Problem J. Disks and Gems

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	1024 mebibytes

Monocarp is a jeweler. Recently, he received an extremely weird order — the customer wants a set of metal disks encrusted with gems according to very specific rules.

Every disk is made of one of three materials — copper, silver, or gold. Additionally, every disk is divided into n sectors, numbered from 1 to n in clockwise order. These sectors are encrusted with gems of k types: every sector is either encrusted with exactly one gem, or is not encrusted at all (in that case, that sector is called empty).

Two disks a and b *have the same ornament* if the following condition is met:

- for every sector i of the disk a that is encrusted with a gem, the sector i of the disk b is also encrusted with the same type of gem, and vice versa.

Two disks are *equal* if they are made from the same material and have the same ornament.

For every disk in the set Monocarp will send to the customer, he creates it in the following way:

- first, he chooses one of three materials for the disk and forges a disk of that material with n empty sectors;
- then he encrusts two sectors of the disk with gems of type 1. The first sector he encrusts can be any empty sector; however, the second sector is always the **next empty sector** in clockwise order;
- then he encrusts two sectors of the disk with gems of type 2, choosing them in the same way;
- then he encrusts two sectors of the disk with gems of type 3, choosing them in the same way;
- and so on, until for each of the k types of gems, exactly two sectors are encrusted with gems of that type. Exactly $n - 2k$ sectors of the disk will be empty.

For example, suppose $n = 5$, $k = 2$. One possible way of creating a disk is:

- Monocarp forges a silver disk with n empty sectors;
- then, he chooses to encrust the 4-th sector with a gem of type 1. The next empty sector in clockwise order is the 5-th sector, so he also encrusts that sector with a gem of type 1;
- then, he chooses to encrust the 3-rd sector with a gem of type 2. The next empty sector in clockwise order is the 1-st sector (since both the 4-th and the 5-th sectors are already encrusted), so he also encrusts that sector with a gem of type 2.

Monocarp wants to assemble a set of disks such that:

- no two disks have the same ornament (this also implies that no two disks in the set are equal);
- the number of disks Monocarp makes is the maximum possible.

Your task is to calculate the number of different sets Monocarp can assemble. Two sets are equal if and only if for every disk in the first set, there is an equal disk in the second set, and vice versa.

Input

The first line contains one integer t ($1 \le t \le 3 \cdot 10^5$).

Each test case consists of one line containing two integers n and k ($2 \le n \le 9 \cdot 10^8$; $1 \le k \le \frac{n}{2}$).

Output

For each test case, print one integer — the number of different sets Monocarp can assemble, taken modulo 998 244 353.

Example

standard input	standard output
8 4 1 4 2 5 2 10 2 10 4 10 5 1337 42 888888888 8	81 81 14348907 412733925 572390210 572390210 359416572 358909282

Note

In the first test case, Monocarp can make at most 4 disks:

- with ornament $[0, 0, 1, 1]$ (where 0 represents an empty sector and 1 represents a sector with a gem of type 1);
- with ornament $[1, 0, 0, 1]$;
- with ornament $[0, 1, 1, 0]$;
- with ornament $[1, 1, 0, 0]$.

For each of these ornaments, there are 3 ways to choose the material, so the answer is $3^4 = 81$.

Note that the ornament $[1, 0, 1, 0]$ is impossible:

- if Monocarp chooses the 1-st sector and encrusts it with a gem of type 1, the next empty sector will be the 2-nd sector, so the resulting ornament will be $[1, 1, 0, 0]$;
- if Monocarp chooses the 3-rd sector and encrusts it with a gem of type 1, the next empty sector will be the 4-th sector, so the resulting ornament will be $[0, 0, 1, 1]$.