

ΓΛΩΣΣΕΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΥ Ι

Άσκηση 1

Καταληκτική ημερομηνία και ώρα ηλεκτρονικής υποβολής: **8/5/2026, 23:59:59**

Τηλεπικοινωνιακά Καλώδια ($0.25+0.25 = 0.5$ βαθμοί)

Η Ελένη, φοιτήτρια στη ΣΗΜΜΥ, παίρνει κάποιο μάθημα σε ένα εργαστήριο τηλεπικοινωνιών. Έχει στη διάθεσή της ένα καλώδιο οπτικών ινών μήκους n μέτρων και πρέπει να το κόψει σε **τουλάχιστον δύο κομμάτια** ακέραιου μήκους (κάθε κομμάτι πρέπει να έχει μήκος τουλάχιστον ένα μέτρο). Η απόδοση κάθε κομματιού είναι ανάλογη του μήκους του, και η συνολική απόδοση του συστήματος υπολογίζεται ως το **γινόμενο** των μηκών όλων των κομματιών.

Πιο συγκεκριμένα, αν το καλώδιο κοπεί σε κομμάτια μηκών n_1, n_2, \dots, n_k (με $k \geq 2$, $n_i \geq 1$ για κάθε i , και $n_1 + n_2 + \dots + n_k = n$), η συνολική απόδοση είναι $n_1 \cdot n_2 \cdot \dots \cdot n_k$.

Η άσκηση σας ζητάει να γράψετε δύο προγράμματα (ένα σε C/C++ και ένα σε ML) τα οποία να διαβάζουν το μήκος n ($2 \leq n \leq 42\,000\,000$) από ένα αρχείο κειμένου και να τυπώνουν τη **μέγιστη δυνατή συνολική απόδοση**.

Προσοχή: η απάντηση μπορεί να είναι πολύ μεγάλος αριθμός, οπότε θα χρειαστεί να υλοποιήσετε αριθμητική μεγάλων αριθμών (big integers).

Υπόδειξη: Στην SML/NJ και στον MLton μπορείτε να χρησιμοποιήσετε την ενσωματωμένη δομή `IntInf` για αριθμητική απεριόριστης ακρίβειας.

Τα στοιχεία εισόδου διαβάζονται από ένα αρχείο κειμένου. Η μοναδική γραμμή του αρχείου περιέχει τον ακέραιο n . Παρακάτω δίνονται τρία παραδείγματα εισόδου και οι αντίστοιχες έξοδοι:

```
$ cat maxprod1.txt      $ cat maxprod2.txt      $ cat maxprod3.txt
5                        10                       259
```

Τα προγράμματά σας θα πρέπει να τρέχουν όπως φαίνεται παρακάτω.

```
Σε C/C++, MLton ή σε OCaml      Σε SML/NJ
$ ./maxprod maxprod1.txt        - maxprod "maxprod1.txt";
6                                6
$ ./maxprod maxprod2.txt        val it = () : unit
36                               - maxprod "maxprod2.txt";
$ ./maxprod maxprod3.txt        36
1436701821907442...            val it = () : unit
                                - maxprod "maxprod3.txt";
                                1436701821907442...
                                val it = () : unit
```

Στο πρώτο παράδειγμα, η βέλτιστη τομή είναι $2 + 3 = 5$, με γινόμενο $2 \cdot 3 = 6$.

Εναλλακτικά, οι τομές $1 + 2 + 2$ και $1 + 4$ δίνουν γινόμενο 4, ενώ η $1 + 1 + 3$ δίνει γινόμενο 3.

Στο δεύτερο παράδειγμα, η βέλτιστη τομή είναι $3 + 3 + 4 = 10$, με γινόμενο $3 \cdot 3 \cdot 4 = 36$.

Στο τρίτο παράδειγμα, η απάντηση είναι ένας αριθμός με 42 ψηφία.

Μετακινήσεις στο ΕΜΠ (0.25+0.25 = 0.5 βαθμοί)

Δεν ξέρω αν το προσέξατε, αλλά αυτές τις ημέρες διάφορες περιφράξεις και εμπόδια ξεφυτρώνουν σε διάφορα μέρη του campus του ΕΜΠ. Κάποιες φορές είναι δύσκολο να πας από το ένα σημείο στο άλλο με τη συνηθισμένη σου διαδρομή και πρέπει να βρεις εναλλακτικές.

