

Opdracht 1. KMO en een extraatje

In deze opgave moet je een aantal problemen met woorden en letters oplossen.

We spreken af dat we de letters A, E, I, O, U en Y als klinkers beschouwen. De andere 20 letters zijn medeklinkers.

Bij een woord kun je de K-waarde, de M-waarde en O-waarde berekenen. De K-waarde is de frequentie van een klinker die het meest in het woord voorkomt. De M-waarde is de frequentie van een medeklinker die het meest in het woord voorkomt. De O-waarde is de frequentie van een letter die het meest in het woord voorkomt als je de, al voor de K-waarde en de M-waarde gebruikte, meest frequente klinker en medeklinker weglaat. Het woord 'FREQUENTLY' heeft een K-waarde van 2, want er zitten 2 E's in, een M-waarde van 1, want alle medeklinkers komen maar 1 keer voor, en ook een O-waarde van 1 omdat geen enkele andere letter meer dan 1 keer voorkomt.

Woorden met dezelfde K-waarde, M-waarde en O-waarde heten KMO-equivalent.

Invoer:

Invoer bij de opdrachten 1A tot en met 1D is een bestand `word.in` dat bestaat uit één regel. Daarop staat een woord van tenminste 2 en ten hoogste 12 hoofdletters. Alleen de 26 hoofdletters worden gebruikt. Bij de meeste opdrachten moet je daarnaast gebruik maken van een bestand met woorden, het bestand `words.dat` dat bestaat uit alle toegestane woorden. Dit bestand wordt voor alle testen gebruikt; het bevat 94235 woorden, één woord per regel, geordend zoals een woordenboek. Het woord van de invoer komt altijd in deze lijst voor.

Om problemen met accenten en andere bijzondere dingen van het Nederlands te voorkomen wordt gebruik gemaakt van een Engelse woordenlijst.

Voorbeeld bestand `word0.in`:

```
FREQUENTLY
```

Dit bestand wordt bij de voorbeelden bij alle opgaven (behalve 1E) als invoer gebruikt.

Voorbeeldbestanden en testen:

Er zijn bestanden `word0.in`, `word1.in` tot en met `word5.in` beschikbaar waarmee je je programma kunt uitproberen.

Er is een batchfile `test1.bat`, die je kunt gebruiken op de volgende manier:

```
test1 niola word0.in
```

Met deze opdracht test je het programma `niola` (of op deze plaats één van je andere programma's), waarbij vooraf eerst de invoer uit `word0.in` (of op deze plaats één van de andere bestanden) naar het bestand `word.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 1. Taakoverzicht.

Opdracht	Programma	Invoer	Uitvoer	Tijdlimiet	Testen	Punten per test	Totaal
1A	<code>niola</code>	<code>word.in</code>	<code>1a.uit</code>	1 s	5	2	10
1B	<code>niolb</code>	<code>word.in</code> , <code>words.dat</code>	<code>1b.uit</code>	5 s	5	3	15
1C	<code>niolc</code>	<code>word.in</code> , <code>words.dat</code>	<code>1c.uit</code>	5 s	5	4	20
1D	<code>niold</code>	<code>word.in</code> , <code>words.dat</code>	<code>1d.uit</code>	5 s	5	6	30
1E	---	<code>words.dat</code>	<code>1e.uit</code>	---	1	25	25

Opdracht 1A: De K-waarde, M-waarde en O-waarde.

Schrijf een programma `nio1a` dat een bestand `word.in` inleest. Uitvoer is een bestand `1a.uit` van drie regels. De eerste regel geeft de K-waarde van het woord uit de invoer. De tweede regel geeft de M-waarde van het woord uit de invoer. De derde regel geeft de O-waarde van het woord uit de invoer.

Voorbeeld: 2
 1
 1

Opdracht 1B: Aantal KMO-equivalente woorden.

Schrijf een programma `nio1b` dat een bestand `word.in` inleest en het bestand `words.dat` gebruikt. Uitvoer is een bestand `1b.uit` van één regel. Op deze regel staat hoeveel woorden uit het bestand `words.dat` KMO-equivalent zijn met het woord uit het invoerbestand. Let op: Het woord uit de invoer mag niet worden meegeteld.

