```c
/// Final 2557-1: Array Sampling {Method 2: Struct]
/// Main idea: Since we need to know the length of each array so that
///   we can check whether a referred position is correct or not. To
///   organize the code in a way that a data array and its legnth are
///   together, we use a struct. Then, employ an array of this struct
///   to hold the entire data.
#include <stdio.h>
#include <limits.h>

struct array_pack {
    int D[1000];
    int length;
} typedef ArrayPack;

ArrayPack A[101];

int main() {
    int N;
    scanf("%d", &N);
    for(int i = 1; i <= N; ++i) {
        scanf("%d", &A[i].length);
        for(int j = 0; j < A[i].length; ++j) {
            scanf("%d", &A[i].D[j]);
        }
    }

    int sum = 0;
    int invalid = 0;
    int min = INT_MAX;
    int max = INT_MIN;

    while(1) {
        int p, q;
        scanf("%d", &p);
        if(p == 0)
            break;

        scanf("%d", &q);
        if(p > N) {
            ++invalid;
            continue;
        }

        if(q >= A[p].length) {
            ++invalid;
        } else {
            int val = A[p].D[q];
            sum += val;
            if(val > max)
                max = val;
            if(val < min)
                min = val;
        }
    }
    printf("%d\n%d\n%d\n%d", sum, max, min, invalid);

    return 0;
}
```