1) Zenbaki oso bat emanda >=1 kalkulatu bere zatitzaile guztiak (bere burua kontutan izanda).

1) Espezifikazioa

Sarrera: Zenbaki oso bat
Aurre: n:balioa >=1
Irteera: zenbaki oso bat edo gehiago
Post: idatzi balioak >=1 | balio horiek n:balioaren zatitzaileak izango diren

2) Proba kasuak

1 zatitzaileak 1
2 zatitzaileak 1 eta 2
7 zatitzaileak 1 eta 7

3) Algoritmoa

```
n, zatitzaile_posiblea: integer;
Irakurri(n);
zatitzaile_posiblea ← 1;
Errepikatu atera zatitzaile_posiblea > n egia denean
        n hondarra zatitzaile_posiblea =0 egia_balitz
                idatzi(zatitzaile_posiblea);
        amaiera;
zatitzaile_posiblea ← zatitzaile_posiblea + 1;
amaiera_errepikatu;
```

4) Simulazioa edo traza

Buelta kopurua	Sartu?	n	zatitzaile_posible
0	**-**	**1**	**1**
1	**bai**		

Buelta kopurua	Sartu?	n	zatitzaile_posible
0	**-**	**2**	**1**
1	**bai**		

Buelta kopurua	Sartu?	n	zatitzaile_posible
0	**-**	**7**	**1**
1	**bai**		

1) Zenbaki oso bat emanda >=1 erabaki ea zenbakia lehena[1] den ala ez.

1) Espezifikazioa

Sarrera: zenbaki oso bat
Aurre: n:balioa >=1
Irteera: mezu bat
Post: idatzi "zenbakia lehena da" zenbakiak ez badauka zatitzaile osorik (bere burua eta unitatea kontuan izan gabe), salbuespena: unitatea. Unitatea ez da lehena, beraz idatzi beharko genuke "zenbakia ez da lehena".

2) Proba kasuak

1 ez da lehena
2 lehena da
4 ez da lehena

3) Algoritmoa

```
znb, zatitzaile_posiblea, kontadorea: integer
zatitzaile_posiblea ← 1;
Irakurri(znb);
kontadorea ← 0;
Errepikatu atera zatitzaile_posiblea > znb egia denean
        znb hondarra zatitzaile_posiblea = 0 egia_balitz
                kontadorea ← kontadorea + 1;
        amaiera;
        zatitzaile_posiblea ← zatitzaile_posiblea + 1;
amaiera;
kontadorea = 2 egia_balitz
        idatzi ("Zenbakia lehena da.");
bestela
        idatzi ("Zenbakia ez da lehena");
amaiera;
```

4) Simulazioa edo traza

Buelta kopurua	**Sartu?**	**n**	**zatitzaile_posiblea**	**Kontadorea**
0	**-**	**1**	**1**	**0**
1	**bai**			

[1] Zenbaki lehenen multzoa, zenbaki arrunt multzoaren azpimultzoa da. **UNITATEA** ez da lehena (hitzarmenez finkatutako balioa da). Lehenengo hogei zenbaki lehenak, hauek dira: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67 y 71.

Buelta kopurua	Sartu?	n	zatitzaile_posiblea	Kontadorea
0	**-**	**2**	**1**	**0**
1	**bai**			

Buelta kopurua	Sartu?	n	zatitzaile_posiblea	Kontadorea
0	**-**	**4**	**1**	**0**
1	**bai**			

Algoritmoaren beste bertsio posible bat:

znb, zatitzaile_posiblea, kontadorea: integer;
zatitzaile_posiblea ← 2;
kont ← 0;
Irakurri(znb);
Errepikatu atera zatitzaile_posiblea > znb – 1 **egia denean**
    znb **hondarra** zatitzaile_posiblea = 0 egia_balitz
        kont ← kont +1;
    amaiera;
    zatitzaile_posiblea ← zatitzaile_posiblea + 1;
amaiera_errepikatu;
kont = 0 AND znb /= 1 **egia_balitz**
    Idatzi **(**“Zenbakia LEHENA da”**);**
bestela
    Idatzi**(**“Zenbakia EZ da lehena**);**
amaiera;

Baina, zer gertatuko litzaidake zenbakia 30.002 izango balitz? ¿2 eta 30.001 arteko zatitzaile posible guztiak probatu behar ditut 30.002 ez dela Lehena jakiteko? Benetan, lehenengo bueltan, 2rekin probatzen dudanean ZIUR nago LEHENA EZ DELA.

Buelta kopurua	Sartu?	n	zatitzaile_posiblea	Kontadorea
0	**-**	**30002**	**2**	**0**
1	**bai**			**1**

Orduan, zertarako probatu beste zatitzaile posible guztiak? Lehena ez dela ziur dakigunean iteraziotik irten gaitezke. Horixe egingo dugu ebazpen honetan:

Honi flag edo salataria deitzen zaio. Zatitzaile bat aurkitu dugula erakutsiko digu bere balioa true-tik false-ra aldatuz.

```
znb, zatitzaile_posiblea: integer;
lehena:boolean;
zatitzaile_posiblea ← 2;
lehena ← true; --hasieraketan zenbakia lehena da
Irakurri(znb);
erepikatu atera (zatitzaile_posiblea > znb - 1) OR (lehena = false);
        znb hondarra zatitzaile_posiblea = 0 egia_balitz
                lehena ← false;
        bestela
                zatitzaile_posiblea ← zatitzaile_posiblea + 1;
        amaiera;
amaiera;
lehena and znb /= 1 egia_balitz
        Idatzi (“Zenbakia LEHENA da”);
bestela
        Idatzi(“Zenbakia EZ da lehena);
amaiera;
```