

```

int n;
cin >> n;
int arr[n];
for(int i = 0; i < n; i++){
    cin >> arr[i];
}
sort(arr, arr+n);
int Min = n;
for(int i = 0; i < n; i++){
    Min = min (Min, (n - i -1 + arr[i]));
}
cout << Min << '\n';

```

```

string s;
vector<char> st;
cin >> s;
for(int i = 0; i < s.size(); i++){
    if(s[i] != '<'){
        st.push_back(s[i]);
    }else if(st.size() != 0){
        st.pop_back();
    }
}
for(int i = 0; i < st.size(); i++){
    cout << st[i];
}

```

```

int n;
while(cin >> n && n){
    cin.ignore();
    string s;
    map<string, set<string>> dish;
    for(int i = 0; i < n; i++){
        getline(cin, s);
        stringstream ss;
        ss << s;
        string x, y;
        ss >> y;
        while(ss >> x){
            dish[x];
            dish[x].insert(y);
        }
    }
    for(auto x : dish){
        cout << x.first;
        for(auto z : x.second){
            cout << " " << z;
        }
        cout << '\n';
    }
    cout << '\n';
}

```

```

string s;
while(cin >> s){
    bool start = 0;
    bool zero = 1;
    int num = 0;
    unsigned long long int ans = 0;

```

```

for(int i = 0; i < s.size(); i++){
    if(start == false){
        if( s[i] == 'x' || s[i] == 'X'){
            if(i && s[i-1] == '0'){
                start = 1;
                cout << "0" << s[i];
            }
        }
        continue;
    }
    if(isdigit(s[i]) ){
        num++;
        cout << s[i];
        ans *= 16;
        ans += s[i] - '0';
        if(zero && ans == 0) num--;
        else zero =false;
        if(num >= 8){
            start = 0;
            zero =1;
            cout << " " << ans << '\n';
            ans = 0;
            num = 0;
        }
    }
    }else if(( s[i] >= 'a' && s[i] <= 'f' )){
        num++;
        zero =false;
        cout << s[i];
        ans *= 16;
        ans += s[i] - 'a' + 10;
        if(num >= 8){
            start = 0;
            zero =1;
            cout << " " << ans << '\n';
            ans = 0;
            num = 0;
        }
    }
    }else if(( s[i] >= 'A' && s[i] <= 'F' )){
        num++;
        zero =false;
        cout << s[i];
        ans *= 16;
        ans += s[i] - 'A' + 10;
        if(num >= 8){
            start = 0;
            zero =1;
            cout << " " << ans << '\n';
            ans = 0;
            num = 0;
        }
    }
    }else{
        start = 0;
        zero =1;
        cout << " " << ans << '\n';
        ans = 0;
        num = 0;
        continue;
    }
}
if(start){
    start = 0;
    zero =1;
    cout << " " << ans << '\n';
    ans = 0;
    num = 0;
}

```

```
}
```

```
#define int long long
int32_t main(){
    int t, n;
    cin >> t;
    while(t--){
        cin >> n;
        priority_queue<int, vector<int>, greater<int>> pq;
        for(int i = 0; i < n; i++){
            int x;
            cin >> x;
            pq.push(x);
        }
        int ans = 0;
        while(pq.size() > 1 ){
            int sum = 0;
            sum += pq.top();
            pq.pop();
            sum += pq.top();
            pq.pop();
            ans += sum;
            pq.push(sum);
        }
        cout << ans << '\n';
    }
}
```

```
int n, t;
cin >> n >> t;
set<pair<int,int>> person;
for(int i = 0; i < n; i++){
    int a, b;
    cin >> a >> b;
    person.insert(make_pair(b,a));
}
int ans = 0;
for(int i = t - 1; i >= 0; i--){
    auto x = person.lower_bound(make_pair(i, 0));
    auto Max = x;
    for(; x != person.end(); x++){
        if(Max->second < x->second) Max = x;
    }
    if(Max != person.end()){
        ans += Max->second;
        person.erase(Max);
    }
}
cout << ans << '\n';
```

```
int n, m, p;
cin >> n >> m >> p;
double c[n];
double d[m];
for(int i = 0; i < n; i++){cin >> c[i];}
for(int i = 0; i < m; i++){cin >> d[i];}
double r[n*m] = {};
int k = 0;
for(int i = 0; i < n; i++){
    for(int j = 0; j < m; j++){
        r[k++] = c[i]/d[j];
    }
}
sort(r, r + k);
```

```

for(int i = 0; i < k - 1; i++){
    double cur = r[i];
    double next = r[i + 1];
    if(cur + cur * p / 100 < next){
        cout << "Time to change gears!\n";
        return 0;
    }
}
cout << "Ride on!\n";

```

```

int n;
cin >> n;
int arr[n] = {};
for(int i = 0; i < n; i++) {
    for(int j = 0; j < n; j++) {
        int x;
        cin >> x;
        arr[n] |= x;
    }
}
for(int i = 0; i < n; i++) {
    cout << arr[i] << " ";
}
cout << '\n';

```

```

import math
n, k = map(int, input().split())
if n <= math.pow(2,k):
    print("Your wish is granted!")
else:
    print("You will become a flying monkey!")

