



## Θέμα 1 [3 μονάδες]

A. Κατασκευάστε μια templated συνάρτηση με όνομα `equal_or_better()` που να δέχεται ως όρισμα, με όνομα `n`, ένα διάνυσμα (`std::vector`) στοιχείων τύπου `T`, και ένα επιπλέον όρισμα τύπου `T` με ονόματα `base`. Η συνάρτηση να επιστρέφει το πλήθος των στοιχείων του διανύσματος που είναι μεγαλύτερα ή ίσα του `base`.

B. Από τη συνάρτηση `main()`, να κληθεί η συνάρτηση `equal_or_better()` και να εμφανίζει το αποτέλεσμα που επιστρέφει:

- Για έναν διάνυσμα ακεραίων.
- Για ένα διάνυσμα με αντικείμενα `monie` (ταινία), όπου κάθε αντικείμενο έχει τα πεδία `title` (`string`, τίτλος ταινίας), και `imdb_score` (`double`, βαθμός ταινίας). Η σύγκριση των αντικειμένων `monie` να γίνεται με βάση την τιμή του `imdb_score` υπερφορτώνοντας τον τελεστή `>=`.

Μπορείτε να χρησιμοποιήσετε τις ακόλουθες ταινίες: {"The Godfather", 9.2}, {"Pulp Fiction", 8.9}, {"Parasite", 8.5}, {"Casino", 8.2}, {"The Matrix", 8.7}

## Θέμα 2 [4 μονάδες]

A. Γράψτε μια βασική κλάση με όνομα `vehicle` (όχημα) που να περιέχει τα πεδία `year` (`int`, έτος κατασκευής) και `price` (`double`, τιμή) και έναν κατασκευαστή με παραμέτρους για τα πεδία `year` και `price`. Επίσης, η ίδια κλάση να διαθέτει μια `pure virtual` συνάρτηση με όνομα `info()`.

B. Γράψτε μια παράγωγη κλάση της `vehicle` με όνομα `car` (`string`, αυτοκίνητο) που να περιέχει το πεδίο `cc` (`double`, κυβισμός). Η κλάση διαθέτει κατασκευαστή (με παραμέτρους για όλα τα πεδία της) και υλοποίηση για τη συνάρτηση `info()` που εμφανίζει τις τιμές όλων των πεδίων.

Γ. Γράψτε μια παράγωγη κλάση της `car` με όνομα `sports_car` (σπορ αυτοκίνητο) που περιέχει το πεδίο `hp` (`double`, ιπποδύναμη). Η κλάση διαθέτει κατασκευαστή (με παραμέτρους για όλα τα πεδία της) και υλοποίηση για τη συνάρτηση `info()` που εμφανίζει τις τιμές όλων των πεδίων.

Δ. Δημιουργήστε στη `main()` ένα διάνυσμα (`std::vector`) με στοιχεία τύπου `vehicle*`, που να περιέχει 2 δείκτες προς αντικείμενα `car` και 1 δείκτη προς αντικείμενο `sports_car`. Διασχίστε το διάνυσμα και εμφανίστε με την `info()` τα στοιχεία των αντικειμένων. Η δημιουργία των αντικειμένων να γίνεται δυναμικά με τον τελεστή `new`.

Θέμα 3 [3 μονάδες]

A. Γράψτε μια κλάση με όνομα `ebook` που να περιέχει τα πεδία `title` (string, τίτλος βιβλίου), `authors` (string, συγγραφείς βιβλίου), `file_size` (int, μέγεθος ebook σε bytes). Συμπληρώστε έναν κατασκευαστή που να δέχεται τιμές και για τα τρία πεδία.

B. Συμπληρώστε getters για όλα τα πεδία. Ο getter για το πεδίο `file_size` να δέχεται μια παράμετρο με όνομα `magnitude` (string) που αν λαμβάνει την τιμή `"b"` (προκαθορισμένη τιμή) να επιστρέφει το μέγεθος του αρχείου σε bytes, αν λαμβάνει την τιμή `"kb"` να επιστρέφει το μέγεθος του αρχείου σε kilobytes ( $1\text{kb}=1024$  bytes), αν λαμβάνει την τιμή `"mb"` να επιστρέφει το μέγεθος του αρχείου σε megabytes ( $1\text{mb}=1048576$  bytes) και αν λαμβάνει οποιαδήποτε άλλη τιμή να παράγει μια εξαίρεση κάνοντας `throw` το αλφαριθμητικό `"invalid magnitude specifier"`.

Γ. Στη `main()` θεωρήστε ότι ορίζεται ο ακόλουθος πίνακας με 3 αντικείμενα `ebook`:

```
ebook a[3] = {ebook("The Pragmatic Programmer", "Andrew Hunt and David  
Thomas", 1234567), ebook("Clean Code", "Robert C. Martin", 12345678),  
ebook("Introduction to the Theory of Computation", "Sipser",  
12345678)};
```

Συμπληρώστε κώδικα που να ζητά από το χρήστη να εισάγει τον τρόπο με τον οποίο επιθυμεί να εμφανίζονται τα μεγέθη των ebooks με επιλογές `b`, `kb` και `mb`. Να εμφανίζει τον τίτλο, τους συγγραφείς και το μέγεθος του κάθε βιβλίου σε bytes, kilobytes ή megabytes ανάλογα με την επιλογή του χρήστη. Αν δημιουργείται εξαίρεση, να γίνεται `catch`, να εμφανίζεται ένα μήνυμα που να πληροφορεί το χρήστη για το λόγο που συμβαίνει η εξαίρεση και το πρόγραμμα να τερματίζει.