Έστω ότι αναπαριστούμε το campus του ΕΜΠ με ένα grid από $2 \leq N \leq 100$ γραμμές και $2 \leq M \leq 100$ στήλες όπου κάποια από τα τετράγωνα του έχουν εμπόδια τα οποία τα καθιστούν απροσπέλαστα. Έστω επίσης ότι κάθε βήμα της διαδρομής γίνεται σε μια χρονική στιγμή και οδηγεί κάποιον σε κάποιο από τα (το πολύ) τέσσερα γειτονικά τετράγωνα από αυτό που βρισκόταν την αμέσως προηγούμενη χρονική στιγμή. Έστω επίσης ότι γνωρίζουμε ότι κάποιος έκανε μια διαδρομή διάρκειας T ξεκινώντας από το τετράγωνο με συντεταγμένες (Γ, Σ) και καταλήγοντας στο τετράγωνο με συντεταγμένες (Γ', Σ') . Το ζητούμενο είναι με πόσους διαφορετικούς τρόπους μπορεί κάποιος να έχει βρεθεί από το (Γ, Σ) στο (Γ', Σ') σε T χρονικές μονάδες με δεδομένο ότι κινείται συνέχεια κατά τη διάρκεια της μετακίνησής του. Ο χρόνος είναι διακριτός.

Η άσκηση σας ζητάει να γράψετε ένα πρόγραμμα σε C/C++ και ένα σε ML που να διαβάζει από ένα αρχείο κειμένου τα $N, M, T, \Gamma, \Sigma, \Gamma', \Sigma'$ και τις συντεταγμένες των εμποδίων στο campus του ΕΜΠ και να τυπώνει ως έξοδο τον αριθμό των διαφορετικών τρόπων με τους οποίους η συγκεκριμένη μετακίνηση θα μπορούσε να έχει γίνει. Προσέξτε ότι κάποιος μπορεί να έχει βρεθεί σε κάποιο τετράγωνο περισσότερες από μία φορές κατά τη διάρκεια της διαδρομής του.

Παρακάτω δίνονται δύο παραδείγματα εισόδου και οι αντίστοιχες εξοδοί:

		Σε C/C++, MLton	Σε SML/NJ
\$ cat grid1.txt	\$ cat grid2.txt	\$./walks grid1.txt	- walks "grid1.txt";
4 5 6	6 6 6	1	1
1 3 1 5	5 4 3 2	\$./walks grid2.txt	val it = () : unit
2	7	10	- walks "grid2.txt";
1 4	1 2		10
2 4	2 3		val it = () : unit
	2 4		
	3 1		
	4 2		
	4 4		
	6 3		

Εξηγούμε το πρώτο παράδειγμα. Η είσοδος του λέει ότι το grid του campus έχει 4 γραμμές και 5 στήλες, ότι σε 6 χρονικές στιγμές κάποιος μετακινήθηκε από τη θέση (1,3) στη θέση (1,5) δεδομένου ότι υπάρχουν 2 εμπόδια στις θέσεις με συντεταγμένες (1,4) και (2,4). Η αντίστοιχη έξοδος λέει ότι υπάρχει μία μόνο διαδρομή: αυτή γύρω από τα δύο εμπόδια.

Στο παράδειγμα του δεύτερου αρχείου εισόδου υπάρχουν 7 εμπόδια και οι δυνατές διαφορετικές διαδρομές είναι 10 σε αυτήν την περίπτωση.

Περαιτέρω οδηγίες για τις ασκήσεις

- Μπορείτε να δουλέψετε σε ομάδες το πολύ δύο ατόμων, τόσο σε αυτή όσο και στις επόμενες σειρές ασκήσεων. Όμως, έχετε υπ' όψη σας ότι, αν δεν περάσετε το μάθημα φέτος, οι βαθμοί των προγραμματιστικών ασκήσεων κρατούνται μόνο για όσους τις έκαναν σε μονομελή ομάδα (δηλ. τις έκαναν μόνοι τους).
- 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.
- Σε αντίθεση με το παραπάνω, επιτρέπεται να μοιράζεστε test cases.
- 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 C/C++ πρέπει να είναι σε ένα αρχείο και να μεταγλωττίζονται χωρίς warnings με `gcc/g++ 14.2.0` με εντολές της μορφής, π.χ.

```
gcc -std=c17 -Wall -Werror -O3 -lm -o maxprod maxprod.c  
g++ -std=c++17 -Wall -Werror -O3 -lm -o maxprod maxprod.cpp
```
- 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 5.3.0. Το σύστημα ηλεκτρονικής υποβολής σας επιτρέπει να επιλέξετε μεταξύ αυτών των διαλέκτων της ML.
- 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 άλλου είδους έξοδο διότι δεν θα γίνουν δεκτά από το σύστημα στο οποίο θα υποβληθούν.