

## ΗΥ – 150 Εαρινό 2022

### 2η Σειρά Ασκήσεων

18/3/2022 - 1/4/2022

#### Άσκηση 1 : Μαθηματικές Πράξεις (20%)

Ζητείται πρόγραμμα που αρχικά να ζητάει έναν ακέραιο αριθμό  $N$  από τον χρήστη και μετά να ζητάει  $N$  αριθμούς και θα τους αποθηκεύει σε έναν πίνακα (vector). Στην συνέχεια το πρόγραμμα θα ζητάει από τον χρήστη τι πράξη να γίνει σε αυτούς τους αριθμούς:

1. Πρόσθεση
2. Πολλαπλασιασμό
3. Αφαίρεση

Τέλος, με την χρήση **αναδρομικής** συνάρτησης θα πρέπει να υπολογίζετε την επιλεγμένη πράξη μεταξύ όλων των αποθηκευμένων αριθμών ξεκινώντας από το τέλος προς την αρχή, και να εκτυπώνει το κάθε βήμα. Ο χρήστης θα πρέπει να ελέγχει ότι ο αριθμός  $N$  είναι θετικός ακέραιος.

Π.Χ:

Για το input [3, 4, 3, 1, 11] και επιλογή της αφαίρεσης θα πρέπει να εκτυπωθεί:

1.  $11 - 1 = 10$
2.  $10 - 3 = 7$
3.  $7 - 4 = 3$
4.  $3 - 3 = 0$

#### Άσκηση 2 : Επισκευή Αμαξιών (30%)

Ζητείται να δημιουργηθεί η κλάση **Car**, η οποία έχει τα **private** πεδία *ProblemTires*, *PaintPercentage* και *ProblemLights*, τα οποία αντιστοιχούν στα λάστιχα που έχουνε πρόβλημα, στο ποσοστό βαφής που του λείπει και τα προβληματικά φώτα. Για αυτά τα πεδία θα πρέπει να φτιαχτούν setters και getters στα οποία οι setters θα ελέγχουν εάν οι τιμές που δίνονται είναι σωστές. Ένα αμάξι μπορεί να έχει μέχρι 4 λάστιχα, το *PaintPercentage* είναι ποσοστό από το 0 έως το 100 και ένα αμάξι έχει μέχρι 8 φώτα.

Επίσης, θα πρέπει να έχει 2 constructors με υπογραφές:

- **Car()** : δημιουργεί ένα ολοκαίνουριο αμάξι
- **Car(int Tires, int Paint, int Lights)** : δημιουργεί ένα μεταχειρισμένο αμάξι με τα προβλήματα που θα δοθούν ως arguments. Προσοχή! και εδώ πρέπει να γίνετε έλεγχος ότι είναι σωστές οι παράμετροι.

Επίσης η κλάση Car θα πρέπει να έχει δύο μεθόδους:

**void PrintCarStatus():**

Εκτυπώνει την κατάσταση του αμαξιού, όσον αφορά τα λάστιχα, την μπογιά και τα φώτα

**float CalculateRepairCosts(float TireCost, float PaintCost, float LightsCost):**

Υπολογίζει το συνολικό κόστος επισκευής του αμαξιού. Ένα αμάξι χρειάζεται βάψιμο εάν του λείπει πάνω από 30% μπογιά.

Τέλος, θα πρέπει να γράψετε ένα πρόγραμμα που αρχικά θα ζητάει από τον χρήστη τα κόστη για τα λάστιχα, το κόστος του βαψίματος και το κόστος για κάθε λάμπα. Στην συνέχεια θα φτιάχνετε 2 στιγμιότυπα (instances) από αμάξια, ένα καινούριο και ένα μεταχειρισμένο. Στο καινούριο θα του δημιουργήσετε προβλήματα με τις set συναρτήσεις του και στο τέλος θα εκτυπώσετε την κατάσταση και των 2 αμαξιών όπως και το κόστος τους να επισκευαστούν.

### Άσκηση 3 : Αγορά Ταινίας (30%)

Ζητείτε πρόγραμμα που αρχικά θα ζητάει από τους χρήστες το όνομα ενός αρχείου. Το αρχείο αυτό θα περιέχει ονόματα ταινιών, εάν τις έχει ο χρήστης (status) και την τιμή αγοράς τους. Αυτές οι 3 μεταβλητές θα χωρίζοντε με κενό, υποθέστε ότι στα ονόματα των ταινιών δεν υπάρχουν κενά. Οι χρήστες θα πρέπει να διαβάζουνε το αρχείο και να αποθηκεύουν τα δεδομένα αυτά σε πίνακες. Στην συνέχεια να εκτυπώνουν όλη την πληροφορία του αρχείου στον χρήστη και να του δίνουν την επιλογή να επιλέξει μία ταινία και να την αγοράσει ή να την νοικιάσει. Αυτή η διαδικασία πρέπει να επαναλαμβάνετε μέχρι ο χρήστης να δώσει input -1 ή exit στην επιλογή ταινίας. Τέλος, θα πρέπει να ξαναγράφετε όλα τα δεδομένα των ταινιών στο αρχείο με ανανεωμένο το status της κάθε ταινίας και να εκτυπώνετε το συνολικό κόστος που χρωστάει ο χρήστης.

Διευκρινίσεις:

- Το status μπορεί να έχει τρεις τιμές, 0 (not owned), 1 (rent), 2 (bought)
- Η ενοικίαση μιας ταινίας κοστίζει 3 ευρώ

### Άσκηση 4 : Βιβλιοθήκη για Strings (20%)

Δημιουργήστε μια στατική βιβλιοθήκη (στον ίδιο φάκελο του ερωτήματος αυτού), η οποία θα περιλαμβάνει μια κλάση MyString που θα έχει κατάλληλες συναρτήσεις-μέλη που θα κάνουν διάφορες πράξεις με strings. Οι πράξεις αυτές είναι:

1. Concatanation: ένωση δύο strings
2. Compare: σύγκριση των δύο strings
3. ToUppper: μετατροπή του string σε κεφαλαία γράμματα
4. ToLower: μετατροπή του string σε μικρά σε μικρά γράμματα

Τέλος,

- a) Χτίστε και συνδέστε (build & link) την στατική αυτή βιβλιοθήκη μέσω του cmake.
- b) Υλοποιήστε παραδείγματα που την καλούν (ένα για κάθε συνάρτηση-μέλος της κλάσης).

