1195 Distribute Candies to People (link).

Description

We distribute some number of `candies,` to a row of `n = num_people` people in the following way:

We then give 1 candy to the first person, 2 candies to the second person, and so on until we give `n` candies to the last person.

Then, we go back to the start of the row, giving `n + 1` candies to the first person, `n + 2` candies to the second person, and so on until we give `2 * n` candies to the last person.

This process repeats (with us giving one more candy each time, and moving to the start of the row after we reach the end) until we run out of candies. The last person will receive all of our remaining candies (not necessarily one more than the previous gift).

Return an array (of length `num_people` and sum `candies`) that represents the final distribution of candies.

Example 1:

```
Input: candies = 7, num_people = 4
Output: [1,2,3,1]
Explanation:
On the first turn, ans[0] += 1, and the array is [1,0,0,0].
On the second turn, ans[1] += 2, and the array is [1,2,0,0].
On the third turn, ans[2] += 3, and the array is [1,2,3,0].
On the fourth turn, ans[3] += 1 (because there is only one candy left), and the final
```

Example 2:

```
Input: candies = 10, num_people = 3
Output: [5,2,3]
Explanation:
On the first turn, ans[0] += 1, and the array is [1,0,0].
On the second turn, ans[1] += 2, and the array is [1,2,0].
On the third turn, ans[2] += 3, and the array is [1,2,3].
On the fourth turn, ans[0] += 4, and the final array is [5,2,3].
```

Constraints:

- 1 <= candies <= 10^9
- 1 <= num_people <= 1000

(scroll down for solution)

Solution

Language: cpp

Status: Accepted

```cpp
#include <vector>
using namespace std;

class Solution {
public:
    vector<int> distributeCandies(int candies, int num_people) {
        vector<int> result(num_people, 0);
        int candy = 1;
        int idx = 0;

        while (candies > 0) {
            if (candies >= candy) {
                result[idx % num_people] += candy;
                candies -= candy;
            } else {
                result[idx % num_people] += candies;
                candies = 0;
            }

            candy++;
            idx++;
        }

        return result;
    }
};
```