Voorbeeld: 17319

Opdracht 1C: Kortste KMO-equivalente woord.

Schrijf een programma `nio1c` dat een bestand `word.in` inleest en het bestand `words.dat` gebruikt. Uitvoer is een bestand `1c.uit` van één regel. Op deze regel staat een woord uit het bestand `words.dat` dat KMO-equivalent is met het woord uit het invoerbestand, maar niet het woord uit de invoer zelf! De lengte van het uitgevoerde woord is zo kort mogelijk, en als er uit verschillende woorden kan worden gekozen wordt het woord weergegeven dat het eerst in het bestand `words.dat` voorkomt. Als er geen KMO-equivalent woord kan worden gevonden geef je als uitvoer het woord 'geen' in kleine letters.

Voorbeeld: AAHS

Opdracht 1D: De KMO-puzzel.

Bij de KMO-puzzel is het de bedoeling om gegeven een zekere K-, M- en O-waarde voor alle 120 combinaties van een klinker en een medeklinker het kortste woord te vinden waarbij deze klinker en deze medeklinker precies K en M keer voorkomen, terwijl alle andere letters niet meer dan O keer voorkomen (en minstens één ervan precies O keer). Als het lukt om zo'n woord te vinden is de lengte van het woord het aantal strafpunten dat wordt behaald; als het niet lukt om zo'n woord te vinden worden er voor deze combinatie 15 strafpunten geteld. Uiteraard wil je zo wienig mogelijk strafpunten halen.

Schrijf een programma `nio1d` dat een bestand `word.in` inleest en het bestand `words.dat` gebruikt. Uitvoer is een bestand `1d.uit` van één regel. Op deze regel staat het laagste aantal strafpunten dat kan worden behaald met de KMO-puzzel die hoort bij de K-, M- en O-waarde van het woord uit het invoerbestand.

Voorbeeld: 751

Opdracht 1E: Een extraatje.

Bij deze opdracht wordt alleen een bestand met je oplossing verwacht. Gevraagd wordt een aantal woorden uit het bestand `words.dat` te zoeken, waarvoor geldt dat alle 26 letters van het alfabet in die woorden worden gebruikt. Het aantal woorden dat je gebruikt moet zo klein mogelijk zijn, en de totale lengte van de gebruikte woorden ook.

Uiteraard mag je een computerprogramma schrijven om dit probleem op te lossen. Je levert echter alleen je oplossing in!

De uitvoer wordt geschreven in een bestand `1e.uit`. Dat bestand begint met een regel waarop staat aangegeven hoeveel woorden je gebruikt. Daarna komt er een regel waarop het totaal aan gebruikte letters wordt vermeld. Vervolgens wordt ieder woord op een aparte regel vermeld. Voor woorden worden hoofdletters gebruikt, net als in de rest van de opgave. Ieder woord moet voorkomen in het bestand `words.dat`

Een voorbeeld: Stel je hebt de woorden `ABCDEFGH`, `HIJKLMN`, `OPQRSTU` en `VWXYZ` gevonden (maar deze woorden staan niet in het bestand). Je uitvoer zal er dan als volgt uitzien:

```
4
29
ABCDEFGH
HIJKLMN
OPQRSTU
VWXYZ
```

Er is geen tijdlimiet verbonden aan deze opgave; je moet een bestand hebben gesubmit voor het einde van de wedstrijd.

Het is mogelijk om bij deze opgave een gedeeltelijke score te halen. Voor de beste oplossing krijg je 25 punten, voor ieder woord meer dan de beste oplossing gaan er 5 punten af. Ook voor het gebruik van meer letters dan nodig gaan er punten af.

Stel de beste oplossing bestaat uit W woorden van in totaal L letters. De score wordt dan als volgt vastgesteld:

Letters Woorden	t/m L	L+1	L+2, L+3	L+4 - L+6	L+7 en meer
W	25	24	23	22	21
W+1	20	19	18	17	16
W+2	15	14	13	12	11
W+3	10	9	8	7	6
W+4	5	4	3	2	1