

A person wants to determine the most expensive computer keyboard and USB drive that can be purchased with a give budget. Given price lists for keyboards and USB drives and a budget, find the cost to buy them. If it is not possible to buy *both* items, return  $-1$ .

**Example**

```
b = 60
keyboards = [40, 50, 60]
drives = [5, 8, 12]
```

The person can buy a **40 keyboard + 12 USB drive = 52**, or a **50 keyboard + 8 USB drive = 58**. Choose the latter as the more expensive option and return **58**.

**Function Description**

Complete the *getMoneySpent* function in the editor below.

*getMoneySpent* has the following parameter(s):

- *int keyboards[n]*: the keyboard prices
- *int drives[m]*: the drive prices
- *int b*: the budget

**Returns**

- *int*: the maximum that can be spent, or  $-1$  if it is not possible to buy both items

**Input Format**

The first line contains three space-separated integers *b*, *n*, and *m*, the budget, the number of keyboard models and the number of USB drive models.  
The second line contains *n* space-separated integers *keyboard[i]*, the prices of each keyboard model.  
The third line contains *m* space-separated integers *drives*, the prices of the USB drives.

**Constraints**

- $1 \leq n, m \leq 1000$
- $1 \leq b \leq 10^6$
- The price of each item is in the inclusive range  $[1, 10^6]$ .

**Sample Input 0**

```
10 2 3
3 1
5 2 8
```

### Sample Output 0

9

### Explanation 0

Buy the **2<sup>nd</sup>** keyboard and the **3<sup>rd</sup>** USB drive for a total cost of  **$8 + 1 = 9$** .

### Sample Input 1

```
5 1 1
4
5
```

### Sample Output 1

-1

### Explanation 1

There is no way to buy one keyboard and one USB drive because  **$4 + 5 > 5$** , so return  **$-1$** .