```

```

class BigInteger
{
    friend ostream& operator << (ostream &out, const BigInteger &num);
public:
    BigInteger(long long num = 0){
        *this = num;
    }
    BigInteger(const string &numStr){
        *this = numStr;
    }
    BigInteger& operator = (long long num) {
        m_bits.clear();
        do {
            m_bits.push_back(num % BASE);
            num /= BASE;
        } while (num > 0);
        return *this;
    }
    BigInteger& operator = (const string &numStr){
        m_bits.clear();
        int bit, len = (numStr.length() - 1) / WIDTH + 1;
        for (int i = 0; i < len; i++){
            int end = numStr.length() - i * WIDTH;
            int start = max(0, end - WIDTH);
            sscanf(numStr.substr(start, end - start).c_str(), "%d", &bit);
            m_bits.push_back(bit);
        }
        return *this;
    }
    BigInteger operator + (const BigInteger &rhs) const{
        BigInteger result;
        result.m_bits.clear();

```

```

        for (int i = 0, carry = 0; ; i++){
            if (i >= m_bits.size() && i >= rhs.m_bits.size() && carry == 0)
                break;
            int num = carry;
            if (i < m_bits.size())
                num += m_bits[i];
            if (i < rhs.m_bits.size())
                num += rhs.m_bits[i];
            result.m_bits.push_back(num % BASE);
            carry = num / BASE;
        }
        return result;
    }

    BigInteger& operator += (const BigInteger &rhs){
        *this = *this + rhs;
        return *this;
    }

private:
    static const int BASE = 10000;
    static const int WIDTH = 4;
    vector<int> m_bits;
};

ostream& operator << (ostream &out, const BigInteger &num){
    static string format = "%0" + to_string(BigInteger::WIDTH) + "d";
    out << num.m_bits.back();
    for (int i = num.m_bits.size() - 2; i >= 0; i--){
        char buf[20];
        sprintf(buf, format.c_str(), num.m_bits[i]);
        out << buf;
    }
    return out;
}

map<pair<int, int>, BigInteger> F;
int c = 0, d = 0;
vector<int> a;
vector<int> b;

BigInteger f(int x, int y){
    if(x > 0 && y > 0){
        pair<int,int> t = make_pair(x,y);
        if(F.find(t) != F.end()){
            return F[t];
        }
        BigInteger ans = 0;
        for(int i = 0; i < a.size(); i++){
            ans += f(x - a[i], y - b[i]);
        }
        ans += c;
        F[t] = ans;
        return ans;
    }else{
        return d;
    }
}

int main(){
    int n;
    cin >> n;
    cin.ignore();
    while(n--){
        a.clear();
        b.clear();
        F.clear();
        string s;
        getline(cin, s);
        stringstream ss;
        ss << s;
        int num1, num2;

```

```

while(ss >> num1 >> num2){
    a.push_back(num1);
    b.push_back(num2);
}
c = a[a.size() - 1];
d = b[b.size() - 1];
a.pop_back();
b.pop_back();
s.clear();
getline(cin, s);
ss.clear();
ss << s;
vector<int> x;
vector<int> y;
while(ss >> num1 >> num2){
    x.push_back(num1);
    y.push_back(num2);
}
for(int i = 0; i < x.size(); i++){
    cout << f(x[i], y[i]) << '\n';
}
cout << '\n';
}

```

```

int n;
cin >> n;
bool TF[n];
for(int i = 0; i < n; i++){
    char x;
    cin >> x;
    if(x == 'T'){
        TF[i] = 1;
    }else{
        TF[i] = 0;
    }
}
cin.ignore();
string s;
getline(cin, s);
stack<bool> st;
int num = 0;
for(int i = 0; i < s.size(); i += 2){
    if(s[i] == '*'){
        bool x = st.top();
        st.pop();
        bool y = st.top();
        st.pop();
        st.push(x && y);
    }else if(s[i] == '-'){
        bool x = st.top();
        st.pop();
        st.push(!x);
    }else if(s[i] == '+'){
        bool x = st.top();
        st.pop();
        bool y = st.top();
        st.pop();
        st.push(x || y);
    }else{
        st.push(TF[num++]);
    }
}
if(st.top() == 1){
    cout << "T\n";
}else{

```

```
    cout << "F\n";  
}
```

```
int n;  
cin >> n;  
string s;  
cin.ignore();  
getline(cin,s);  
stack<char> st;  
for(int i = 0; i < n; i++){  
    char x = s[i];  
    if(x == ' '){  
        continue;  
    }else if(x == '{'){  
        st.push(x);  
    } else if(x == '['){  
        st.push(x);  
    } else if(x == '('){  
        st.push(x);  
    } else if(x == ')'){  
        if(st.size() == 0){  
            cout << x << " " << i << '\n';  
            return 0;  
        }  
        if(st.top() == '('){  
            st.pop();  
        }else{  
            cout << x << " " << i << '\n';  
            return 0;  
        }  
    } else if(x == ']'){  
        if(st.size() == 0){  
            cout << x << " " << i << '\n';  
            return 0;  
        }  
        if(st.top() == '['){  
            st.pop();  
        }else{  
            cout << x << " " << i << '\n';  
            return 0;  
        }  
    } else if(x == '}'){  
        if(st.size() == 0){  
            cout << x << " " << i << '\n';  
            return 0;  
        }  
        if(st.top() == '{'){  
            st.pop();  
        }else{  
            cout << x << " " << i << '\n';  
            return 0;  
        }  
    }  
}  
cout << "ok so far\n";
```

```
int ans = 0;  
int n;  
cin >> n;  
int h[n];  
for(int i = 0; i < n; i++){  
    cin >> h[i];  
}  
stack<pair<int,int>> st;
```

```

for(int i = 0; i < n; i++){
    int bottom = h[i];
    while(st.size() > 0){
        pair<int, int> last = st.top();
        st.pop();
        cout << "bottom:" << bottom << '\n';
        cout << "last.first :" << last.first << '\n';
        cout << "last.second :" << last.second << '\n';
        cout << "h[i]:" << h[i] << '\n';
        bottom = min(last.second, bottom);
        cout << "bottom:" << bottom << '\n';
        if(last.first > h[i]){
            st.push(make_pair(last.first, bottom));
            ans = max(ans, h[i] - bottom);
            cout << ans << '\n';
            break;
        }
        ans = max(ans, last.first - bottom);
        cout << ans << '\n';
    }
    st.push(make_pair(h[i], h[i]));
    cout << "h[i] x" << h[i] << '\n';
}
cout << ans << '\n';

