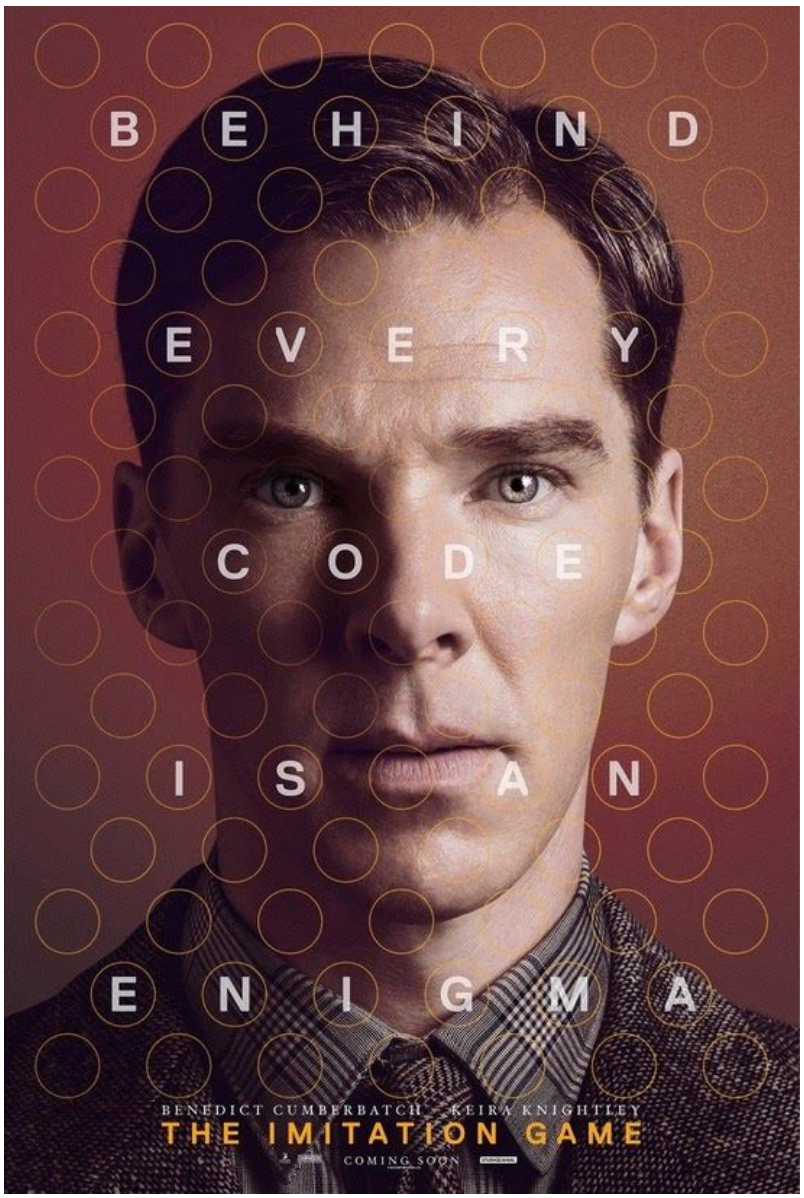
The Imitation Game Redefined



Are you paying attention? Good. If you are not listening carefully, you will miss things. Important things. I will not pause, I will not repeat myself, and you will not interrupt me. You think that because you're sitting where you are, and I am sitting where I am, that you are in control of what is about to happen. You're mistaken. I am in control, because I know things that you do not know. What I will need from you now is a commitment. You will listen closely, and you will not judge me until I am finished. If you cannot commit to this, then please leave the room. But if you choose to stay, remember you chose to be here. What happens from this moment forward is not my responsibility. It's yours. Pay attention. Solve what's ahead and be a part of what's called "The Imitation Game" .

Based on given input and output try to write a program based on your logic.

Input Format

- The first line contains an integer l1, the total number of elements.
- The second line contains l1 space-separated integers, an array .
- The third line contains an integer l2, the number of queries.
- The fourth line contains l2 space-separated integers, integers for the query.

Constraints

$$1 \leq l1 \leq 5 \times 10^5$$
$$1 \leq l2 \leq 5 \times 10^5$$
$$-2000 \leq \textit{any integer in input} \leq 2000$$

Output Format

For each query, print an integer on a new line.

Sample Input 0

```
3
4 5 6
3
-1 4 5
```

Sample Output 0

```
12
24
39
```

Explanation 0

.....

Sample Input 1

```
5
-1 3 -5 6 2
3
4 -2 3
```

Sample Output 1

```
27
21
30
```

Explanation 1

.....

Sample Input 2

```
5
-2 -1 0 1 2
4
1 -2 3 -4
```

Sample Output 2

```
7
7
10
10
```

Explanation 2

.....

Sample Input 3

```
10
23 -20 5 6 99 89 43 55 12 34
4
-43 50 22 -100
```

Sample Output 3

```
312
442
```

636
456

Sample Input 4

3
6 9 2
2
-3 -2

Sample Output 4

10

8

Expected Solution :

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

#define s(a) scanf("%d",&a)
#define sl(a) scanf("%lld",&a)
#define ss(a) scanf("%s",a)

#define MP          make_pair
#define PB          push_back
#define REP(i, n)    for(int i = 0; i < n; i++)
#define INC(i, a, b) for(int i = a; i <= b; i++)
#define DEC(i, a, b) for(int i = a; i >= b; i--)
#define CLEAR(a)     memset(a, 0, sizeof a)

using namespace std;

typedef long long          LL;
typedef unsigned long long ULL;
typedef vector<int>        VI;
typedef pair<int, int>     II;
typedef vector<II>         VII;

LL inp[500005];
```

```
LL cum[500005];
int main()
{
    int t=1;
    //s(t);
    while(t--)
    {
        int n,Q,q;
        LL add = 0;
        s(n);
        REP(i,n)
            sl(inp[i]);
        sort(inp,inp+n);
        cum[0] = inp[0];
        INC(i,1,n-1)
            cum[i] = cum[i-1]+inp[i];
        s(Q);
        while(Q--)
        {
            s(q);
            add+=q;
            int pos = lower_bound(inp,inp+n,-add)-inp;
            LL ans;
            if(pos>0)
                ans = (cum[n-1]-cum[pos-1]+add*(n-pos))-(cum[pos-1]+add*pos);
            else
                ans = (cum[n-1]+add*n);
            printf("%lld\n",ans);
        }
    }
    return 0;
}
```