ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Εαρινό 2025

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 2

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: 4/6/2025, 23:59:59

Αποστολή Τροφοδοσίας στον Άρη (0.25+0.25 = 0.5 βαθμοί)

Η ελληνική διαστημική υπηρεσία (ΕΔΥ) σχεδιάζει μια αποστολή στον Άρη με σκοπό την εγκατάσταση της πρώτης διαστημικής ταβέρνας: “Το Κοσμικό Σουβλάκι”. Για να πετύχει η αποστολή, πρέπει να σταλεί συγκεκριμένη ποσότητα προμηθειών στον Άρη, χρησιμοποιώντας κουτιά διαφορετικών μεγεθών (βάσει όγκου).

Η αποστολή έχει διαθέσιμα **N** είδη κουτιών ($1 \leq N \leq 30$), το καθένα με συγκεκριμένη χωρητικότητα m_i μονάδων όγκου ($1 \leq m_i \leq 42$). Κάθε τύπος κουτιού μπορεί να χρησιμοποιηθεί **όσες φορές χρειαστεί**. Ο συνολικός διαθέσιμος χώρος στο διαστημόπλοιο είναι **M** μονάδες όγκου ($1 \leq M \leq 42$). Ο στόχος είναι να βρεθούν **όλοι οι πιθανοί συνδυασμοί κουτιών** που μπορούν χρησιμοποιηθούν για να γεμίσουν **ακριβώς** τον διαθέσιμο χώρο του διαστημικού οχήματος.

Σας ζητείται να γράψετε δύο προγράμματα (ένα σε Java και ένα σε Prolog) που θα διαβάζουν τα δεδομένα εισόδου από ένα αρχείο κειμένου και να επιστρέφουν όλους τους μοναδικούς συνδυασμούς κουτιών των οποίων το συνολικό άθροισμα ισούται με τον διαθέσιμο χώρο. Η πρώτη γραμμή του αρχείου εισόδου περιέχει τους αριθμούς N και M, ενώ η δεύτερη τα μεγέθη των N διαφορετικών κουτιών.

Για παράδειγμα, έστω ότι τα περιεχόμενα των αρχείων **boxes1.txt** και **boxes2.txt** είναι τα εξής.

```
$ cat boxes1.txt
4 8
2 3 5 4
```

```
$ cat boxes2.txt
2 10
3 6
```

Το πρόγραμμα σας θα πρέπει να επιστρέφει τους μοναδικούς συνδυασμούς όπως στο πρώτο παράδειγμα παρακάτω. Δεν έχει σημασία η σειρά με την οποία εμφανίζονται οι συνδυασμοί, ούτε η σειρά των κουτιών μέσα σε κάθε συνδυασμό. Σε περίπτωση που δεν υπάρχει λύση, το πρόγραμμα σας σε Java θα πρέπει να τυπώνει **IMPOSSIBLE** ενώ αυτό σε Prolog να αποτυγχάνει, όπως φαίνεται στο δεύτερο παράδειγμα παρακάτω.

Σε Java

```
$ java Boxes boxes1.txt
2 2 2 2
2 2 4
2 3 3
3 5
4 4

$ java Boxes boxes2.txt
IMPOSSIBLE
```

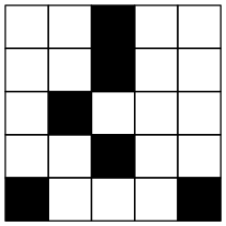
Σε Prolog

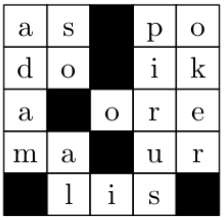
```
?- boxes("boxes1.txt", Solution).
Solution = [2, 2, 2, 2] ;
Solution = [2, 2, 4] ;
Solution = [2, 3, 3] ;
Solution = [3, 5] ;
Solution = [4, 4] ;
false.

?- boxes("boxes2.txt", Solution).
false.
```