```

```

cin.tie(0);
ios_base::sync_with_stdio(0);
int n;
cin >> n;
cin.ignore();
int index = 0;
while(n--){
    string s;
    getline(cin, s);
    list<char> arr;
    index = 0;
    for(int i = 0; i < s.size(); i++){
        if(s[i] == '<'){
            auto x = arr.begin();
            index--;
            if(index < 0){
                index = 0;
            }else if(index == arr.size()-1){
                arr.pop_back();
            }else{
                advance(x, index);
                arr.erase(x);
            }
        }else if(s[i] == '['){
            index = 0;
        }else if(s[i] == ']'){
            index = arr.size();
        }else{
            if(index == arr.size()){
                arr.push_back(s[i]);
            }else{
                auto x = arr.begin();
                advance(x, index);
                arr.insert(x, s[i]);
            }
            index++;
        }
    }
}
for(auto x : arr){
    cout << x;
}

```



```

    }
    cout << '\n';
}

```

```

vector<vector<int>> adj;
vector<int> tag;
vector<int> ans;
int n, m;
void dfs(int now, bool bug){
    if(tag[now] != 0) return;
    bool end = true;
    for(auto next : adj[now]){
        if(next < 0){
            end = false;
            tag[now] = 1;
            dfs(-next, bug);
            tag[now] = 0;
        }else if(bug == false){
            tag[now] = 1;
            dfs(next, true);
            tag[now] = 0;
        }
    }
    if(end){
        ans[now] = 1;
    }
}
int main(){
    cin >> n >> m;
    adj.resize(n + 1);
    tag.resize(n + 1);
    ans.resize(n + 1);
    for(int i = 0; i < m; i++){
        int a, b;
        cin >> a >> b;
        if(a < 0){
            adj[-a].push_back(-b);
        }else{
            adj[a].push_back(b);
        }
    }
    dfs(1, false);
    int num = 0;
    for(auto x : ans){
        num += x;
    }
    cout << num << '\n';
}

```

```

int arr[2501][2501];
vector<int> tag;
vector<pii> ans;
priority_queue<pip, vector<pip>, greater<pip>> pq;
int n;
void bfs(){
    int num = 0;
    tag[0] = 1;
    for(int i = 1; i < n; i++){
        pq.push(make_pair(arr[0][i], make_pair(0, i)));
    }
    while(num < n - 1){
        pip now = pq.top();
        pq.pop();
        if(tag[now.sec.sec] != 0) continue;
        tag[now.sec.sec] = 1;
    }
}

```

```

num++;
ans.push_back(now.sec);
int x = now.sec.sec;
for(int i = 0; i < n; i++){
    if(i == x) continue;
    pq.push(make_pair(arr[x][i], make_pair(x, i)));
}
}
}
int main(){
    cin >> n;
    tag.resize(n);
    for(int i = 0; i < n; i++){
        for(int j = 0; j < n; j++){
            cin >> arr[i][j];
        }
    }
    bfs();
    for(auto x : ans){
        cout << x.fst + 1 << " " << x.sec + 1;
        cout << '\n';
    }
}

```

```

#define pii pair<int, int>
#define pip pair<int, pii>
#define fst first
#define sec second
int r, c;
int arr[1001][1001];
int ans = 0;
int Min;
vector<vector<int>> tag;
void bfs(int x, int y){
    priority_queue<pip, vector<pip>, greater<pip>> pq;
    pq.push(make_pair(arr[x][y], make_pair(x, y)));
    ans = arr[x][y];
    int t = x;
    while(pq.size() != 0){
        pip now = pq.top();
        pq.pop();
        ans = max(now.fst, ans);
        if(tag[now.sec.fst][now.sec.sec] != 0) continue;
        if(t != 0 && ans >= Min) return;
        if(now.sec.sec == c - 1) return;
        tag[now.sec.fst][now.sec.sec] = 1;
        x = now.sec.fst;
        y = now.sec.sec;
        if(x+1 < r) pq.push(make_pair(arr[x+1][y], make_pair(x+1, y)));
        if(x-1 >= 0) pq.push(make_pair(arr[x-1][y], make_pair(x-1, y)));
        if(y+1 < c) pq.push(make_pair(arr[x][y+1], make_pair(x, y+1)));
        if(y-1 >= 0) pq.push(make_pair(arr[x][y-1], make_pair(x, y-1)));
    }
}
int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    cin >> r >> c;
    for(int i = 0; i < r; i++){
        for(int j = 0; j < c; j++){
            cin >> arr[i][j];
        }
    }
    ans = 0;
    tag.clear();
    tag.resize(r);
}

```

```

for(int i = 0; i < r; i++) tag[i].resize(c);
bfs(0,0);
Min = ans;
for(int i = 1; i < r; i++){
    ans = 0;
    tag.clear();
    tag.resize(r);
    for(int i = 0; i < r; i++) tag[i].resize(c);
    bfs(i,0);
    Min = min(Min, ans);
}
cout << Min << '\n';

