

## Opgave 2. Meeluisteren.



Inlichtingendiensten, politie en allerlei andere instanties volgen berichtenverkeer om te onderzoeken of er dingen worden uitgewisseld die hun bijzondere belangstelling hebben. In deze opgave gaan we er van uit dat er een bekend patroon is, dat moet worden opgespoord in een onderschept bestand. Dat bestand is de weergave van een bericht, waarbij vaak één of andere vorm van versleuteling is gebruikt. Want zowel het op te sporen patroon als het bestand waarin moet worden gezocht bestaan in deze opgave uit getallen. Een mogelijk voorkomen van het patroon is een gedeelte van het bestand dat overeenkomt met het patroon of dezelfde kenmerken heeft (zie verder de omschrijving van de deelopgaven). Uitgaande van verschillende aannames over de wijze waarop het bestand is gemaakt ga je in de deelopgaven op zoek naar de eventuele voorkomens van het patroon in het bestand.

Invoer.

Het te onderzoeken bestand staat in een bestand `file.in` dat begint met een regel met daarop één getal  $R$ . Dit getal, met  $1 \leq R \leq 40000$  geeft aan hoeveel regels nog zullen volgen. De volgende  $R$  regels bestaan elk uit 16 getallen  $G$ , met  $0 \leq G \leq 255$  die gescheiden zijn door spaties.

Het op te sporen patroon staat in een bestand `patroon.in` dat bestaat uit één regel van 16 getallen  $G$ , met eveneens  $0 \leq G \leq 255$ .

Voorbeeld:

```
file0.in: 6
          75 12 0 33 79 100 101 101 110 218 79 110 100 101 114 100
          101 101 75 218 33 12 0 79 110 100 44 15 88 90 171 100
          101 102 105 1 2 5 4 33 5 4 4 41 89 70 55 71
          1 2 5 1 2 2 44 106 92 202 254 55 71 1 109 140
          130 131 144 130 131 131 105 248 63 42 30 109 140 130 109 140
          140 43 106 29 20 245 42 30 109 8 7 6 5 4 3 2
```

```
patroon0.in 79 110 100 101 114 100 101 101 75 218 33 12 0 79 110 100
```

**Dit bestand wordt bij de voorbeelden bij alle opgaven als invoer gebruikt.**

De getallen in het bestand `file.in` hebben een volgnummer (de eerste regel bevat de getallen met volgnummer 1 tot en met 16, en zo verder, eventueel tot en met nummer 640.000).

Het is mogelijk dat mogelijke voorkomens van het patroon elkaar overlappen!

## Voorbeeldbestanden en testen:

Er zijn bestanden `file0.in`, `file1.in` tot en met `file5.in` en `patroon0.in` tot en met `patroon5.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test2.bat`, die je kunt gebruiken op de volgende manier:

```
test2 nio2a file0.in
```

Met deze opdracht test je het programma `nio2a` (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit `file0.in` (of op deze plaats één van de andere bestanden) naar het bestand `file.in` wordt gekopieerd en de inhoud van `patroon0.in` naar `patroon.in`. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

## Opgave 2 overzicht

Onderdeel	Programma	Uitvoer	Tijdlimiet per test	Aantal testen	Punten per test	Totaal
2A	<code>nio2a</code>	<code>2a.uit</code>	1 sec	6	2	12
2B	<code>nio2b</code>	<code>2b.uit</code>	3 sec	6	3	18
2C	<code>nio2c</code>	<code>2c.uit</code>	3 sec	6	4	24
2D	<code>nio2d</code>	<code>2d.uit</code>	4 sec	6	5	30
2E	<code>nio2e</code>	<code>2e.uit</code>	4 sec	6	5	30
2F	<code>nio2f</code>	<code>2f.uit</code>	5 sec	6	6	36

## Onderdeel 2A: Tel de getallen uit het patroon in het bestand.

Schrijf een programma `nio2a` dat een bestand `file.in` en een bestand `patroon.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2a.uit` dat bestaat uit 16 regels. Op iedere regel  $N$  staat het  $N$ -de getal uit het patroon, daarna een spatie, en dan een getal  $M$  dat aangeeft hoe vaak het betreffende getal in het bestand voorkomt.

```
Uitvoer bij het gegeven voorbeeld:  file0.in  patroon0.in 79 5
                                         110 2
                                         100 10
                                         101 7
                                         114 2
                                         100 10
                                         101 7
                                         101 7
                                         75 1
                                         218 1
                                         33 1
                                         12 1
                                         0 1
                                         79 5
                                         110 2
                                         100 10
```