Σταυρόλεξο (0.25+0.25 = 0.5 βαθμοί)

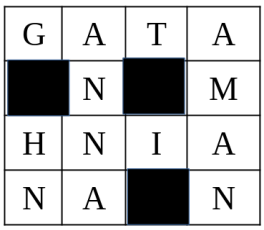
Σας δίνεται πληροφορία για ένα κενό σταυρόλεξο (συγκεκριμένα, οι δύο διαστάσεις του και τα σημεία στα οποία υπάρχουν μαύρα τετράγωνα) και μια ακολουθία από λέξεις δύο ή περισσοτέρων γραμμάτων και ο σκοπός σας είναι να τοποθετήσετε όλες τις λέξεις (καθεμία από αυτές μία φορά μόνο) μέσα στο σταυρόλεξο. Φυσικά, αυτό μπορεί να μην είναι δυνατό (δηλ. να αποτυγχάνει). Τρία παραδείγματα με τις λύσεις τους (τα δεδομένα εισόδου και οι αντίστοιχες εικόνες του σταυρόλεξου κενό και συμπληρωμένο για το πρώτο παράδειγμα και συμπληρωμένα για τα άλλα δύο) φαίνονται παρακάτω. Η πρώτη γραμμή της εισόδου περιέχει τις δύο διαστάσεις του σταυρόλεξου (τον αριθμό από τις γραμμές και τις στήλες), τον αριθμό των μαύρων τετραγώνων και τον αριθμό των λέξεων.

```
$ cat input1.txt
5 5 6 13
1 3
2 3
3 2
4 3
5 1
5 5
adam
as
al
do
ik
lis
ma
oker
ore
pirus
po
so
ur
```






```
$ cat input2.txt
4 4 3 6
2 1
2 3
4 3
AMAN
ANNA
HN
HNIA
GATA
NA
```




```
$ cat input3.txt
5 5 0 10
ALAMO
AMARA
ANOMA
ILARA
KARIM
PAMIR
SALAM
SALAN
TSAKA
TSIPA
```

T	S	A	K	A
S	A	L	A	M
I	L	A	R	A
P	A	M	I	R
A	N	O	M	A

Σας ζητείται να γράψετε δύο προγράμματα, ένα σε ML και ένα σε Prolog, που να διαβάζουν την είσοδό τους από ένα αρχείο και να τυπώνουν τις λέξεις σειρά ανά σειρά, από πάνω προς τα κάτω, όπως φαίνεται στα παρακάτω παραδείγματα. Προσέξτε ότι επειδή όλες οι λέξεις εισόδου έχουν τουλάχιστον δύο γράμματα, αν δεν υπάρχουν λέξεις εισόδου σε κάποια γραμμή το πρόγραμμά σας θα πρέπει να τυπώνει κενή γραμμή σε αυτήν την περίπτωση (όπως φαίνεται στο δεύτερο παράδειγμα παρακάτω). Σε κάθε περίπτωση, για σταυρόλεξα με λύση, το πρόγραμμά σας θα πρέπει να τυπώνει τόσες γραμμές όσες και η πρώτη διάσταση της εισόδου.

Σε MLton ή σε OCaml

```
$ ./crossword input1.txt
as po
do ik
ore
ma ur
lis

$ ./crossword input2.txt
GATA

HNIA
NA
```

Σε SML/NJ

```
- crossword "input1.txt";
as po
do ik
ore
ma ur
lis
val it = () : unit
- crossword "input2.txt";
GATA

HNIA
NA
val it = () : unit
```

Σε Prolog

```
?- crossword("input1.txt").
as po
do ik
ore
ma ur
lis
true.
?- crossword("input2.txt").
GATA

HNIA
NA
true.
```

Οι διαστάσεις του σταυρόλεξου θα είναι το πολύ 21x21. Αν υπάρχουν περισσότερες από μία λύσεις για τα δεδομένα εισόδου (όπως π.χ. στο τρίτο παράδειγμα που έχει δύο λύσεις, αν και θα προσπαθήσουμε να αποφύγουμε τέτοια δεδομένα εισόδου), το πρόγραμμά σας μπορεί να τυπώνει

οποιαδήποτε από αυτές. Σε περίπτωση που δεν υπάρχει λύση για τα δεδομένα εισόδου, το πρόγραμμά σας (σε όλες τις γλώσσες) θα πρέπει να τυπώνει **`IMPOSSIBLE`**.

