

Kids with the Greatest Number of Candies

Given the array `candies` and the integer `extraCandies`, where `candies[i]` represents the number of candies that the *ith* kid has.

For each kid check if there is a way to distribute `extraCandies` among the kids such that he or she can have the **greatest** number of candies among them. Notice that multiple kids can have the **greatest** number of candies.

Solution:

```
class Solution(object):

    def kidsWithCandies(self, candies, extraCandies):

        """
        :type candies: List[int]
        :type extraCandies: int
        :rtype: List[bool]
        """

        sol_array=[]

        max_value = max(candies)

        for i in candies:

            if i + extraCandies >= max_value:

                sol_array.append(True)

            else:

                sol_array.append(False)

        return sol_array
```

Success Details >

Runtime: **24 ms**, faster than **79.80%** of Python online submissions for Kids With the Greatest Number of Candies.

Memory Usage: **12.6 MB**, less than **86.74%** of Python online submissions for Kids With the Greatest Number of Candies.

Next challenges:

[Insert Delete GetRandom O\(1\) - Duplicates allowed](#)

[Add to Array-Form of Integer](#)

[Largest Unique Number](#)

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
a few seconds ago	Accepted	24 ms	12.6 MB	python
a few seconds ago	Accepted	24 ms	12.6 MB	python

```
1 class Solution(object):
2     def kidsWithCandies(self, candies, extraCandies):
3         """
4         :type candies: List[int]
5         :type extraCandies: int
6         :rtype: List[bool]
7         """
8         sol_array=[]
9         max_value = max(candies)
10        for i in candies:
11            if i + extraCandies >= max_value:
12                sol_array.append(True)
13            else:
14                sol_array.append(False)
15        return sol_array
```

Your previous code was restored from your local storage. [Reset to default](#)

Testcase Run Code Result Debuqer

Accepted Runtime: 28 ms

Your input

[2,3,5,1,3]
3

Output

[true,true,true,false,true]

Diff

Expected

[true,true,true,false,true]