```

```

vector<int> arr;vector<int> Rank;
int find(int x){
    if(arr[x] == x)
        return x;
    return arr[x] = find(arr[x]);
}
void Union(int a, int b){
    int ra = find(a);
    int rb = find(b);
    if(ra != rb){
        if(Rank[ra] > Rank[rb]){
            arr[rb] = ra;
            Rank[ra] += Rank[rb];
        }else{
            arr[ra] = rb;
            Rank[rb] += Rank[ra];
        }
    }
}
int main(){
    int n, m;cin >> n >> m;n++;Rank.resize(n);arr.resize(n);
    for(int i = 1; i < n; i++){
        arr[i] = i;
        Rank[i] = 1;
    }
    for(int i = 0; i < m; i++){
        int x, y;
        cin >> x >> y;
        Union(x,y);
    }
    int t = find(1);
    if(Rank[t] == n - 1){
        cout << "Connected\n";
    }else{
        for(int i = 2; i < n; i++){
            if(find(i) != t){
                cout << i << '\n';
            }
        }
    }
}

```

```

vector<int> arr;vector<int> Rank;
int find(int x){
    if(arr[x] == x)
        return x;
    return arr[x] = find(arr[x]);
}
void Union(int a, int b){
    int ra = find(a);
    int rb = find(b);
    if(ra != rb){

```

```

        if(Rank[ra] > Rank[rb]){
            arr[rb] = ra;
            Rank[ra] += Rank[rb];
        }else{
            arr[ra] = rb;
            Rank[rb] += Rank[ra];
        }
    }
}

int main(){
    int n, q;
    cin >> n >> q;
    n++;
    Rank.resize(n);
    arr.resize(n);
    for(int i = 1; i < n; i++){
        arr[i] = i;
        Rank[i] = 1;
    }
    for(int i = 0; i < q; i++){
        char op;
        int x, y;
        cin >> op >> x >> y;
        if(op == '?'){
            if(find(x) == find(y)){
                cout << "yes\n";
            }else{
                cout << "no\n";
            }
        }else{
            Union(x,y);
        }
    }
}

```

```

vector<int> num;
vector<ll> sum;
vector<int> id;
vector<int> fa;
int find(int x){
    if(fa[x] == x)
        return x;
    return fa[x] = find(fa[x]);
}

void Union(int a, int b){
    int ra = find(a);
    int rb = find(b);
    if(ra != rb){
        if(num[ra] > num[rb]){
            fa[rb] = ra;
            sum[ra] += sum[rb];
            num[ra] += num[rb];
        }else{
            fa[ra] = rb;
            sum[rb] += sum[ra];
            num[rb] += num[ra];
        }
    }
}

int main(){
    cin.tie(0);
    ios_base::sync_with_stdio(0);
    int n, q;
    while(cin >> n >> q){
        n++;
        id.resize(n);

```

```

num.resize(n);
fa.resize(n);
sum.resize(n);
for(int i = 1; i < n; i++){
    id[i] = i;
    fa[i] = i;
    sum[i] = i;
    num[i] = 1;
}
for(int i = 0; i < q; i++){
    int op;
    int x, y;
    cin >> op;
    if(op == 1){
        cin >> x >> y;
        Union(id[x], id[y]);
    }else if(op == 2){
        cin >> x >> y;
        int fx = find(id[x]);
        int fy = find(id[y]);
        if(fx != fy){
            sum[fx] -= x;
            num[fx]--;
            int new_index = sum.size();
            id[x] = new_index;
            fa.push_back(new_index);
            sum.push_back(x);
            num.push_back(1);
            Union(id[x], id[y]);
        }
    }else{
        cin >> x;
        int index = find(id[x]);
        cout << num[index] << " " << sum[index] << '\n';
    }
}
}

```

```

int n, m;
int arr[502][502] = {};
int tag[502][502] = {};
int count1 = 0;
bool TF = false;
void dfs(int x, int y, int h){
    if(tag[x][y] == 0) return;
    if(arr[x][y] < h){
        TF = false;
        return;
    }
    if(tag[x][y] != -1) return;
    if(arr[x][y] > h){
        return;
    }
    tag[x][y] = 10;
    count1++;
    dfs(x+1, y, h);
    dfs(x-1, y, h);
    dfs(x, y+1, h);
    dfs(x, y-1, h);
}
int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    cin >> m >> n;
    for(int i = 1; i <= n; i++){

```

```

        for(int j = 1; j <= m; j++){
            cin >> arr[i][j];
            tag[i][j] = -1;
        }
    }
    for(int i = 1; i <= m; i++){
        tag[n + 1][i] = 0;
    }
    for(int i = 1; i <= n; i++){
        tag[i][m + 1] = 0;
    }
    int ans = 0;
    for(int i = 1; i <= n; i++){
        for(int j = 1; j <= m; j++){
            count1 = 0;
            TF = true;
            dfs(i, j, arr[i][j]);
            ans += TF * count1;
        }
    }
    cout << ans << '\n';

```

```

void discompute(vector<vector<pii>>& adj, vector<int>& spiders, vector<int>& dis){
    priority_queue<pii, vector<pii>, greater<pii>> pq;
    for(auto x : spiders){
        pq.push({0, x});
        dis[x] = 0;
    }
    while(pq.size() != 0){
        pii now = pq.top();
        pq.pop();
        for(auto next : adj[now.sec]){
            int d = now.fst + next.sec;
            if(dis[next.fst] == -1){
                dis[next.fst] = d;
                pq.push({d, next.fst});
            }else{
                if(d >= dis[next.fst]) continue;
                dis[next.fst] = d;
                pq.push({d, next.fst});
            }
        }
    }
}

bool farpath(vector<vector<pii>>& adj, vector<int>& dis, const int S, const int E, const int allow, const int t){
    if(dis[S] < allow) return false;
    if(dis[E] < allow) return false;
    priority_queue<pii, vector<pii>, greater<pii>> pq;
    pq.push({0, S});
    vector<int> meter(dis.size(), -1);
    while(pq.size() != 0){
        pii now = pq.top();
        pq.pop();
        if(now.fst > t) return false;
        for(auto next : adj[now.sec]){
            if(dis[next.fst] < allow) continue;
            if(next.fst == E) return true;
            int d = now.fst + next.sec;
            if(meter[next.fst] == -1){
                meter[next.fst] = d;
                pq.push({d, next.fst});
            }else{
                if(d >= meter[next.fst]) continue;
                meter[next.fst] = d;
                pq.push({d, next.fst});
            }
        }
    }
}

```

```

    }
}
return false;
}
int main(){
    int n, m, t, S, E, num;
    cin >> n >> m >> t;
    vector<vector<pii>> adj(n);
    vector<int> spiders;
    vector<int> dis(n, -1);
    for(int i = 0; i < m; i++){
        int u, v, d;
        cin >> u >> v >> d;
        adj[u].push_back({v, d});
        adj[v].push_back({u, d});
    }
    cin >> S >> E;
    cin >> num;
    spiders.resize(num);
    for(auto& x : spiders) cin >> x;
    discompute(adj, spiders, dis);
    const int inf = 1000000000;
    int l = 0, r = inf;
    int ans = 0;
    while(l <= r){
        int m = (l + r) / 2;
        if(farpath(adj, dis, S, E, m, t)){
            ans = max(m, ans);
            l = m + 1;
        }else{
            r = m - 1;
        }
    }
    cout << ans << '\n';
}

```

```

int T;
cin >> T;
while(T--){
    int n;
    cin >> n;
    int arr[n];
    for(int i = 0; i < n; i++){
        cin >> arr[i];
    }
    for(int i = 1; i < n; i++){
        arr[i] += arr[i - 1];
    }
    map<ll, int> dif47;
    dif47[0] = 1;
    int ans = 0;
    for(int i = 0; i < n; i++){
        ans += dif47[arr[i] - 47];
        dif47[arr[i]]++;
    }
    cout << ans << '\n';
}

```

```

int t[4];
for(int i = 0; i < 4; i++){
    double x;
    cin >> x;
    t[i] = x * 1000;
}

```

```

double x;
cin >> x;
int win = x * 1000;
sort(t, t+4);
int sum1 = t[0] + t[1] + t[2];
int sum2 = t[1] + t[2] + t[3];
if(sum1 / 3 > win){
    cout << "impossible\n";
}else if(sum2 / 3 <= win){
    cout << "infinite\n";
}else{
    int ans = win * 3 - t[1] - t[2];
    int x = ans % 10;
    ans /= 10;
    int x1 = ans % 10 + (x >=5);
    ans/=10;
    int x2 = ans % 10 + x1/10;
    ans/= 10;
    ans += x2/10;
    cout << ans << '.' << x2%10 << x1%10 << '\n';
}

```

```

void modify(vector<int>& bit, int index, int num){
    if(index == 0) {
        bit[0] += num;
        return;
    }
    int n = bit.size();
    for(int i = index; i <= n; i += lowbit(i) ){
        bit[i] += num;
    }
}

int query(vector<int>& bit, int index){
    int ans = 0;
    for(int i = index; i > 0; i -= lowbit(i) ){
        ans += bit[i];
    }
    if(index != -1) ans += bit[0];
    return ans;
}

int32_t main(){
    vector<int> bit;
    int n, q;
    cin >> n >> q;
    bit.resize(n);
    while(q--){
        char op;
        cin >> op;
        int index, num;
        switch(op){
            case '+':{
                cin >> index >> num;
                modify(bit, index, num);
                break;
            }
            case '?':{
                cin >> index;
                cout << query(bit, index - 1) << '\n';
                break;
            }
        }
    }
}

```

```

#include <bits/stdc++.h>

```



```

#define ll long long
using namespace std;
struct countV{
    int numV[7];
};
int lowbit(int x){
    return x & (-x);
}
void modify(vector<countV>& bit, int index, int minV, int addV){
    int n = bit.size();
    for(int i = index; i <= n; i += lowbit(i)){
        bit[i].numV[minV]--;
        bit[i].numV[addV]++;
    }
}
ll query(vector<countV>& bit, vector<int>& V, int index){
    ll ans = 0;
    for(int i = index; i > 0; i -= lowbit(i)){
        for(int j = 1; j < 7; j++){
            //cout << i << " " << j << " " << V[j] << " " << bit[i].numV[j] << '\n';
            ans += (ll) V[j] * bit[i].numV[j];
        }
    }
    return ans;
}
void create(vector<countV>& bit, vector<int>& V, string& s){
    for(int i = 1; i < s.size(); i++){
        modify(bit, i, 0, int(s[i]) - int('0'));
    }
}
int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    int n, q, x, y;
    char op;
    string s;
    vector<countV> bit;
    vector<int> V(7);
    cin >> n >> q;
    bit.resize(n + 1);
    for(int i = 1; i < 7; i++){
        cin >> V[i];
    }
    cin >> s;
    s = " " + s;
    create(bit, V, s);
    while(q--){
        cin >> op >> x >> y;
        switch(op){
            case '1':{
                modify(bit, x, int(s[x]) - int('0'), y);
                s[x] = '0' + y;
                break;
            }
            case '2':{
                V[x] = y;
                break;
            }
            case '3':{
                cout << query(bit, V, y) - query(bit, V, x - 1) << '\n';
                break;
            }
        }
    }
}