Σημείωση: Το ότι η συνάρτηση **`crossword`** έχει τύπο επιστροφής **`unit`** στην ML και το κατηγόρημα **`crossword`** στην Prolog έχει μόνο ένα όρισμα εισόδου δεν είναι ότι πιο φυσικό για αυτές τις γλώσσες. Η άσκηση χρησιμοποιεί τα παραπάνω τόσο για ομοιομορφία με την MLton και την OCaml, όσο και για να μην σας δεσμεύσει σε κάποια συγκεκριμένη δομή δεδομένων (π.χ. λίστα, πίνακα, …) για την αναπαράσταση των σταυρόλεξων.

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων. Μπορείτε αν θέλετε να σχηματίσετε διαφορετική ομάδα σε σχέση με την προηγούμενη σειρά ασκήσεων – οι ομάδες στο σύστημα υποβολής είναι έτσι και αλλιώς καινούργιες για κάθε σειρά ασκήσεων.

- Δεν επιτρέπεται να μοιράζεστε τα προγράμματά σας με συμφοιτητές εκτός της ομάδας σας ή να τα βάλετε σε μέρος που άλλοι μπορούν να τα βρουν (π.χ. σε κάποια σελίδα στο διαδίκτυο, σε ιστοσελίδες συζητήσεων, …). Σε περίπτωση που παρατηρηθούν «περίεργες» ομοιότητες σε προγράμματα, ο βαθμός των εμπλεκόμενων φοιτητών *σε όλες τις σειρές ασκήσεων* γίνεται αυτόματα μηδέν ανεξάρτητα από το ποια ομάδα... «εμπνεύστηκε» από την άλλη.

- Μπορείτε να χρησιμοποιήσετε «βοηθητικό» κώδικα (π.χ. κάποιο κώδικα που διαχειρίζεται κάποια δομή δεδομένων) που βρήκατε στο διαδίκτυο στα προγράμματά σας, με την προϋπόθεση ότι το πρόγραμμά σας περιέχει σε σχόλια την παραδοχή για την προέλευση αυτού του κώδικα και ένα σύνδεσμο σε αυτόν.

- Τα προγράμματα σε ML πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε SML/NJ ≥ v110.79 ή σε MLton ≥ 20241230 ή σε Objective Caml version ≥ 4.13.1. Το σύστημα ηλεκτρονικής υποβολής σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML.

- Τα προγράμματα σε Java μπορεί να είναι και σε πολλά αρχεία αν θέλετε, αλλά θα πρέπει να περιέχουν μια κύρια κλάση με το όνομα που ζητείται και να δουλεύεουν σε Java 17 (17.0.15).

- Τα προγράμματα Prolog πρέπει να είναι σε ένα αρχείο και να δουλεύουν σε κάποιο από τα παρακάτω συστήματα SWI Prolog (9.0.4), GNU Prolog (1.4.5) ή YAP (7.6.0), το οποίο θα πρέπει να επιλέξετε.

- Η υποβολή των προγραμμάτων θα γίνει ηλεκτρονικά μέσω του helios και για να μπορέσετε να τα υποβάλλετε και να βαθμολογηθείτε για αυτά, τα μέλη της ομάδας σας (και οι δύο) θα πρέπει να έχετε εγγραφεί στο μάθημα στο helios. Τα προγράμματά σας πρέπει να διαβάζουν την είσοδο όπως αναφέρεται και δεν πρέπει να έχουν κάποιου άλλους είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.