International Islamic University Chittagong

Department of Computer Science and Engineering

Course Title: Competitive Programming II

Course Code: CSE-2430

Final Term Exam

SET A

A(7 marks)

Time Limit: 1 second
Memory Limit: 64 MB

A palindromic number or numeral palindrome is a 'symmetrical' number like 19891, that remains the same when its digits are reversed. In this problem you will be given an integer, you have to say whether the number is a palindromic number or not.

Input:
Input starts with an integer T (≤ 20000), denoting the number of test cases.

Each case starts with a line containing an integer n ($0 \leq n < 1e9$).

Output:
For each case, print the case number and `Yes` if n is palindromic, otherwise print `No`.

Example:

Input
5 1 21 16161 523125 0
Output
Case 1: Yes Case 2: No Case 3: Yes Case 4: No Case 5: Yes

B(8 marks)

Time Limit: 1 second
Memory Limit: 512 MB

You have two coin piles containing a and b coins. On each move, you can either remove one coin from the left pile and two coins from the right pile, or two coins from the left pile and one coin from the right pile.

Your task is to efficiently find out if you can empty both the piles.

Input:

The first input line has an integer t: the number of tests.

After this, there are t lines, each of which has two integers a and b: the numbers of coins in the piles.

Output:
For each test, print "YES" if you can empty the piles and "NO" otherwise.

Constraints

- $1 \leq t \leq 1e5$
- $0 \leq a,b <= 1e9$

Example:

Input
3 2 1 2 2 3 3
Output
YES NO YES

C(8 marks)

Time Limit: 1 second
Memory Limit: 256 megabytes

Bachgold problem is very easy to formulate. Given a positive integer n represent it as a sum of **maximum possible** number of prime numbers. One can prove that such representation exists for any integer greater than 1.

Recall that integer k is called prime if it is greater than 1 and has exactly two positive integer divisors — 1 and k.

Input:
The only line of the input contains a single integer n ($2 \le n \le 100\,000$).

Output:

The first line of the output contains a single integer k — maximum possible number of primes in representation.

The second line should contain k primes with their sum equal to n. You can print them in any order. If there are several optimal solution, print any of them.

Example:

Input
6
Output
3 2 2 2

D(7 marks)

Time Limit: 2 second
Memory Limit: 256 megabytes

You are given two arrays of integers *a* and *b*. For each element of the second array *bj* you should find the number of elements in array *a* that are less than or equal to the value *bj*.

Input:

The first line contains two integers *n*, *m* ($1 \le n, m \le 2 \cdot 1e5$) — the sizes of arrays *a* and *b*.

The second line contains *n* integers — the elements of array *a* ($-1e9 \le ai \le 1e9$).

The third line contains *m* integers — the elements of array *b* ($-1e9 \le bj \le 1e9$).

Output:
Print *m* integers, separated by spaces: the *j*-th of which is equal to the number of such elements in array *a* that are less than or equal to the value *bj*.
Example:

Input
5 4 1 3 5 7 9 6 4 2 8
Output
3 2 1 4