```

```

int n, m, a, c, x;
cin >> n >> m >> a >> c >> x;
ll arr[n];
arr[0] = ( (ll) a * x + c ) % m;
for(int i = 1; i < n; i++){
    arr[i] = (a * arr[i - 1] + c) % m;
}
int ans = 0;
for(int i = 0; i < n; i++){
    int l = 0, r = n - 1;
    while(r - l >= 0){
        int m = (l + r) / 2;
        if(arr[m] == arr[i]){
            ans++;
            break;
        }else if(arr[i] > arr[m]){
            l = m + 1;
        }else{
            r = m - 1;
        }
    }
}
cout << ans << '\n';

```

```

int main(){
    int n, k;
    cin >> n >> k; vector<int> bit; vector<bool> arr; bit.resize(n + 1); arr.resize(n + 1);
    while(k--){
        char op;
        cin >> op;
        int l, r;
        switch(op){
            case 'F':{
                cin >> l;
                arr[l] = !arr[l];
                modify(bit, l, arr[l] ? 1 : -1);
                break;
            }
            case 'C':{
                cin >> l >> r;
                cout << query(bit, r) - query(bit, l - 1) << '\n';
                break;
            }
        }
    }
}

```

```

operators = [' + ', ' - ', ' * ', ' // ']
values = {}
for a in operators:
    for b in operators:
        for c in operators:
            val_str = "4{:s}4{:s}4{:s}4".format(a, b, c)
            val = eval(val_str)
            values[val] = val_str.replace('///', '/') + " = {:d}".format(val)

for i in range(0, int(input())):
    n = int(input())
    if n < -60 or n > 256 or n not in values:
        print("no solution")
    else:
        print(values[n])

```

```

const int N = 100001;

```

```

const ll K = 1000000000000000000;
vector<ll> fib;
void sol(int n, ll k){
    if(n == 1) cout << "N\n";
    else if(n == 2) cout << "A\n";
    else{
        ll x = fib[n - 2];
        if(k <= x){
            sol(n - 2, k);
        }else{
            sol(n - 1, k - x);
        }
    }
}
int32_t main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    fib.resize(N);
    fib[0] = 0;
    fib[1] = 1;
    for(int i = 2; i < N; i++){
        fib[i] = fib[i - 2] + fib[i - 1];
        if(fib[i] > K) fib[i] = K + 1;
    }
    int n;
    ll k;
    cin >> n >> k;
    sol(n, k);
}

```

```

bool isvalid(string& s){
    bool isL = false, isAB = false;
    int numAB = 0;
    for(auto x : s){
        if(x == 'A' || x == 'E' || x == 'I' || x == 'O' || x == 'U'){
            if(isAB){
                numAB++;
            }else{
                isAB = true;
                numAB = 1;
            }
        }else{
            if(x == 'L') isL = true;
            if(!isAB){
                numAB++;
            }else{
                isAB = false;
                numAB = 1;
            }
        }
    }
    if(numAB >= 3) return false;
}
return isL;
}

void solve(ll& ans, string& s, vector<int>& ch, int index){
    if(index >= s.size()){
        if(isvalid(s)){
            ll num = 1;
            for(auto x : ch) num *= x;
            ans += num;
        }
        return;
    }
    if(s[index] != '_') solve(ans, s, ch, index + 1);
    else{
        s[index] = 'L';
    }
}

```

```

        solve(ans, s, ch, index + 1);
        s[index] = 'B';
        ch[index] = 20;
        solve(ans, s, ch, index + 1);
        s[index] = 'A';
        ch[index] = 5;
        solve(ans, s, ch, index + 1);
        s[index] = '_';
        ch[index] = 1;
    }
}

int main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    string s;
    cin >> s;
    vector<int> ch(101, 1);
    ll ans = 0;
    solve(ans, s, ch, 0);
    cout << ans << '\n';
}

```

```

#define square vector<vector<int>>
square mult(square& A, square& B){
    square C;
    int n = A.size();
    int m = B.size();
    int l = B[0].size();
    C.resize(n);
    for(int i = 0; i < n; i++){
        C[i].resize(l);
        for(int j = 0; j < m; j++){
            for(int k = 0; k < l; k++){
                C[i][k] += A[i][j] * B[j][k];
            }
        }
    }
    return C;
}

square pow1(square base, int num){
    square ans;
    int n = base.size();
    int m = base[0].size();
    ans.resize(n);
    for(int i = 0; i < n; i++){
        ans[i].resize(m);
        ans[i][i] = 1;
    }
    while(num){
        if(num & 1) ans = mult(ans, base);
        base = mult(base, base);
        num >>= 1;
    }
    return ans;
}

int fib(int n){
    square A = {
        {1, 1},
        {1, 0}
    };
    if(n <= 1) return n;
    return pow1(A, n-1)[0][0];
}

```

```

vector<int> legal_num;

