Comiendo

Autor del problema: Félix Moreno Peñarrubia (Miembro del comité científico)

Resumen del enunciado

Tenemos montones de galletas y hay que comerlas todas en unos cuantos pasos. En cada paso, podemos comer galletas de cada montón tal que de un montón comamos tantas como la suma de las comidas del resto.

Explicación de la solución

- La suma tiene que ser par y no puede haber ningún montón con más galletas que la suma del resto.
- Iterativamente nos comemos todas las galletas del montón con más galletas y del resto de manera que nos aseguremos que la condición anterior se siga cumpliendo.
- Cuando la condición de la suma ya no vaya a cumplirse, nos podemos comer todas las otras galletas en dos días.

Solución en C++

```
#include<bits/stdc++.h>

using namespace std;

typedef long long ll;
typedef vector<ll> vi;
typedef vector<vi> vvi;
typedef pair<ll, ll> ii;
typedef vector<ii> vii;

void solve() {
    int n;
    cin >> n;
    vi a(n);
    ll mx = 0;
    ll sum = 0;
    for(int i=0; i < n; ++i) {
         cin >> a[i];
         mx = max(mx, a[i]);
         sum += a[i];
    }
    if(sum%2 == 1 || 2*mx > sum) {
        cout << "NO" << endl;
        return;
    }
    vii b;
    for(int i=0; i < n; ++i) {
        b.emplace_back(a[i], i);
    }
    sort(b.begin(), b.end());
```

```
sort(a.begin(), a.end());
vi ra(n);
for(int i=0; i < n; ++i) {
    ra[b[i].second] = i;
}

vvi days;
int fp = 0;
int sp = n-1;
bool sumstate = false;
while(!sumstate && fp < sp-1) {
    ll mn = min(a[fp], a[sp]);
    vi nd(n);
    if(sum-2*mn <= 2*a[sp-1]) {
           mn = sum/2-a[sp-1];
           nd[fp] = mn;
           nd[sp] = mn;
           sum -= 2*mn;
           sumstate = true;
    }
    else if (mn == a[sp]) {
           nd[fp] = mn;
           nd[sp] = mn;
           sum -= 2*mn;
    }
    else {
           while(!sumstate && a[fp] + nd[sp] < a[sp]) {
                  if(sum - 2*a[fp] <= 2*a[sp-1]) {
                         mn = sum/2-a[sp-1];
                         nd[fp] = mn;
                         nd[sp] += mn;
                         sum -= 2*mn;
                         sumstate = true;
                  }
                  else {
                         nd[fp] = a[fp];
                         nd[sp] += a[fp];
                         sum -= 2*a[fp];
                         fp++;
                  }
           }
           if(!sumstate) {
                  if(sum - 2*(a[sp]-nd[sp]) <= 2*a[sp-1]) {
                         mn = sum/2-a[sp-1];
                         nd[fp] = mn;
                         nd[sp] += mn;
                         sum -= 2*mn;
                         sumstate = true;
                  }
                  else {
                         nd[fp] = a[sp]-nd[sp];
                         sum -= 2*(a[sp]-nd[sp]);
                         nd[sp] = a[sp];
                  }
```

```
                }
            }
            ll sum2 = 0;
            ll sumnd = 0;
            ll mxnd = 0;
            for(int i=0; i < n; ++i) {
                a[i] -= nd[i];
                sumnd += nd[i];
                mxnd = max(mxnd, nd[i]);
                sum2 += a[i];
            }
            days.push_back(nd);
            if(a[fp] == 0) fp++;
            if(a[sp] == 0) sp--;
        }
        bool nz = false;
        ll mx2 = 0;

        for(int i=0; i < n; ++i) {
            if(a[i] > 0) nz = true;
            mx2 = max(mx2, a[i]);
        }
        if(nz) days.push_back(a);

        cout << "SI" << endl;
        int d = days.size();
        cout << d << endl;
        for(int i=0; i < d; ++i) {
            for(int j=0; j < n; ++j) {
                cout << days[i][ra[j]];
                if(j < n-1) cout << " ";
            }
            cout << endl;
        }

}

int main() {
    int T;
    cin >> T;
    while(T--) {
        solve();
    }
}
```

Autor de la solución: Félix Moreno Peñarrubia (Miembro del comité científico)