Opgave 1. NIO-getallen.

Eigenschappen van gehele getallen zijn altijd erg belangrijk in de informatica. Zo werken allerlei beveiligingssystemen met het kunnen ontbinden van een getal in zogenaamde priemfactoren. Vooraf een stukje getaltheorie.

Een paar begrippen en feiten uit de getaltheorie

(De tak van de wiskunde die zich bezighoudt met de eigenschappen van natuurlijke en gehele getallen).

Deler en veelvoud
Een natuurlijk getal `b` is een deler (engels divisor) van een positief natuurlijk getal `a` als er een natuurlijk getal `k` bestaat zodanig dat `a = k * b`. Het getal `a` noem je een veelvoud van `b`. Als `b` geen deler is van `a` dan zeg je dat `a` niet deelbaar is door `b`.
Voorbeelden:
3 is een deler van 12 want 12 = 4 * 3
13 is een deler van 260 want 260 = 20 * 13
3 is geen deler van 14 want er bestaat geen geheel getal k waarvoor geldt 14 = k * 3
1 is een deler van 11 want 11 = 11 * 1
11 is een deler van 11 want 11 = 1 * 11

Priemgetallen
Priemgetallen zijn positieve, gehele getallen die niet te schrijven zijn als het product van twee kleinere positieve, gehele getallen. Bijvoorbeeld 15 is niet priem, want 15 = 3 * 5 maar 11 is wel priem. Je kunt ook zeggen dat een priemgetal precies twee positieve delers heeft, namelijk 1 en zichzelf. De eerste 10 priemgetallen zijn:
2 , 3 , 5 , 7 , 11 , 13 , 17 , 19, 23 , 29

Hoofdstelling getaltheorie
Elk natuurlijk getal groter dan 1 is op eenduidige manier te ontbinden in priemfactoren. Dat wil zeggen dat je elk getal op precies een manier kunt schrijven als het product van priemgetallen
Voorbeelden:
12 = 2 * 2 * 3
21 = 3 * 7
23 = 23
100 = 2 * 2 * 5 * 5

In deze opgave ga je enkele problemen oplossen over (priem)getallen en ontbinden in factoren.

Invoerbestand:

Je programma's krijgen invoer uit een bestand `GETAL.IN`
Dit bestand bestaat uit twee regels.
Op de eerste regel staat een geheel getal `a`, met `1 < a < 29999`.
Op de tweede regel staat een geheel getal `b`, met `a < b < 30000`.
Er ligt tenminste één priemgetal tussen `a en b`.

Voorbeeld (wordt ook bij alle deelopgaven gebruikt):

GETAL0.IN
```
36
53
```

Beschikbare invoerbestanden:

`GETAL0.IN` `GETAL1.IN` `GETAL2.IN` `GETAL3.IN`

Testprogramma.

Er is een bestand TEST1.BAT dat je kunt gebruiken op de volgende manier:

```
TEST1 NIO1A GETAL0.IN
```

Deze opdracht start het programma `NIO1A` op nadat `GETAL0.IN` is gekopieerd naar `GETAL.IN`

Taakoverzicht opgave 1, NIO-getallen.

Onderdeel	Programma	Uitvoer	Tijdlimiet	Aantal testen	Punten per test	Totaal
1A	NIO1A	NIO1A.UIT	2 seconde	5	3	15
1B	NIO1B	NIO1B.UIT	1 seconde	5	4	20
1C	NIO1C	NIO1C.UIT	1 seconde	5	5	25
1D	NIO1D	NIO1D.UIT	2 seconde	5	8	40

Onderdeel 1A. Zoek de priemgetallen.

Schrijf een programma NIO1A. Invoer is uit GETAL.IN, uitvoer gaat naar NIO1A.UIT
Het programma zoekt alle priemgetallen tussen `a` en `b`. Ieder gevonden priemgetal (er is er minstens één) komt op een aparte regel, oplopend van kleinst af aan.

Voorbeeld:

```
37
41
43
47
```

Onderdeel 1B. Ontbinden in factoren.

Schrijf een programma NIO1B. Invoer is uit GETAL.IN, uitvoer gaat naar NIO1B.UIT
Op de eerste regel van de uitvoer staat de ontbinding in priemfactoren van `a`, op de tweede regel die van `b`.
De priemfactoren staan in oplopende volgorde, rondom de '`*`' staat aan weerszijden één spatie.

Voorbeeld:

```
2 * 2 * 3 * 3
53
```

.

NIO-getallen.

Je krijgt het **NIO-getal** `N(g)` van een getal `g (g >1)` door `g` te ontbinden in priemfactoren, al deze priemfactoren met 1 te verhogen en het resultaat weer met elkaar te vermenigvuldigen.
Zo is:

```
N(36)=N(2*2*3*3)=3*3*4*4=144.
N(53)=54.
```

Als `N(g)` het NIO-getal is van `g`, is `g` *een* **retourgetal** van `N(g)`. Eén van de vragen die hierbij interessant zijn, is of er getallen `g1` en `g2` bestaan waarvoor geldt `N(g1)=N(g2)`. In zo'n geval heeft `N(g1)` verschillende retourgetallen. Bij onderdeel 1D kom je daar een voorbeeld van tegen.
Ook blijkt het zo te zijn dat niet ieder getal een retourgetal heeft!

Onderdeel 1C. Bereken NIO-getallen.

Schrijf een programma NIO1C. Invoer is uit GETAL.IN, uitvoer gaat naar NIO1C.UIT
Op de eerste regel van de uitvoer staat N(a), op de tweede regel N(b).

Voorbeeld:

```
144
54
```

Onderdeel 1D. Retourgetallen.

Schrijf een programma NIO1D. Invoer is uit GETAL.IN, uitvoer gaat naar NIO1D.UIT.
Op de eerste regel van de uitvoer staat een getal `na`. Dat geeft aan dat er `na` retourgetallen van `a` gevonden zijn. Vervolgens komen er `na` regels, met op elke regel één retourgetal van `a`. Deze getallen worden gepresenteerd in oplopende volgorde. Daarna komt een regel met een getal `nb`, die aangeeft dat er `nb` retourgetallen zijn bij `b`. Tenslotte komen er `nb` regels, met daarop telkens één retourgetal van `b`, weer in oplopende volgorde.

Voorbeeld:

```
3
12
22
25
0
```