

24 June 2023

<https://my.newtonschool.co/playground/code/5zfezaoj3b0i>

Sample Input

2

10 14
1 2 3 4 5 6 7 8 9 10

Sample Output

1

0

Explanation:

Testcase 1: arr[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10} and sum =

$$\begin{aligned} 14 - 1 &= 13 \\ 14 - 2 &= 12 \\ 14 - 3 &= 11 \\ &\vdots \\ 14 - 10 &= 4 \end{aligned} \quad [4 \dots 13]$$

for (num : arr)
{
 complement = sum - num;
 if (hs.contains(complement))
 return 1;
}

return 0;

10
11
12
13
14
15
16
17
18
19
20
21

```
Scanner sc = new Scanner(System.in);
int testCases = sc.nextInt();

for(int i = 1; i <= testCases; i++)
{
    int arrSize = sc.nextInt();
    int sum = sc.nextInt();
    int arr[] = new int[arrSize];

    for(int j = 0; j < arrSize; j++)
        arr[j] = sc.nextInt();
    System.out.println(pairFound(arr, arrSize, sum));
}
```

arr[10]

0	1	2	3	4	5	6	7	8	9
1	2	3	4	5	6	7	8	9	10

arrSize = 10 sum = 14

hSet = {1, 2, 3, 4, ...}

```
static int pairFound(int arr[], int arrSize, int sum)
{
    HashSet<Integer> hSet = new HashSet<>();
    for(int i = 0; i < arrSize; i++)
    {
        1. int complement = sum - arr[i];
        2. if (hSet.contains(complement) == true)
            return 1;
        3. hSet.add(arr[i]);
    }

    return 0;
}
```

i = 0 & 23
complement = sum - arr[i]
= 14 - arr[i] = 14 - 1 = 13
= 14 - 2 = 12
= 14 - 3 = 11
= 14 - 4 = 10
⋮

Stack:

Linear Data Structure

LIFO i.e. Last In First Out
or

FILO i.e. First In Last Out

2 Operations:

>Push

>Pop

Some real life examples of Stack



⋮
40
30 50
20
10

	Push	Pop
1	10	—
2	20	—
3	30	—
4	—	✓
5	50	—
6	40	—
7	Display: 40, 50, 20, 10	
8	—	✓
9	—	✓
Display: 20, 10		

<https://yongdanielliang.github.io/animation/web/Stack.html>

Stack Operations:



Push item = 10

if (top == maxSize - 1)

 Sop (Stack full. Cannot push)

top = top + 1

s[top] = item;

or
s[++top] = item;



top = 1 s

Pop:

if (top == -1)

 Sop (Stack is empty. Cannot pop)

 return -1

else

 return s[top--];

Peck:

if (top == -1)

 Sop ("empty. cannot peck")

 return -1

else

 return s[top];

Traverse: 40, 30, 20, 10

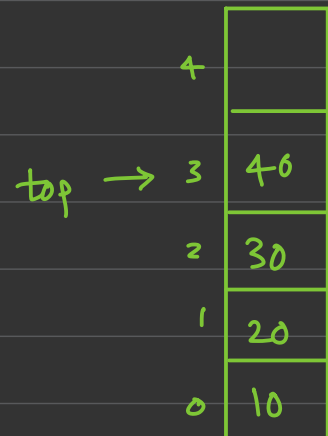
if (top == -1)

 Sop ("Stack empty. Cannot traverse")

else

 for (int i = top; i >= 0; i--)

 Sop(s[i])



top = 3 s

×