PHASE 1 – BUET IUPC CORE (Math & Number Theory + Implementation)

Math & Number Theory

1️⃣ CF 1372B – Omkar and Last Class of Math Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    ios::sync_with_stdio(false);
    cin.tie(NULL);
    int t; cin >> t;
    while(t--) {
        long long n; cin >> n;
        if(n % 2 == 0) cout << n/2 << " " << n/2 << "\n";
        else {
            bool ok=false;
            for(long long i=3; i*i<=n; i+=2) {
                if(n%i==0) {
                    cout << n/i << " " << n - n/i << "\n";
                    ok=true;
                    break;
                }
            }
            if(!ok) cout << 1 << " " << n-1 << "\n";
        }
    }
}
```

2️⃣ CF 230B – T-primes Link

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

bool isPrime(long long x) {
    if(x < 2) return false;
    for(long long i=2; i*i<=x; i++)
        if(x % i == 0) return false;
```

```
    return true;
}

int main() {
    int n; cin >> n;
    while(n--) {
        long long x; cin >> x;
        long long r = sqrt(x);
        if(r*r == x && isPrime(r)) cout << "YES\n";
        else cout << "NO\n";
    }
}
```

3️⃣LightOJ 1138 – Trailing Zeroes (III) Link

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

long long trailingZero(long long n) {
    long long cnt=0;
    for(long long i=5; i<=n; i*=5) cnt += n/i;
    return cnt;
}

int main() {
    int t; cin >> t;
    for(int cs=1; cs<=t; cs++) {
        long long q; cin >> q;
        long long l=1, r=1e18, ans=-1;
        while(l<=r){
            long long mid=(l+r)/2;
            if(trailingZero(mid)>=q) ans=mid, r=mid-1;
            else l=mid+1;
        }
        cout << "Case " << cs << ": ";
        if(ans!=-1 && trailingZero(ans)==q) cout << ans << "\n";
        else cout << "impossible\n";
    }
}
```

4️⃣LightOJ 1098 – A New Function Link

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

```
    int t; cin >> t;
    for(int cs=1; cs<=t; cs++) {
        long long n; cin >> n;
        long long ans=0;
        for(long long i=1; i*i<=n; i++) {
            long long x=n/i;
            long long y=n/(i+1);
            ans += (x+y+1)*(x-y)/2 * i;
        }
        cout << "Case " << cs << ": " << ans << "\n";
    }
}
```

5️⃣ UVA 543 – Goldbach's Conjecture Link

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

int main() {
    const int N=1000000;
    vector<bool> prime(N,true);
    prime[0]=prime[1]=false;
    for(int i=2;i*i<N;i++) if(prime[i]) for(int j=i*i;j<N;j+=i) prime[j]=false;

    int n;
    while(cin >> n && n) {
        for(int i=2;i<n;i++) {
            if(prime[i] && prime[n-i]) {
                cout << n << " = " << i << " + " << n-i << "\n";
                break;
            }
        }
    }
}
```

6️⃣ UVA 11466 – Largest Prime Divisor Link

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

int main() {
    long long n;
    while(cin >> n && n) {
        n = llabs(n);
        long long mx=-1;
```

```
        for(long long i=2;i*i<=n;i++) {
            if(n%i==0) { mx=i; while(n%i==0) n/=i; }
        }
        if(n>1) mx=max(mx,n);
        cout << (mx==-1?-1:mx) << "\n";
    }
}
```

Implementation

1️⃣ CF 158A – Next Round [Link]

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

2️⃣ CF 339A – Helpful Maths [Link]

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s; cin >> s;
    vector<char> v;
    for(char c:s) if(c!='+') v.push_back(c);
    sort(v.begin(),v.end());
    for(int i=0;i<v.size();i++) {
        if(i) cout << "+";
        cout << v[i];
    }
}
```

3️⃣ CF 110A – Nearly Lucky Number [Link]

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    string s; cin >> s;
    int c=0;
```

```
    for(char x:s) if(x=='4'||x=='7') c++;
    cout << (c==4||c==7?"YES":"NO");
}
```

4️⃣UVA 11764 – Jumping Mario Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int t; cin >> t;
    for(int cs=1; cs<=t; cs++) {
        int n; cin >> n;
        int up=0, down=0, prev, cur;
        cin >> prev;
        for(int i=1;i<n;i++) {
            cin >> cur;
            if(cur>prev) up++;
            if(cur<prev) down++;
            prev=cur;
        }
        cout << "Case " << cs << ": " << up << " " << down << "\n";
    }
}
```

PHASE 2 – Binary Search + Dynamic Programming

Binary Search

1️⃣CF 474B – Worms Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n; cin >> n;
    vector<int> a(n), pref(n);
    for(int i=0;i<n;i++) { cin >> a[i]; pref[i]=a[i]+(i?pref[i-1]:0); }
    int q; cin >> q;
    while(q--) {
        int x; cin >> x;
        cout << lower_bound(pref.begin(),pref.end(),x)-pref.begin()+1 << "\n";
    }
}
```

2️⃣CF 1201C – Maximum Median Link

