



## Monk and Multiplication

**Link:** <https://www.hackerearth.com/practice/data-structures/trees/heapspriority-queues/practice-problems/algorithm/monk-and-multiplication/>

**Solution:**

C++	<a href="https://ideone.com/UxKsGI">https://ideone.com/UxKsGI</a>
Java	<a href="https://ideone.com/uYfYEP">https://ideone.com/uYfYEP</a>
Python	<a href="https://ideone.com/nZ5GqT">https://ideone.com/nZ5GqT</a>

**Tóm tắt đề:** Monk đang tìm hiểu về hàng đợi ưu tiên. Cho anh ta một mảng số nguyên đối với mỗi phần tử khi thêm vào hàng đợi ưu tiên, anh ta muốn tìm phần tử lớn nhất, nhì và ba. Hai số có thể cùng giá trị với nhau nhưng khác nhau về index.

Hai phần tử đầu tiên khi thêm vào hàng đợi không đủ lấy 3 số lớn nhất, nhì và ba thì in ra -1.

**Input:**

Dòng đầu tiên chứa số N là số lượng phần tử trong mảng A.  $1 \leq N \leq 100000$ .

N dòng tiếp theo chứa các phần tử trong mảng A.  $0 \leq A[i] \leq 1000000$ .

**Output**

Với mỗi phần tử khi thêm vào thì in ra số lớn nhất \* số lớn nhì \* số lớn 3. Không có đủ 3 số thì in ra -1.

5	-1
1 2 3 4 5	-1
	6
	24
	60

**Giải thích:**

Thêm 1 vào hàng đợi ưu tiên  $\rightarrow$  in ra -1.

Thêm 2 vào hàng đợi ưu tiên  $\rightarrow$  in ra -1.

Thêm 3 vào hàng đợi ưu tiên  $\rightarrow$  in ra  $3 * 2 * 1 = 6$ .

Thêm 4 vào hàng đợi ưu tiên  $\rightarrow$  in ra  $4 * 3 * 2 = 24$ .

Thêm 5 vào hàng đợi ưu tiên  $\rightarrow$  in ra  $5 * 4 * 3 = 60$ .

**Hướng dẫn giải:** Bỏ hết các phần tử vào mảng.

Lần lượt duyệt qua các phần tử và bỏ vào hàng đợi ưu tiên. Mỗi lần bỏ phần tử mới vào thì lấy top 3 để nhân với nhau và in ra kết quả. Nếu như số lượng phần tử trong hàng đợi ưu tiên nhỏ hơn 3 thì ta xuất ra -1.

**Độ phức tạp:**  $O(N \log N)$  với N là số lượng phần tử trong mảng.

Big-O Coding