CodeShows Meeting Editorial

Let L be the minimum coordinate, and R be the maximum coordinate, where a person lives.

Then, we do not have to consider holding a meeting at coordinate x such that $x < L$ or $R < x$. That is, in such case the total energy is no less than when holding a meeting at coordinates L or R.

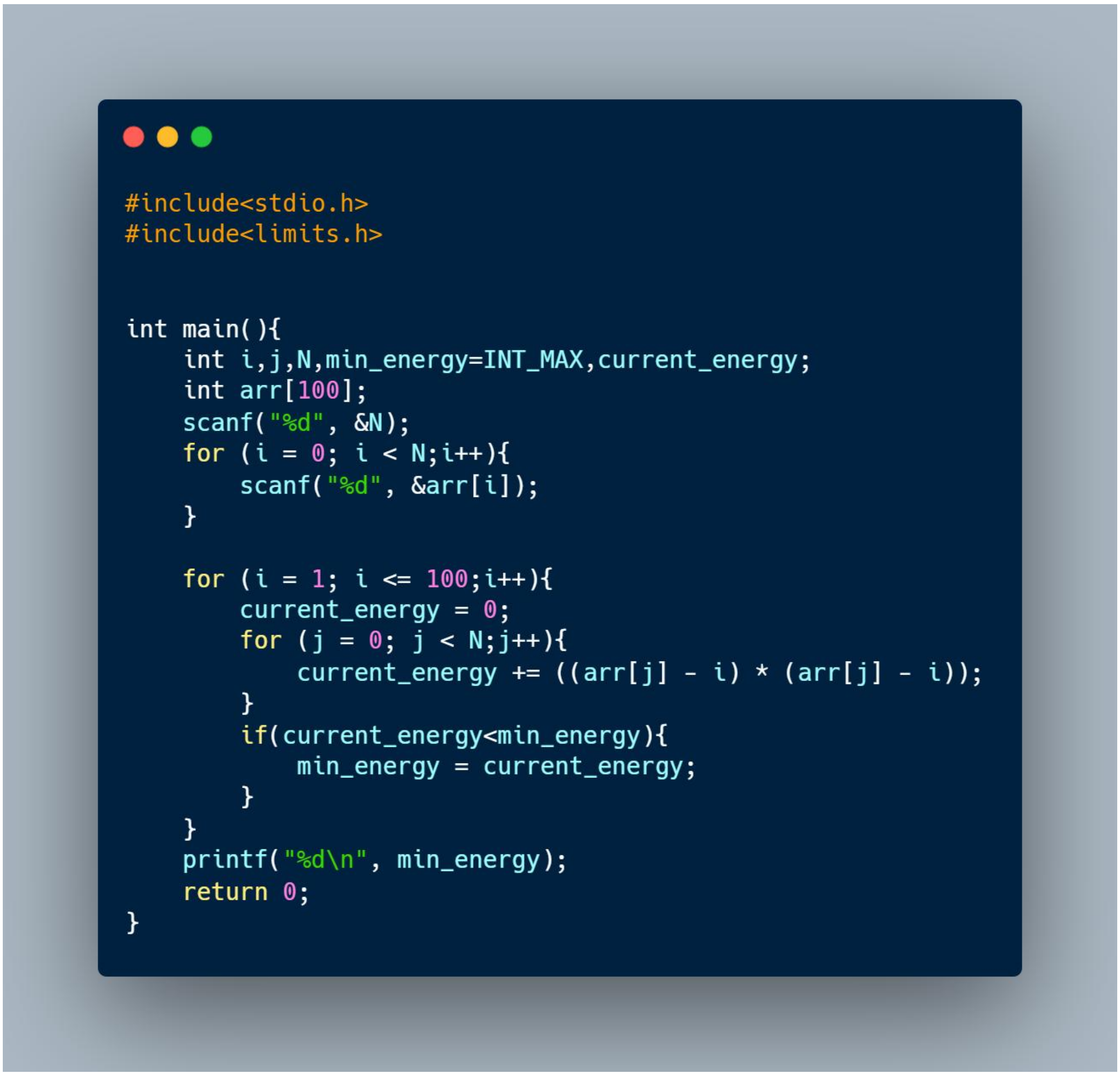
Strictly speaking, when holding a meeting at coordinate x which can be represented as $x = R + d$ $(d > 0)$ for example, the energy that i-th person will spend is:

$$(a_i - (R + d))^2 = ((R + d) - a_i)^2 = (R - a_i)^2 + 2d(R - a_i) + d^2,$$

And since the second term of the rightmost hand side is non-negative and the third term is positive, the energy spent is strictly greater than when holding a meeting at coordinate R. It is the same when holding a meeting at coordinate x such that $x < L$.

Based on this idea, we can decrease the number of coordinates we have to consider as students of meeting to at most 100 places. When a place of holding the meeting is fixed, the sum of points of stamina can be calculated in a total of O(N) operations. Therefore, the algorithm of looking up all the students left and finding the minimum sum of energy consumed is fast enough.

IDEAL SOLUTION :-



```c
#include<stdio.h>
#include<limits.h>


int main(){
    int i,j,N,min_energy=INT_MAX,current_energy;
    int arr[100];
    scanf("%d", &N);
    for (i = 0; i < N;i++){
        scanf("%d", &arr[i]);
    }

    for (i = 1; i <= 100;i++){
        current_energy = 0;
        for (j = 0; j < N;j++){
            current_energy += ((arr[j] - i) * (arr[j] - i));
        }
        if(current_energy<min_energy){
            min_energy = current_energy;
        }
    }
    printf("%d\n", min_energy);
    return 0;
}
```