Largest Triple Products

You're given a list of n integers `arr[0..(n-1)]`. You must compute a list `output[0..(n-1)]` such that, for each index `i` (between `0` and `n-1`, inclusive), `output[i]` is equal to the product of the three largest elements out of `arr[0..i]` (or equal to `-1` if `i < 2`, as `arr[0..i]` then includes fewer than three elements).

Note that the three largest elements used to form any product may have the same values as one another, but they must be at different indices in `arr`.

Signature

```
int[] findMaxProduct(int[] arr)
```

Input

n is in the range [1, 100,000].
Each value `arr[i]` is in the range [1, 1,000].

Output

Return a list of n integers `output[0..(n-1)]`, as described above.

Example 1

n = 5
`arr` = [1, 2, 3, 4, 5]
`output` = [-1, -1, 6, 24, 60]

The 3rd element of `output` is `3*2*1 = 6`, the 4th is `4*3*2 = 24`, and the 5th is `5*4*3 = 60`.

Example 2

n = 5
`arr` = [2, 1, 2, 1, 2]
`output` = [-1, -1, 4, 4, 8]

The 3rd element of `output` is `2*2*1 = 4`, the 4th is `2*2*1 = 4`, and the 5th is `2*2*2 = 8`.