

HY – 150 Εαρινό 2021

1^η Σειρα Ασκήσεων

26/2/2021 – 12/3/2021

Άσκηση 1 : Vector Magnitude

Φτιάξτε ένα πρόγραμμα που υπολογίζει το μέτρο ενός διανύσματος. Αρχικά, το πρόγραμμα σας θα πρέπει να ζητάει από τον χρήστη τις διαστάσεις του διανύσματος και στην συνέχεια τις συντεταγμένες του στον χώρο. Υποθέστε ότι οι διαστάσεις θα είναι 2 ή 3. Στην συνέχεια, ζητείται να φτιάξετε δύο συναρτήσεις με τις εξής υπογραφές :

- **double vector_magnitude(double a, double b)** (για διανύσματα 2 διαστάσεων)
- **double vector_magnitude(double a, double b, double c)** (για διανύσματα 3 διαστάσεων)

οι οποίες θα υπολογίζουν το μέτρο των διανυσμάτων και θα το επιστρέφουν. Τέλος, εκτυπώστε το μέτρο του διανύσματος. Το πρόγραμμα σας θα πρέπει να χρησιμοποιεί c++ exceptions για να χειρίζεται πιθανόν λάθη στο input από τον χρήστη.

Tips/Help : function sqrt() for calculating square root

2D vector magnitude formula: $\sqrt{x^2 + y^2}$

2D vector magnitude formula: $\sqrt{x^2 + y^2 + z^2}$

Άσκηση 2 : String Merging

Ζητείται πρόγραμμα το οποίο δέχεται δύο strings από τον χρήστη και δημιουργεί ένα τρίτο string που είναι η συγχώνευση αυτών των δύο. Η συγχώνευση θα πρέπει να γίνει παίρνοντας εναλλάξ ένα χαρακτήρα από κάθε input string και προσθέτοντας τον στο τέλος του καινούριου string. Στην περίπτωση που ένα από τα δύο input strings είναι μεγαλύτερο, οι περισσευούμενοι χαρακτήρες πρέπει να προστεθούν στο τέλος του output string χωρίς καμία μετατροπή. Τέλος, το πρόγραμμα σας πρέπει να εκτυπώνει στη γραμμή εντολών το αποτέλεσμα.

Ενδεικτικά παραδείγματα:

Input String 1	Input String 2	Output String
first	second	fsiercsotnd
longstring	short	lsohnogrstring
nc_upt	ieotu	nice_output
Long test.	Empty spaces can exist	LEomnpgt yt espsta.ces can exist

Άσκηση 3 : Horizontal Projection

Ζητείται ένα πρόγραμμα το οποίο λύνει το γνωστό πρόβλημα φυσικής, οριζόντια βολή. Υποθέστε ότι ένα σώμα κινείται με ευθύγραμμη ομαλή κίνηση, αρχική ταχύτητα v_0 (m/s) προς ένα γκρεμό με ύψος h (km). Υπολογίστε το χρόνο που θα κάνει το σώμα να χτυπήσει το έδαφος, και την απόσταση που θα διανύσει στον x άξονα.

Το πρόγραμμα σας πρέπει να ζητάει από τον χρήστη την αρχική ταχύτητα του σώματος και το ύψος του γκρεμού και στην συνέχεια να υπολογίζει και να εκτυπώνει την απόσταση που θα διανύσει σε km και τον χρόνο που θα βρίσκεται στον αέρα σε seconds. Το πρόγραμμα σας θα πρέπει να χρησιμοποιεί c++ exceptions για να χειρίζεται πιθανόν λάθη στο input από τον χρήστη.

Άσκηση 4 : Valid Parentheses

Ζητείται πρόγραμμα το οποίο δέχεται ένα string από τον χρήστη και ελέγχει εάν υπάρχει σωστό ταίριασμα παρενθέσεων. Συγκεκριμένα το πρόγραμμα σας θα πρέπει να ελέγχει εάν για όποια ανοιχτή παρένθεση που υπάρχει στο input string υπάρχει μία αντίστοιχη κλειστή. Τέλος, θα πρέπει να εκτυπώνει στην γραμμή εντολών εάν το string είναι valid ή όχι και στην περίπτωση που δεν είναι θα πρέπει να εκτυπώνει το σημείο στο string που εμφανίζεται το πρώτο λάθος.

Ενδεικτικά παραδείγματα:

Input String	Output
Th(is) (is a (v)al(id example.))	Correct.
(())(())(())	Correct.
((())	Wrong at position 2.
(This (is) (another wrong example.)	Wrong at position 11.
)((())	Wrong at position 0.
(()) (Wrong at position 4.

Στα παραδείγματα που έχουν μόνο παρενθέσεις έχουν προστεθεί επιπλέον κενά για να είναι πιο ευδιάκριτα και δεν λαμβάνονται υπόψιν στον υπολογισμό της λάθος θέσης. Το πρόγραμμα σας θα πρέπει να τα μετράει κανονικά.

Άσκηση 5 : Change Calculation

Ζητείται πρόγραμμα το οποίο θα υπολογίζει τα ρέστα μιας χρηματικής συναλλαγής. Αρχικά, το πρόγραμμα σας θα πρέπει να ζητάει από τον χρήστη τρία είδη νομισμάτων που έχει στην διάθεση του για να χρησιμοποιήσει ως ρέστα, για παράδειγμα 5, 20, 50 (cents). Στην συνέχεια θα επαναλαμβάνεται η εξής διαδικασία:

1. Θα ζητάει από τον χρήστη το κόστος του αντικειμένου που θέλει. Το κόστος θα είναι δεκαδικό (π.χ. 9.65 ευρώ)
2. Θα ζητάει από τον χρήστη να πληρώσει χρησιμοποιώντας 200 (2 ευρώ) 500 (5 ευρώ) ή 1000 (10 ευρώ), μέχρι να καλυφθεί η τιμή του κόστους.
3. Θα υπολογίζει και θα εκτυπώνει στην οθόνη πόσα νομίσματα, από τα τρία είδη που του έχουν δοθεί στην αρχή, πρέπει να δώσει πίσω στον χρήστη. Ζητείται να

βελτιστοποιήσετε αυτόν τον υπολογισμό ώστε να δίνει πάντα τον λιγότερο δυνατό αριθμό νομισμάτων.

Αυτή η διαδικασία θα επαναλαμβάνεται μέχρι ο χρήστης να δώσει κόστος = -1. Στην περίπτωση που τα ρέστα δεν γίνεται να υπολογιστούν ακριβώς με κάποιον συνδυασμό των διαθέσιμων νομισμάτων θα πρέπει να υπολογίζετε τον ποιο κοντινό συνδυασμό και να εκτυπώνετε πόσα χρήματα το πρόγραμμα σας δεν μπορεί να δώσει πίσω στον χρήστη. Τέλος, το πρόγραμμα σας θα πρέπει να ελέγχει για τυχόν λάθη στο input του χρήστη.

Ενδεικτικό παράδειγμα:

```
Please insert the available currency coins: 5 20 50
Please insert the cost of the product (insert -1 to stop) :9.85
Please pay using a 200 500 or 1000 bill : 500
Please pay using a 200 500 or 1000 bill : 500
Your change is 50x0 , 20x0 , 5x3
Please insert the cost of the product (insert -1 to stop) :9.82
Please pay using a 200 500 or 1000 bill : 1000
The vending machine cannot give you the following change 3
Your change is 50x0 , 20x0 , 5x3
Please insert the cost of the product (insert -1 to stop) :-1
```