

**Algorithm:**

A construction worker called Antony has many bricks of height 1 and different lengths, and he is now trying to build a wall of width  $w$  and height  $h$ . Since the construction worker previously known as Antony knows that the subset sum problem is NP-hard, he does not try to optimize the placement but he just lays the bricks in the order they are in his pile and hopes for the best. First he places the bricks in the first layer, left to right; after the first layer is complete he moves to the second layer and completes it, and so on. He only lays bricks horizontally, without rotating them. If at some point he cannot place a brick and has to leave a layer incomplete, then he gets annoyed and leaves. It does not matter if he has bricks left over after he finishes.

Yesterday the construction worker previously known as Antony got really annoyed when he realized that he could not complete the wall only at the last layer, so he tore it down and asked you for help. Can you tell whether the construction worker previously known as Antony will complete the wall with the new pile of bricks he has today?

**Input Format:**

The first line contains three integers  $h, w, n$ . The height of the wall, the width of the wall, and the number of bricks respectively. The second line contains ' $n$ ' integers  $x[i]$ , the length of each brick.

**Output Format:**

Output 'YES' if the construction worker previously known as Antony will complete the wall, and 'NO' otherwise.

**Sample Input:**

```
2 10 7
5 5 5 3 5 2 2
```

**Sample Output:**

```
NO
```

**Solution:**

```
h,w,_=map(int,input().split())
bricks=list(map(int,input().split()))
width,complete=w,False
for b in bricks:
    if not complete:
        if b > w:
```

```
        complete=-1
    else:
        w-=b
        if w==0:
            w=width
            h-=1
            complete=(h == 0)
if complete == -1:
    print("NO")
else:
    print("YES")
```