

### Θέμα 1 [A=3, B=1]

A. Κατασκευάστε μια templated συνάρτηση που να δέχεται ένα διάνυσμα `std::vector` και να επιστρέφει τις δύο μεγαλύτερες τιμές του διανύσματος (bonus + 1 μονάδα για εύρεση των δύο μεγαλύτερων τιμών χωρίς να ταξινομηθούν τα δεδομένα). Να κληθεί η συνάρτηση:

1. Για ένα διάνυσμα ακεραίων.
2. Για ένα διάνυσμα με αντικείμενα `student` (σπουδαστής) όπου κάθε `student` διαθέτει το πεδίο `name` (όνομα) και το πεδίο `grade` (βαθμός). Η σύγκριση των σπουδαστών να γίνει με υπερφόρτωση του τελεστή `>`.

B. Να εμφανιστούν τα αποτελέσματα. Για την περίπτωση των αντικειμένων `student` να υπερφορτωθεί ο τελεστής `<<`.

### Θέμα 2 [A=1, B=1, Γ=1, Δ=1, Ε=1, ΣΤ=1]

A. Κατασκευάστε την ακόλουθη ιεραρχία. Την υπερκλάση `person` (άτομο) με πεδίο δεδομένων `age` (ηλικία) και τις υποκλάσεις της `person`: `teacher` (καθηγητής) με πεδίο δεδομένων `profession` (εξειδίκευση) και `footballer` (ποδοσφαιριστής) με πεδίο δεδομένων `team` (ομάδα).

B. Δημιουργήστε από έναν κατασκευαστή σε κάθε κλάση και έναν `virtual` καταστροφέα στην κλάση `person`.

Γ. Προσθέστε τις κατάλληλες μεθόδους έτσι ώστε να μπορεί να κληθεί η συνάρτηση `compute_earnings` (υπολογισμός εσόδων) και να επιστρέφει την τιμή `1000.0` για τα αντικείμενα `teacher` και `100000.0` για τα αντικείμενα `footballer`.

Δ. Προσθέστε στην κλάση `person` στατική μεταβλητή με όνομα `number_of_instances` που θα καταμετρά το πλήθος των αντικειμένων τύπου `person` (ή των υποτύπων της) που θα δημιουργούνται. Εμφανίστε το πλήθος στη `main`.

Ε. Στη `main`, δημιουργήστε έναν πίνακα 5 θέσεων που ο χρήστης θα γεμίζει με δείκτες προς αντικείμενα `teacher` ή `footballer` πραγματοποιώντας εισαγωγή τιμών από το πληκτρολόγιο.

ΣΤ. Στη `main`, καλέστε τη συνάρτηση `compute_earnings` για κάθε στοιχείο του πίνακα και εμφανίστε τα αποτελέσματα.