"<<endl;
}
int main(){
    int t;
    lli x;
    cin>>t;
    while(t--){
        cin>>x;
        solve(x);
    }
}
```



- ❖ ایده ی حل این سوال مانند ایده ی سوال آخر کانتست شماره ۸ است. در آن سوال باید آرایه را طوری رنگ میکردیم که بیشترین تعداد سگمنت را داشته باشیم. هر دو عنصر متوالی که مانند هم نباشند، یک عدد به تعداد سگمنت ها اضافه میکنند. استراتژی ما برای رنگ آمیزی اعداد به گونه ای بود که این سگمنت ها بیشینه شوند. در این سوال استراتژی برعکس است و باید کمینه شود.
- ❖ 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 x واگر با y یکی بود، آن را به رنگ y 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، دیرتر به مقدار مشابه آن میرسیم.

Python:

```
n, x, y, res = 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 z in range(1, n+1):
    if a[x] == a[z]:
        x = z
    elif a[y] == a[z]:
        y = z
    elif nxt[x] > nxt[y]:
        res += 1
        x = z
    else:
        res += 1
        y = z
print(res)
```



C++:

```
#include <bits/stdc++.h>
using namespace std;
typedef Long Long int lli;

const int MAXN = 100010;
int n;
int a[MAXN];
int pos[MAXN], nxt[MAXN];

void solve()
{
    cin>>n;
    for (int i = 1; i <= n; ++i) cin>>a[i];
    for (int i = 0; i <= n; ++i) pos[i] = n + 1;
    for (int i = n; i >= 0; --i)
    {
        nxt[i] = pos[a[i]];
        pos[a[i]] = i;
    }
    int x = 0, y = 0; // the last elements of the two subarrays
    int res = 0;
    for (int z = 1; z <= n; ++z)
    {
        // if one of the two last elements matches a[z], just append a[z] after it.
        if (a[x] == a[z]) x = z;
        else if (a[y] == a[z]) y = z;
        // otherwise, replace the later one.
        else if (nxt[x] > nxt[y])
        {
            res += 1;
            x = z;
        }
        else
        {
            res += 1;
            y = z;
        }
    }
    cout<<res<<endl;
}

int main()
{
    solve();
    return 0;
}
```

