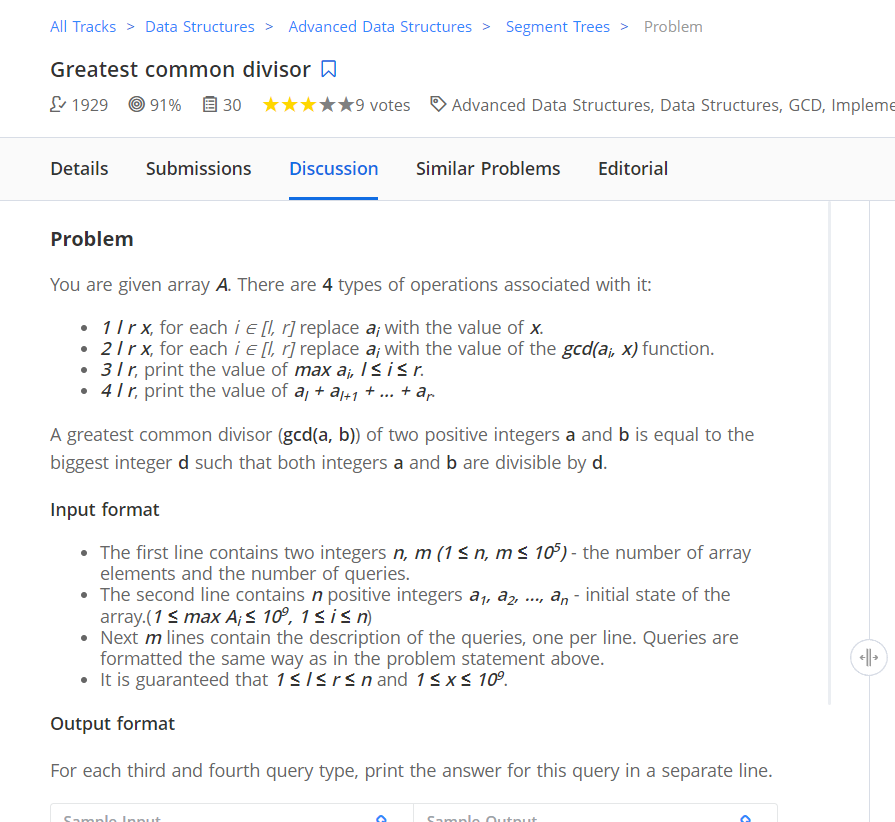
**Name :- Aarya Tiwari**

**Roll number : - 16010421119**

**Topic :- Segment Tree**

**Question:-** 

**Program Code: -**

#include<bits/stdc++.h>

using namespace std;

const int MAX\_N = 2e5 + 111;

struct node{

long long sum;

int element;

int mx;

int lz;

int l, r;

node(){

element = -1;

sum = 0;

mx = 0;

lz = -1;

l = 0;

r = 0;

}

node(int value){

element = value;

sum = value;

mx = value;

}

};

node t[4 \* MAX\_N];

int a[MAX\_N];

int n, q;

inline node merge(node a, node b){

node c = node();

c.sum = a.sum + b.sum;

c.mx = max(a.mx, b.mx);

if(a.element == b.element && a.element > 0)

c.element = a.element; else c.element = -1;

return c;

}

void build(int v,int tl,int tr){

if(tl == tr){

t[v] = node(a[tl]);

t[v].l = t[v].r = tl;

return;

}

int tm = (tl + tr) >> 1,

L = v << 1,

R = L | 1;

build(L, tl, tm);

build(R, tm + 1, tr);

t[v] = merge(t[L],t[R]);

t[v].l = tl;

t[v].r = tr;

}

inline void assignment(int v,int val){

t[v].lz = val;

t[v].sum = (t[v].r - t[v].l + 1) \*1ll\* val;

t[v].mx = val;

t[v].element = val;

}

inline void push(int v,int L,int R){

if(t[v].lz != -1){

if(t[v].l != t[v].r){

assignment(L, t[v].lz);

assignment(R, t[v].lz);

int l = t[v].l,

r = t[v].r;

t[v] = merge(t[L],t[R]);

t[v].l = l;

t[v].r = r;

}

t[v].lz = -1;

}

}

void update1(int v,int l,int r,int val){

if(t[v].l > r || t[v].r < l)return;

if(t[v].l >= l && t[v].r <= r){

assignment(v, val);

return;

}

int L = v << 1,

R = L | 1;

push(v, L, R);

update1(L, l, r, val);

update1(R, l, r, val);

int l\_ = t[v].l,

r\_ = t[v].r;

t[v] = merge(t[L], t[R]);

t[v].l = l\_;

t[v].r = r\_;

}

void update2(int v,int l,int r,int val){

if(t[v].l > r || t[v].r < l)return;

if(t[v].l >= l && t[v].r <= r && t[v].element > 0){

assignment(v, \_\_gcd(val, t[v].element));

return;

}

int L = v << 1,

R = L | 1;

push(v, L, R);

update2(L, l, r, val);

update2(R, l, r, val);

int l\_ = t[v].l,

r\_ = t[v].r;

t[v] = merge(t[L], t[R]);

t[v].l = l\_;

t[v].r = r\_;

}

node NULL\_;

node get(int v,int l,int r){

if(t[v].l > r || t[v].r < l)return NULL\_;

int L = v << 1,

R = L | 1;

push(v, L, R);

if(t[v].l >= l && t[v].r <= r)return t[v];

int tm = (t[v].l + t[v].r) >> 1;

if(r <= tm)return get(L, l, r);

if(l > tm)return get(R, l, r);

node ans1 = get(L, l, r),

ans2 = get(R, l, r);

node ans = merge(ans1, ans2);

return ans;

}

int main(){

ios\_base::sync\_with\_stdio(false);

cin.tie(NULL);

cout.tie(NULL);

// freopen("input.txt","r",stdin);freopen("output.txt","w",stdout);

int n, q;

cin >> n >> q;

for(int i = 1; i <= n; ++i){

cin >> a[i];

}

build(1, 1, n);

while(q-->0){

int type, l, r;

cin >> type >> l >> r;

if(type == 1){

int x;

cin >> x;

update1(1, l, r, x);

}else

if(type == 2)

{

int x;

cin >> x;

update2(1, l, r, x);

}else{

if(type == 3)cout << get(1, l, r).mx << '\n';

else cout << get(1, l, r).sum << '\n';

}

}

return 0;

}

**Output:-**