```cpp
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n; long long k; cin >> n >> k;
    vector<long long> a(n);
    for(auto &x:a) cin >> x;
    sort(a.begin(),a.end());
    int mid=n/2;
    for(int i=mid+1;i<n;i++){
        long long need=(a[i]-a[mid])*(i-mid);
        if(need<=k){ k-=need; a[mid]=a[i]; }
        else { a[mid]+=k/(i-mid); break; }
    }
    cout << a[mid];
}
```

3️⃣LightOJ 1048 – Conquering Keokradong Link

```cpp
#include <bits/stdc++.h>
using namespace std;
int n,k;
vector<int> a;
bool ok(long long m) {
    long long cur=0; int cnt=1;
    for(int x:a) {
        if(x>m) return false;
        if(cur+x<=m) cur+=x;
        else cur=x,cnt++;
    }
    return cnt<=k+1;
}
int main() {
    int t; cin >> t;
    for(int cs=1; cs<=t; cs++) {
        cin >> n >> k;
        a.resize(n+1);
        for(int i=0;i<=n;i++) cin >> a[i];
        long long l=0,r=1e18,ans;
        while(l<=r){
            long long m=(l+r)/2;
            if(ok(m)) ans=m, r=m-1;
            else l=m+1;
        }
        cout << "Case " << cs << ": " << ans << "\n";
```

```
    }
}
```

Dynamic Programming (DP)

1️⃣ CF 455A – Boredom Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n; cin >> n;
    map<int,long long> cnt;
    for(int i=0;i<n;i++){ int x; cin >> x; cnt[x]++; }
    vector<long long> dp(100005,0);
    for(auto [x,c]:cnt) dp[x]=max(dp[x-1],dp[x-2]+c*x);
    cout << dp.rbegin()->second;
}
```

2️⃣ CF 580A – Kefa and First Steps Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n; cin >> n;
    vector<int> a(n); for(auto &x:a) cin >> x;
    int ans=1,cur=1;
    for(int i=1;i<n;i++){
        if(a[i]>=a[i-1]) cur++;
        else cur=1;
        ans=max(ans,cur);
    }
    cout << ans;
}
```

3️⃣ CF 189A – Cut Ribbon Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n,a,b,c; cin >> n >> a >> b >> c;
    vector<int> dp(n+1,-1);
    dp[0]=0;
    for(int i=1;i<=n;i++){
        if(i>=a && dp[i-a]!=-1) dp[i]=max(dp[i],dp[i-a]+1);
```

```
        if(i>=b && dp[i-b]!=-1) dp[i]=max(dp[i],dp[i-b]+1);
        if(i>=c && dp[i-c]!=-1) dp[i]=max(dp[i],dp[i-c]+1);
    }
    cout << dp[n];
}
```

4️⃣ LightOJ 1013 – Love Calculator Link

```
#include <bits/stdc++.h>
using namespace std;
string a,b;
long long dp[35][35];
long long solve(int i,int j){
    if(i==a.size() || j==b.size()) return 1;
    long long &res=dp[i][j];
    if(res!=-1) return res;
    res=0;
    if(a[i]==b[j]) res+=solve(i+1,j+1);
    else res+=solve(i+1,j)+solve(i,j+1);
    return res;
}
int main() {
    int t; cin >> t;
    for(int cs=1; cs<=t; cs++){
        cin >> a >> b;
        memset(dp,-1,sizeof(dp));
        cout << "Case " << cs << ": " << solve(0,0) << "\n";
    }
}
```

PHASE 3 – Graph Theory

1️⃣ CF 520B – Two Buttons Link

```
#include <bits/stdc++.h>
using namespace std;
int main() {
    int n,m; cin >> n >> m;
    vector<int> dist(100005,-1);
    queue<int> q; q.push(n); dist[n]=0;
    while(!q.empty()){
        int u=q.front(); q.pop();
        if(u*2<100005 && dist[u*2]==-1) dist[u*2]=dist[u]+1,q.push(u*2);
```

```
        if(u-1>0 && dist[u-1]==-1) dist[u-1]=dist[u]+1,q.push(u-1);
    }
    cout << dist[m];
}
```

2️⃣ CF 862B – Mahmoud and Ehab and the Bipartiteness [Link]

```
#include <bits/stdc++.h>
using namespace std;
vector<int> g[200005];
int color[200005];
bool dfs(int u,int c){
    color[u]=c;
    for(int v:g[u]){
        if(color[v]==c) return false;
        if(color[v]==0 && !dfs(v,3-c)) return false;
    }
    return true;
}
int main(){
    int n,m; cin >> n >> m;
    while(m--){ int u,v; cin >> u >> v; g[u].push_back(v); g[v].push_back(u); }
    bool ok=true;
    for(int i=1;i<=n;i++) if(color[i]==0) ok &= dfs(i,1);
    cout << (ok?"YES":"NO");
}
```

(Phase 4 – HARD / Rank Booster Problems can be appended in the same style.)