



Arrays-Sum Of Two Arrays

Take as input N, the size of array. Take N more inputs and store that in an array. Take as input M, the size of second array and take M more inputs and store that in second array. Write a function that returns the sum of two arrays. Print the value returned.

Example 1

Input

4
1 0 2 9
5
3 4 5 6 7

Output

3, 5, 5, 9, 6, END ✓✓

Explanation

Sum of [1, 0, 2, 9] and [3, 4, 5, 6, 7] is [3, 5, 5, 9, 6] and the first digit represents carry over, if any (0 here).

$$\begin{array}{r} 601 \\ 1029 \\ 34567 \\ \hline 35596 \end{array}$$

$$\text{Sum} = 9 + 7 + 10 = 16$$

Sum < 10

$$\begin{array}{r} 0 \quad i \\ \begin{array}{|c|c|} \hline 2 & 2 \\ \hline 5 & 5 \\ \hline \end{array} \\ 0 \\ \hline 7 \quad 7 \end{array}$$

$$\begin{array}{r} \downarrow \downarrow \downarrow \quad 001 \\ 2734567 \\ \hline 2735596 \end{array}$$

carry = 0
carry = sum / 10
= 9 / 10 = 0

```
while(i >= 0 & j >= 0) {  
    int sum = arr1[i] + arr2[j] + carry  
    // add (sum % 10) ✓  
    carry = sum / 10  
    i--  
    j--  
}
```

int carry = 0

$$\begin{array}{r} 1 \\ 88 \\ 34 \\ \hline 122 \end{array}$$

7 | 2 | 5 | 4 | 7 | 2

$$\begin{array}{r} 1029 \\ 273498 \\ \hline 274527 \end{array}$$

2, 7, 4, 5, 4, 7

$$\begin{array}{r} i \quad i \\ \begin{array}{|c|c|} \hline 3 & 2 \\ \hline 9 & 9 \\ \hline \end{array} \\ j \quad j \end{array}$$

$$\begin{array}{|c|c|c|} \hline 1 & 3 & 1 \\ \hline \end{array}$$

$$2 + 4 = 6 + 6 = 12 + 6 = 18 = 15$$

The coding blocks members went to the success party of their first ever online boot-camp at Murthal. They ordered P number of paranthas. The stall has L cooks and each cook has a rank R. A cook with a rank R can cook 1 parantha in the first R minutes 1 more parantha in the next 2R minutes, 1 more parantha in 3R minutes and so on (he can only cook a complete parantha) (For example if a cook is ranked 2, he will cook one parantha in 2 minutes one more parantha in the next 4 mins and one more in the next 6 minutes hence in total 12 minutes he cooks 3 paranthas. In 13 minutes also he can cook only 3 paranthas as he does not have enough time for the 4th parantha). Calculate the minimum time needed to cook all the paranthas.

$$\begin{array}{r} 2+4+6+8 \\ 3+5+7 \\ \hline 18 \end{array}$$

$$1 + 2 + 3 + 4 + 5 = 15$$

$$1 + 2 + 3 + 4 = 10$$

$$2 \quad 4 \quad 6 = 12 \quad 20$$

P = 10

$$\begin{array}{r} 4 \leftarrow 1 \rightarrow 4 \\ 2 \leftarrow 2 \rightarrow 3 \\ 2 \leftarrow 3 \rightarrow 2 \\ 1 \leftarrow \rightarrow 2 \\ \hline 9 \end{array}$$

$$\begin{array}{r} 1 \quad 2 \quad 3 \quad 4 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 5 \quad 4 \quad 3 \quad 2 \\ \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \\ 5 \quad 3 \quad 2 \quad 2 \\ \hline 3 \end{array}$$

$$\begin{array}{r} 25 \\ 15 \\ 13 \\ \hline 55 \end{array}$$

$$\begin{array}{r} 1 \quad 2 \quad 3 \quad 4 \\ \hline 4 \quad (1+2+3+4) = 10 \\ \hline \text{Row} (P \times CPH) \\ \hline 2 \end{array}$$

$$lo = 0$$
$$hi = \frac{\text{high Rank} \times (P \times CPH)}{2}$$

while (lo <= hi) {

$$mid = \frac{(lo + hi)}{2}$$

if (is it possible (cook, parnth, time)) {

$$ans = mid$$
$$hi = mid - 1$$

else {

$$lo = mid + 1$$

54

$$1 + 2 \times 1 + 3 \times 1 + 4 \times 1$$

while (i < cook) {

if (curr time + parnth * Row(i) <= time) {

$$\text{curr time} = (\text{curr time} + \text{parnth} \times \text{Row}(i))$$

parnth++

if (curr time <= 0 parnth > 0