## Onderdeel 2B: Zoek het patroon zelf.

Schrijf een programma `nio2b` dat een bestand `file.in` en een bestand `patroon.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2b.uit` dat bestaat uit een aantal regels. Op iedere regel staat een beginpositie aangegeven van het patroon in het bestand; het weergegeven getal is het volgnummer van deze beginpositie in het bestand. De eventuele beginposities worden gegeven in de volgorde waarin ze in het bestand voorkomen; het bestand wordt afgesloten met een regel met het getal 0.

```
Uitvoer bij het gegeven voorbeeld:  file0.in  patroon0.in 11
                                         0
```

```
file0.in: 6
75 12 0 33 79 100 101 101 110 218 79 110 100 101 114 100
101 101 75 218 33 12 0 79 110 100 44 15 88 90 171 100
101 102 105 1 2 5 4 33 5 4 4 41 89 70 55 71
1 2 5 1 2 2 44 106 92 202 254 55 71 1 109 140
130 131 144 130 131 131 105 248 63 42 30 109 140 130 109 140
140 43 106 29 20 245 42 30 109 8 7 6 5 4 3 2

patroon0.in 79 110 100 101 114 100 101 101 75 218 33 12 0 79 110 100
```

## Onderdeel 2C: Gebruik van een Caesarcode.



Julius Caesar (zie afbeelding) maakte zijn berichten voor anderen onleesbaar door alle letters in het alfabet een vast aantal plaatsen te laten opschuiven; hij sprak met zijn generaals af hoe groot deze afstand was. Dat kan natuurlijk ook wanneer het gaat om een bestand dat is opgebouwd uit getallen; tel bij alle getallen uit het oorspronkelijke bericht een vast getal op, maar kies dat getal zo, dat alle getallen na deze bewerking nog steeds een waarde hebben tussen 0 en 255. Deze bewerking heet een Caesarcode; als het vaste getal 0 is blijft het bestand uiteraard onveranderd. Bij dit onderdeel ga je op zoek naar alle voorkomens van het patroon, waarbij je er rekening mee moet houden dat er een Caesarcode is gebruikt om het bestand te maken.

Schrijf een programma `nio2c` dat een bestand `file.in` en een bestand `patroon.in` als invoer krijgt; het programma geeft als uitvoer een tekstbestand `2c.uit` dat bestaat uit een aantal regels. Op iedere regel staat een beginpositie aangegeven van een mogelijk voorkomen van het patroon in het bestand, waarbij rekening wordt gehouden met het gebruik van een Caesarcode; het weergegeven getal is het volgnummer van deze beginpositie in het bestand. De eventuele beginposities worden gegeven in de volgorde waarin ze in het bestand voorkomen; het bestand wordt afgesloten met een regel met het getal 0.

Uitvoer bij het gegeven voorbeeld:      `file0.in`      `patroon0.in` 11  
63  
0

```
file0.in: 6
75 12 0 33 79 100 101 101 110 218 79 110 100 101 114 100
101 101 75 218 33 12 0 79 110 100 44 15 88 90 171 100
101 102 105 1 2 5 4 33 5 4 4 41 89 70 55 71
1 2 5 1 2 2 44 106 92 202 254 55 71 1 109 140
130 131 144 130 131 131 105 248 63 42 30 109 140 130 109 140
140 43 106 29 20 245 42 30 109 8 7 6 5 4 3 2
```

```
patroon0.in 79 110 100 101 114 100 101 101 75 218 33 12 0 79 110 100
```

## Onderdeel 2D: Gebruik van een substitutie.

Wanneer ieder getal uit het bericht wordt vervangen door een getal, zodat gelijke getallen uit het bericht worden vervangen door gelijke getallen in het bestand, en gelijke getallen in het bestand ook horen bij hetzelfde getal uit het bericht, spreken we van een substitutie.

Voorbeeld:

Bericht	250	249	248	179	210	200	201	214	200	201	175	109	140	130	131	144
Bestand	12	17	190	1	100	35	215	94	35	215	0	4	96	111	131	18

Ieder getal uit het bericht wordt eenduidig vervangen door een bepaald getal in het bestand. Dat mag eventueel het zelfde getal zijn (zie de 131 in het voorbeeld). Een Caesarcode is een bijzonder geval van een substitutie.

Bij dit onderdeel ga je op zoek naar alle voorkomens van het patroon, waarbij je er rekening mee moet houden dat er een substitutie is gebruikt om het bestand te maken.



