

Opdracht 1. Woordzoekers.

Een woordzoeker is een puzzel waarbij je een diagram met letters krijgt en een lijst met woorden die verborgen zitten in dat diagram. De woorden kunnen horizontaal, verticaal en diagonaal worden gevonden, zowel omlaag als omhoog en van links naar rechts als andersom.

In deze opgave krijg je wel een diagram met letters gegeven en voor de latere onderdelen een woordenlijst, **maar niet alle woorden van die lijst komen in het diagram voor.**

Voorbeeld diagram:

D	A	S
E	V	E
R	Z	W
Y	N	V
O	S	H
E	R	T

Invoer:

Invoer is een bestand **diagram.in**.

Op de eerste regel staat een getal R , met $2 \leq R \leq 12$, dat het aantal regels van het diagram aangeeft.

Op de tweede regel staat een getal K , met $2 \leq K \leq 12$, dat het aantal kolommen van het diagram aangeeft.

Vervolgens zijn er R regels, met op iedere regel K hoofdletters.

Voorbeeld bestand **diagram0.in**:

```
6
3
DAG
EVE
RZW
YNV
OSH
ERT
```

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

De woordenlijst is een bestand met de naam **woorden.in**. Dit bestand bestaat uit 94235 regels met op iedere regel één woord. Deze woorden zijn alfabetisch lexicografisch geordend (dat wil zeggen net als in een woordenboek). Ieder woord in dit bestand bestaat uit N hoofdletters, met $2 \leq N \leq 12$.

Voorbeeldbestanden en testen:

Er zijn bestanden `diagram0.in`, `diagram1.in` tot en met `diagram5.in` beschikbaar waarmee je je programma kunt uitproberen.

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

```
test1 niola diagram0.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 `diagram0.in` (of op deze plaats één van de andere bestanden) naar het bestand `diagram.in` wordt gekopieerd. Je zult dan zelf moeten controleren of het programma binnen de tijdlimiet stopt en de goede uitvoerfile maakt.

Opdracht 1. Taakoverzicht.

Onderdeel	Programma	Uitvoer	Tijdlimiet per test	Aantal testen	Punten per test	Totaal
1A	<code>niola</code>	<code>1a.uit</code>	0,1 sec	6	1	6
1B	<code>niolb</code>	<code>1b.uit</code>	0,1 sec	6	2	12
1C	<code>niolc</code>	<code>1c.uit</code>	1 sec	6	3	18
1D	<code>niold</code>	<code>1d.uit</code>	2 sec	6	4	24
1E	<code>niole</code>	<code>1e.uit</code>	5 sec	4	6	24
1F	<code>niolf</code>	<code>1f.uit</code>	5 sec	4	4	16

Opdracht 1A: Aantal letters in het diagram.

Schrijf een programma `niola` dat een bestand `diagram.in` inleest. Uitvoer is een bestand `1a.uit` van één regel; hierop staat aangegeven hoeveel letters er in het ingevoerde diagram staan.

Voorbeeld: 18

Opdracht 1B: De ongebruikte letters.

Schrijf een programma `niolb` dat een bestand `diagram.in` inleest. Uitvoer is een bestand `1b.uit` dat bestaat uit één regel; hierop staan alle letters aangegeven die niet in het diagram voorkomen, in de vorm van hoofdletters, op alfabetische volgorde, zonder spaties ertussen, afgesloten door een punt. Als alle 26 verschillende letters minstens eenmaal voorkomen bevat de uitvoer dus alleen een regel met een punt erop.

Voorbeeld: BCFG IJKLM PQUX.

Opdracht 1C: Aantal mogelijke woorden.

Schrijf een programma `niolc` dat een bestand `diagram.in` inleest. Uitvoer is een bestand `1c.uit` dat bestaat uit één regel; hierop staat aangegeven hoeveel mogelijke woorden van 2 of meer letters in het diagram kunnen worden gevonden.

Voorbeeld: 182

Opdracht 1D: De verschillende mogelijke woorden.

Schrijf een programma `ni01d` dat een bestand `diagram.in` inleest; uitvoer is een bestand `1d.uit` met op de eerste regel aangegeven hoeveel verschillende mogelijke woorden M er in het diagram worden aangetroffen. Vervolgens zijn er M regels met op elke regel één van die mogelijke woorden, op alfabetische volgorde.

Voorbeeld (voor de leesbaarheid in kolommen geplaatst):

169	ERYOE	NV	RT	THVWE	WVD
AD	ES	NW	RV	THVWES	WVH
AE	ESV	NY	RVS	TR	WVHT
AS	EV	NZ	RY	TRE	WZ
AV	EVE	NZV	RYO	TS	WZR
AVZ	EW	NZVA	RYOE	TSY	YN
AVZN	EWV	OE	RZ	VA	YNV
AVZNS	EWVH	ON	RZW	VD	YO
AVZNSR	EWVHT	ONW	SA	VE	YOE
DA	EZ	OR	SAD	VH	YR
DAS	EZV	OS	SE	VHT	YRE
DE	EZY	OSH	SEW	VN	YRED
DER	HH	OY	SEWV	VNY	YS
DERY	HHR	OYR	SEWVH	VR	YST
DERYO	HN	OYRE	SEWVHT	VS	YZ
DERYOE	HNR	OYRED	SH	VSE	YZE
DV	HR	RE	SN	VW	ZE
DVW	HS	RED	SNZ	VWE	ZN
EA	HSO	RH	SNZV	VWES	ZNS
ED	HT	RHH	SNZVA	VZ	ZNSR
EO	HV	RHHR	SO	VZE	ZR
EOY	HVW	RN	SR	VZN	ZV
EOYR	HVWE	RNH	ST	VZNS	ZVA
EOYRE	HVWES	RO	SV	VZNSR	ZW
EOYRED	NH	RS	SVR	WE	ZY
ER	NO	RSN	SY	WES	
ERT	NR	RSNZ	TH	WN	
ERY	NS	RSNZV	THV	WNO	
ERYO	NSR	RSNZVA	THVW	WV	

Opgave 1E: De werkelijke woorden.

Schrijf een programma `ni01e` dat een bestand `diagram.in` en een woordenlijst `woorden.in` inleest; uitvoer is een bestand `1e.uit` met op de eerste regel aangegeven hoeveel verschillende woorden W uit de woordenlijst er in het diagram worden aangetroffen. Vervolgens zijn er W regels met op elke regel één van deze woorden, op alfabetische volgorde.

Voorbeeld (voor de leesbaarheid in kolommen geplaatst):

20	DA	EVE	OR	RED	WE
AD	DE	NO	OS	SAD	
AE	ER	OE	OY	SEW	
AS	ES	ON	RE	SH	

Opgave 1F: Wat blijft er over.

Schrijf een programma `nio1e` dat een bestand `diagram.in` en een woordenlijst `woorden.in` inleest; uitvoer is een bestand `1f.uit` van één regel. Daarop staat aangegeven welke letters in het diagram niet voorkomen in één of meer van de woorden uit de woordenlijst. De letters worden aangegeven in de volgorde waarop je het diagram leest: Rij voor rij van links naar rechts. De regel wordt afgesloten met een punt.

Voorbeeld: ZVT.

Voorbeelddiagram:

