
Problem A. Suite de Fibonacci

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 256 megabytes

Fibonacci, grand mathématicien du XII-XIII siècles, a une suite à son nom appelée la suite de Fibonacci. Elle est définie récursivement de la manière suivante.

Si $f(n)$ représente le n -ième nombre de la suite. Alors $f(n) = f(n-1) + f(n-2)$ pour $n \geq 2$ avec comme cas de base $f(0) = 0$ et $f(1) = 1$.

Ton but est de calculer $f(n)$

Input

Il y a une seule ligne d'entrée, contenant un entier n ($0 \leq n \leq 20$)

Output

La sortie doit contenir une seule ligne contenant l'entier $f(n)$.

Example

standard input	standard output
3	2

Problem B. Est-ce une puissance de k?

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Tu as un but très simple, pour deux entier n et x tu dois déterminer si n est une puissance de x . Cet à dire, est-ce qu'il existe un entier k tel que $x^k = n$.

Input

La première ligne contient un entier t ($1 \leq t \leq 100$) — le nombre de test. Il y aura ensuite t lignes contenant chacunes deux entier n, x ($1 \leq x \leq n \leq 10^9$)

Output

Tu dois imprimer t lignes. Sur la i -ème tu dois dire "YES" si dans i -ème test n est une puissance de x et "NO" sinon.

Example

standard input	standard output
3	YES
2 8	NO
10 20	YES
3 3	

Note

Ta sortie est "case insensitive" c'est à dire que tu peux imprimer "YES", "Yes", "yes", ...

Problem C. Jeu de Nim II

Input file: **standard input**
Output file: **standard output**
Time limit: 4 seconds
Memory limit: 256 megabytes

Comme vous l'avez vu en cours, le jeu de Nim fonctionne de la manière suivante. Il y a n allumettes en ligne. Chaque joueur, à tour de rôle, va prendre une, deux ou trois allumettes. Celui qui prend la dernière allumette a perdu.

Cependant pour rendre le jeu plus intéressant, nous allons introduire une variante. Les deux joueurs peuvent maintenant enlever une, trois, ou quatre, allumettes.

Alice et Bob jouent au jeu, Alice commence. Ton but est de déterminer qui gagnera s'ils jouent optimalement.

Input

L'entrée contient une ligne avec un entier n ($1 \leq n \leq 40$) — le nombre d'allumettes

Output

La sortie doit contenir un seul mot "ALICE" si Alice gagne en jouant optimalement, "BOB" sinon.

Examples

standard input	standard output
3	BOB
4	ALICE

Problem D. Vacances familiales

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 256 megabytes

Dans l'avion de retour du Vietnam, ta petite sœur joue à côté de toi avec la tour de Hanoï qu'elle a achetée à l'aéroport. Bien qu'elle s'en sorte assez bien, tu veux jouer au Puissance 4 avec elle, alors tu as décidé de l'aider.

La Tour de Hanoï est composée de trois piles (gauche, milieu et droite) et de n disques de différentes tailles. Initialement, la pile de gauche contient tous les disques, classés par ordre croissant de taille de haut en bas. L'objectif est de déplacer tous les disques vers la pile de droite en utilisant la pile du milieu. À chaque mouvement, tu peux déplacer le disque du dessus d'une pile vers une autre pile. De plus, il n'est pas permis de placer un disque plus grand sur un disque plus petit.

Pour jouer à Puissance 4 le plus rapidement possible, trouve une séquence qui minimise le nombre de mouvements.

Input

Il y a une ligne d'entrée contenant un entier n ($2 \leq n \leq 20$) — le nombre avec lequel tu dois commencer la séquence.

Output

Sur la première ligne, imprime un entier k représentant le nombre minimum de mouvements.

Ensuite, sur les k lignes suivantes, imprime deux entiers, a et b , signifiant que tu as déplacé le disque du dessus de la pile a vers la pile b .

Example

standard input	standard output
2	3 1 2 1 3 2 3

Note

