야근 지수

Level 3

야근 지수  
회사원인 수민이는 많은 일이 쌓여 있습니다. 수민이는 야근을 최소화하기 위해 남은 일의 작업량을 숫자로 메기고, 일에 대한 야근 지수를 줄이기로 결정했습니다. 야근 지수는 남은 일의 작업량을 제곱하여 더한 값을 의미합니다. 수민이는 1시간 동안 남은 일 중 하나를 골라 작업량 1만큼 처리할 수 있습니다. 수민이의 퇴근까지 남은 N 시간과 각 일에 대한 작업량이 있을 때, noOvertime 함수를 제작하여 수민이의 야근 지수를 최소화 한 결과를 출력해 주세요. 예를 들어, N=4 일 때, 남은 일의 작업량이 [4, 3, 3] 이라면 야근 지수를 최소화하기 위해 일을 한 결과는 [2, 2, 2]가 되고 야근 지수는 22 + 22 + 22 = 12가 되어 12를 반환해 줍니다.

================================Ver.1=============================

#include<iostream>

#include<vector>

#include<algorithm>

#include<math.h>

using namespace std;

int noOvertime(int no,vector<int> works)

{

int answer = 0;

while(no > 0)

{

sort(works);

works.back() -= 1;

no--;

}

for(int i = 0; i < works.size(); i++)

{

answer += pow(works[i], 2);

}

return answer;

}

int main()

{

vector<int> works{4,3,3};

int testNo = 4;

int testAnswer = noOvertime(testNo, works);

cout<<testAnswer;

}

================================Ver.2=============================

#include<iostream>

#include<vector>

#include<algorithm>

#include<math.h>

using namespace std;

int noOvertime(int no,vector<int> works)

{

int answer = 0;

while(no > 0)

{

int tmpi = 0;

int max = 0;

for(int i = 0; i < works.size(); i++)

{

if(max < works[i])

{

max = works[i];

tmpi = i;

}

}

works[tmpi] -= 1;

no--;

}

for(int i = 0; i < works.size(); i++)

{

answer += (works[i] \* works[i]);

}

return answer;

}

int main()

{

vector<int> works{4,3,3};

int testNo = 4;

int testAnswer = noOvertime(testNo, works);

}

/\* 과정 설명

1번째는 정렬하면서 제일 큰 값을 빼주고 연산해서 답을 구하고

2번째는 그냥 큰 값을 바로 찾아서 뺴주고 연산을 한다

\*/