

PREPARE<sup>NEW</sup>

CERTIFY

COMPETE

Search



C\_24060121140112 ▾

[All Contests](#) > [Praktikum ASA Undip pertemuan 4](#) > [Bansos](#)

# Bansos

Problem

Submissions

Leaderboard

Discussions

author : yqisthi

Diketahui pemerintah Indonesia ingin memberikan Bantuan Sosial. Bantuan sosial berupa array dengan size **N** yang berisi berat beras **X** integer positif. Nantinya bansos ini akan dibagikan kepada **K** masyarakat. Untuk mencegah terjadinya korupsi, bantulah mengawasi kegiatan perintah dengan menghitung apakah jumlah pembagian tersebut dapat dibagikan secara merata.

Misalnya: masukan beras berupa arr = [3, 8, 5, 2, 6] dengan target banyak orang **K** = 3 Maka akan mengeluarkan "**Ya**", karena beras dapat dibagikan sebagai berikut [[3, 5], [8], [2, 6]]

## Input Format

- Baris pertama merupakan besaran array **N**
- Baris kedua merupakan elemen array **X**
- Baris ketiga merupakan banyaknya masyarakat **K**

## Constraints

- 1000 ms
- $1 \leq N \leq 1000$
- $1 \leq X \leq 1000$
- $1 \leq K \leq 1000$

## Output Format

- "Ya" atau "Tidak" apabila array bisa dibagi menjadi subset yang hasilnya sama

## Sample Input 0

```
5
3 8 5 2 6
3
```

## Sample Output 0

```
Ya
```

[f](#) [t](#) [in](#)Contest ends in 15 days

Submissions: 1

Max Score: 100

Rate This Challenge:

☆☆☆☆☆

[More](#)



```
1 #include <stdio.h>
2 #include <stdbool.h>
3
4 #define MAXN 1000
5
6 int n;
7 int nums[MAXN];
8 int k;
9
10 int sum(int *nums, int n) {
11     int s = 0;
12     for (int i = 0; i < n; i++) {
13         s += nums[i];
14     }
15     return s;
16 }
17
18 int max(int *nums, int n) {
19     int m = nums[0];
20     for (int i = 1; i < n; i++) {
21         if (nums[i] > m) {
22             m = nums[i];
23         }
24     }
25     return m;
26 }
27
28 bool dfs(int *nums, int n, int k, int target, int *dp, int i) {
29     if (i == n) {
30         for (int j = 1; j < k; j++) {
31             if (dp[j] != dp[0]) {
32                 return false;
33             }
34         }
35         return true;
36     }
37
38     for (int j = 0; j < k; j++) {
39         dp[j] += nums[i];
40         if (dp[j] <= target && dfs(nums, n, k, target, dp, i + 1)) {
41             return true;
42         }
43         dp[j] -= nums[i];
44         if (dp[j] == 0) {
45             break;
46         }
47     }
48
49     return false;
50 }
51
52 bool canPartitionKSubsets(int *nums, int n, int k) {
53     int target = sum(nums, n);
54     if (target % k != 0 || max(nums, n) > target / k) {
55         return false;
56     }
57
58     int dp[k];
59     for (int i = 0; i < k; i++) {
60         dp[i] = 0;
61     }
62
63     return dfs(nums, n, k, target / k, dp, 0);
64 }
65
66 int main() {
67     scanf("%d", &n);
68     for (int i = 0; i < n; i++) {
69         scanf("%d", &nums[i]);
70     }
71     scanf("%d", &k);
```

```
72 |
73 |   if (canPartitionKSubsets(nums, n, k)) {
74 |       printf("Ya\n");
75 |   } else {
76 |       printf("Tidak\n");
77 |   }
78 |
79 |   return 0;
80 | }
```

Line: 1 Col: 1

[Upload Code as File](#)

Test against custom input

Run Code

Submit Code

[Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) |