```

```

vector<vector<int>> next_num = {
    {0},{0, 1, 2, 3 ,4 ,5 ,6 ,7, 8, 9},{2, 3, 5, 6, 8, 9, 0},{3, 6, 9},{4, 5, 6, 7 ,8 ,9, 0},{5, 6, 8, 9, 0},{6, 9},
    {7, 8, 9, 0},{8, 9, 0},{9}};
void dfs(int now, int d, int num){
    if(d <= 0) return;
    legal_num.push_back(num);
    for(auto x : next_num[now]){
        dfs(x, d - 1, num * 10 + x);
    }
}
int main(){
    int t;
    legal_num.push_back(0);
    dfs(1, 3, 1);
    dfs(2, 3, 2);
    for(int i = 3; i < 10; i++) dfs(i, 2, i);
    sort(legal_num.begin(), legal_num.end());
    cin >> t;
    while(t--){
        int n;
        cin >> n;
        int l = lower_bound(legal_num.begin(), legal_num.end(), n) - legal_num.begin();
        if(abs(legal_num[l] - n) < abs(legal_num[l - 1] - n)){
            cout << legal_num[l] << '\n';
        }else{
            cout << legal_num[l - 1] << '\n';
        }
    }
}

```

```

#define MAX 200005
int lowbit(int x){
    return x & (-x);
}
void modify(vector<ll>& arr, int index, ll num){
    int n = arr.size();
    for(int i = index; i <= n; i += lowbit(i)){
        arr[i] += num;
    }
}
ll query(vector<ll>& arr, int index) {
    ll ans = 0;
    for (int i = index; i > 0; i -= lowbit(i)){
        ans += arr[i];
    }
    return ans;
}
int main () {
    int n;
    cin >> n;
    vector<ll> arr(n);
    vector<ll> bit1(n + 1), bit2(n + 1);
    for(auto &x : arr) cin >> x;
    ll ans = 0;
    for (int i = n - 1; i >= 0; i--) {
        ans += query(bit2, arr[i] - 1);
        ll num = query(bit1, arr[i] - 1);
        modify(bit1, arr[i], 1);
        modify(bit2, arr[i], num);
    }
    cout << ans << '\n';
}

```

```

int pow1(int base, int p, int m){
    int ans = 1;
    while(p){

```

```

        if(p & 1) ans = (ans * base) % m;
        base = (base * base) % m;
        p >>= 1;
    }
    return ans;
}

int32_t main(){
    ios::sync_with_stdio(0);
    cin.tie(0);
    int a, b;
    cin >> a >> b;
    cout << pow1((a * (a + 1) / 2) % a, b, a) % a << '\n';
}

```

```

#include <bits/stdc++.h>
#include <ext/pb_ds/assoc_container.hpp>
#include <ext/pb_ds/tree_policy.hpp>
using namespace std;
using namespace __gnu_pbds;
typedef long long ll;
typedef long double ld;
typedef unsigned int uint;
typedef unsigned long long ull;
const ld pi = 4.0*atan1(1.0);
const int iinf = 1e9 + 10;
const ll inf = 1e18 + 10;
const int mod = 1000000007;
const ld prec = .000001;
#define endl endl
#define endl '\n'
#define pb push_back
#define debug(x) cout<<#x<<" -> "<<x<<'\n'
#define all(x) (x).begin(), (x).end()
#define rall(x) (x).rbegin(), (x).rend()
#define uni(x) (x).erase(unique(all(x)), (x).end())
#define rep(i, n) for (ll i = 0; i < (ll)(n); ++i)
#define rep1(i, n) for (ll i = 1; i <= (ll)(n); ++i)
#define umap unordered_map
#define uset unordered_set
template<class TIn>
using indexed_set = tree<
    TIn, null_type, less<TIn>,
    rb_tree_tag, tree_order_statistics_node_update>;
void fast() {
    ios::sync_with_stdio(false);
    cin.tie(NULL); cout.tie(NULL);
}
void file() {
    auto a = freopen("a.in", "r", stdin);
    auto b = freopen("a.out", "w", stdout);
    if(!a || !b) cout << "uh oh" << endl;
}
ll b, n, e;
ll bval, nval, eval;
vector<pair<ll,string>> speeds;
void prep() {
    speeds.pb({bval+bval, "bb"});
    speeds.pb({bval+nval, "bn"});
    speeds.pb({bval+eval, "be"});
    speeds.pb({nval+nval, "nn"});
    speeds.pb({nval+eval, "ne"});
    speeds.pb({eval+eval, "ee"});
    sort(all(speeds));
}
void dec(string s, ll& tb, ll& tn, ll& te) {
    for(auto i : s) {

```

```

        if(i == 'b') {
            tb--;
        }
        if(i == 'n') {
            tn--;
        }
        if(i == 'e') {
            te--;
        }
    }
}

bool ok(string s, ll tb, ll tn, ll te) {
    dec(s, tb, tn, te);
    return (tb >= 0) && (tn >= 0) && (te >= 0);
}

bool works(vector<ll> kayaks, ll speed) {
    ll tb = b;
    ll tn = n;
    ll te = e;
    bool works = true;
    for(auto i : kayaks) {
        bool thisworks = false;
        for(auto j : speeds) {
            if(i * j.first >= speed && ok(j.second,tb,tn,te)) {
                dec(j.second, tb, tn, te);
                thisworks = true;
                break;
            }
        }
        if(!thisworks) {
            works = false;
            break;
        }
    }
    return works;
}

int main() {
    cin >> b >> n >> e;
    cin >> bval >> nval >> eval;
    vector<ll> kayaks((b+n+e)/2);
    for(auto& i : kayaks) {
        cin >> i;
    }
    sort(all(kayaks));
    reverse(all(kayaks));
    prep();
    ll lo = 0;
    ll hi = (ll)100000 * 1000 * 2 * 100;
    while(hi-lo>1) {
        ll mi = (hi+lo)/2;
        if(works(kayaks,mi)) {
            lo = mi;
        }
        else {
            hi = mi;
        }
    }
    cout << lo << endl;
}

