When your worst developers decide to pair program

Productivity

After reading a recent research from a famous university proving people work best in couples, your CEO decides to assign a very important task to the new VP of the company.
The VP has to find if for a given project, there is such a couple of employees at your company that can be as productive together as the project requires. Everybody knows the VPs are lazy but very smart so they assign this task to you.

From the standard input you get a number k that is how much productivity a given project requires. Then separated by a semicolon (;) you get a list of employees' productivities represented by integer numbers.
In the standard output you have to print true or false depending if there are 2 employees from the given ones which productivity is equal to the required for the project

Example:
Input: 15;5,3,0,7
Answer: false
Explanation: no 2 numbers from the list sum to 15. It's a different question why we keep a person with 0 productivity at work.

Example:
Input: 12;2,5,7,1
Answer: true
Explanation: there are 2 numbers which sum is 12, those are 5 and 7
0 <= k <= 2^32
0 <= employee productivity <= 2^31

Here are example code snippets for input read for GoLang and Java. If you write a more optimal one, please let me know to update the examples.

GoLang

```go
package main

import (
	"bufio"
	"fmt"
	"log"
	"os"
	"strconv"
	"strings"
)

func main() {
	input := readInput()
	target, _ := strconv.Atoi(input[0])
	fmt.Println(productivity(toInts(strings.Split(input[1], ",")), target))
}

func readInput() []string {
	scanner := bufio.NewScanner(os.Stdin)

	const maxCapacity = 20000000
	buf := make([]byte, maxCapacity)
	scanner.Buffer(buf, maxCapacity)

	var res string
	for scanner.Scan() {
		res += scanner.Text()
	}

	if err := scanner.Err(); err != nil {
		log.Fatal(err)
	}

	return strings.Split(res, ";")
}

func toInts(nums []string) (result []int) {
	for _, num := range nums {
		x, _ := strconv.Atoi(num)
		result = append(result, x)
	}
```

```
	return
}

func productivity(nums []int, target int) bool {
	// fill code
}
```

Java

```
package productivity;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

public class Productivity {

    public static void main(String[] args) throws IOException {
        BufferedReader bi = new BufferedReader(new InputStreamReader(System.in));
        List<Integer> arr = new ArrayList<>();
        String[] input = bi.readLine().split(";");
        long k = Long.parseLong(input[0].trim());
        String[] elements = input[1].split(",");
        for (String el : elements) {
            arr.add(Integer.parseInt(el.trim()));
        }
        System.out.println(productivity(arr, k));
    }


    public static boolean productivity(List<Integer> nums, long target) {
        // TODO implement
    }
}
```