\*

\* **@author** lena.manukyan

\*

\*ցանկացած թիվ կարելի ներկայացնել 2 -ի աստիճանների գումարով.

\* (2 -ի 2k+1 աստիճան)+1 բաժանվում է 3 , (2 -ի 2k աստիճան)-1 բաժանվում է

\* => (2 -ի 2k+1 աստիճան) + (2 -ի 2k աստիճան) -ը բաժանվում է 3: այսինքն եթե կենտ տեղերում գտնվող

\* 1 -երի քանակը = զույգ տեղերում գտնվող 1 -երի քանակին => թիվը կբաժանվի 3 -ի.

\*

\* Մնում է ցույց տալ որ կենտ տեղերում գտնվող 1 -երի քանակը - զույգ տեղերում գտնվող 1 -երի քանակիը

\* բաժանվում է 3 -ի պայմանից => թիվը բաժանվում է 3 -ի և ցույց տալ որ դա բավարար է.

\*

\* դա <=> որ 2 -ի ցանկագած զույգ աստիճանների գումարը 3k քանակությամբ բաժանվում է 3 -ի <=>

\* k = 1 վերցնելուն.( նույնը կենտ աստիճանների մասին է ճիշտ).

\*

\* ենթադրենք հակառակը ՝ (2 -ի 2k1) + (2 -ի 2k2) + 1 չի բաժանվում 3 -ի => բաժանվելուց տալիս է կամ

\*

\* 1 կամ 2 մնացորդ: 1 -ի դեպքում => (2 -ի 2k1) + (2 -ի 2k2) բաժանվում 3 -ի <=> (2 -ի 2k1) + 1

\* բաժանվում 3 -ի ինչը հակասում է (2 -ի 2k1) - 1 բաժանվում 3 -ի պայմանին:

\* 2 մնացորդի դեպքում => (2 -ի 2k1) + (2 -ի 2k2) - 1 բաժանվում 3 -ի => (2 -ի 2k1) բաժանվում 3 -ի

\* ինչը հնարավոր չէ => ենթադրությունը սխալ է:

\*

\* Մնում է ցույց տալ որ սա բավարար պայման է .

\* 1) (2 -ի 2k1) + (2 -ի 2k2) + 1 + (2 -ի 2k3 ) չի բաժանվում 3 -ի

\* 2) (2 -ի 2k1) + (2 -ի 2k2) + 1 + (2 -ի 2k3 ) + (2 -ի 2k4 ) չի բաժանվում 3 -ի

\* 3) (2 -ի 2k1+1) գումարելը 1 հատ զույգ աստիճանով կվերցնի և ստացվածը նորից չի բաժանվի 3 -ի

\* => բավարարում է:

\*/

**public** **class** DivisionBy3 {

**private** **static** **boolean** divBy3\_1(**int** n) {

**int** tempZuyg;

**int** tempKent;

**int** kent = 0;

**int** zuyg = 0;

**for** (**int** i = 0, j = 1; i < 32; i += 2, j += 2) {

tempKent = (n >>> i) & 1;

tempZuyg = (n >>> j) & 1;

**if** (tempKent == 1) {

kent++;

}

**if** (tempZuyg == 1) {

zuyg++;

}

}

**int** abs = Math.*abs*(kent - zuyg);

**if** (abs == 0 || abs == 3 || abs == 6 || abs == 9 || abs == 12 || abs == 15) {

**return** **true**;

}

// else

**return** **false**;

}

**public** **static** **void** main(String[] args) {

System.***out***.println(*divBy3\_1*(3));

//System.out.println(divBy3\_2(13));

}

}