```

```

template<class TIn>
using indexed_set = tree<
    TIn, null_type, less<TIn>,
    rb_tree_tag, tree_order_statistics_node_update>;

void fast() {
    ios::sync_with_stdio(false);
}

```

```

    cin.tie(NULL); cout.tie(NULL);
}

void file() {
    auto a = freopen("a.in", "r", stdin);
    auto b = freopen("a.out", "w", stdout);
    if(!a || !b) cout << "uh oh" << endl;
}

void nextpow(ll& n) {
    n--;
    n |= n >> 1;
    n |= n >> 2;
    n |= n >> 4;
    n |= n >> 8;
    n |= n >> 16;
    n++;
}

void fft(valarray<complex<double>>& x) {
    ll s = x.size();
    if(s <= 1) {
        return;
    }
    valarray<complex<double>> even = x[slice(0, s/2, 2)];
    valarray<complex<double>> odd = x[slice(1, s/2, 2)];
    fft(even);
    fft(odd);
    for(ll i = 0; i < s/2; i++) {
        complex<double> t = polar(1.0, -2.0 * M_PI * i / s) * odd[i];
        x[i] = even[i] + t;
        x[i+s/2] = even[i] - t;
    }
}

void ifft(valarray<complex<double>>& x) {
    x = x.apply(conj);
    fft(x);
    x = x.apply(conj);
    x /= x.size();
}

int main() {
    string s;
    cin >> s;
    ll size = s.size();
    size++;
    nextpow(size);
    size++;
    nextpow(size);
    valarray<complex<double>> v1(size);
    valarray<complex<double>> v2(size);
    for(ll i = 0; i < s.size(); i++) {
        v1[i] = (s[i] - 'A')^1;
        v2[size - i - 1] = (s[i] - 'A');
    }
    fft(v1);
    fft(v2);
    for(ll i = 0; i < size; i++) {
        v1[i] *= v2[i];
    }
    ifft(v1);
    for(int i = 0; i < s.size()-1; i++) {
        cout << (int)(v1[i].real()+.25) << endl;
    }
    return 0;
}

```

```

template <typename T, typename VT = vector<complex<T>>>
struct FFT {

```



```

const T pi;
FFT(const T pi = acos((T)-1.0)) : pi(pi) {}
unsigned bit_reverse (unsigned a, int len) {
    a = ((a&0x55555555U)<<1) | ((a&0xAAAAAAAAU)>>1);
    a = ((a&0x33333333U)<<2) | ((a&0xCCCCCCCCU)>>2);
    a = ((a&0x0F0F0F0FU)<<4) | ((a&0xFF0F0F0FU)>>4);
    a = ((a&0x00FF00FFU)<<8) | ((a&0xFFFF00FFU)>>8);
    a = ((a&0x0000FFFFU)<<16) | ((a&0xFFFF0000U)>>16);
    return a >> (32-len);
}
void fft (bool is_inv, VT &in, VT &out, int N) {
    int bitlen = __lg(N), num = is_inv ? -1 : 1;
    for(int i = 0; i < N; ++i)
        out[bit_reverse(i, bitlen)] = in[i];
    for(int step = 2, mh = 1; step <= N; step <= 1, mh <= 1){
        for(int i = 0; i < mh; ++i){
            complex<T> wi = exp(complex<T>(0, i * num * pi / mh));
            for(int j = i, k = i + mh; j < N; j += step, k += step) {
                complex<T> u = out[j], t = wi * out[k];
                out[j] = u + t, out[k] = u - t;
            }
        }
    }
    for (int i = 0; is_inv && i < N; ++i)
        out[i] /= N;
}
};
int main () { // polynomial multiplication
    FFT<double> F; int n = 4;
    vector<complex<double>> a = {1, 2, 0, 0};
    vector<complex<double>> b = {2, 3, 0, 0};
    vector<complex<double>> a_fft(n), b_fft(n), ab_fft(n), ab(n);
    F.fft(0, a, a_fft, 4), F.fft(0, b, b_fft, 4);
    for (int i = 0; i < n; i++)
        ab_fft[i] = a_fft[i] * b_fft[i];
    F.fft(1, ab_fft, ab, n);
    for (auto p : ab)
        cout << int(p.real() + 1e-6) << " ";
    return 0;
}

```

```

int count;
void merge(vector<int>& arr, const int start, const int mid, const int end){
    vector<int> Left(arr.begin() + start , arr.begin() + mid + 1),
        Right(arr.begin() + mid + 1, arr.begin() + end + 1);
    int iL = 0, iR = 0;
    for(int i = start; i <= end; i++){
        if(iL == Left.size())        arr[i] = Right[iR++];
        else if(iR == Right.size())  arr[i] = Left[iL++];
        else if(Left[iL] <= Right[iR]) arr[i] = Left[iL++];
        }else{
            count += Left.size() - iL;
            arr[i] = Right[iR++];
        }
    }
}
void merge_sort(vector<int>& arr, const int start, const int end){
    if(start < end){
        int mid = (end + start) / 2;
        merge_sort(arr, start, mid);
        merge_sort(arr, mid + 1, end);
        merge(arr, start, mid, end);
    }
}
}

```