

Δημιουργήστε ένα αρχείο C++ με όνομα main.cpp. Απαντήστε στα παρακάτω θέματα γράφοντας τον κατάλληλο κώδικα μέσα στο παραπάνω αρχείο. Στο τέλος της εξέτασης, ανεβάζετε το αρχείο main.cpp στην εργασία σας.

### Θέμα Α – 1.5 μονάδα

Εισάγετε στο αρχείο main.cpp την κεφαλίδα random. Στη συνέχεια γράψτε μια συνάρτηση void TestMemory() που να δεσμεύει έναν πίνακα από N δείκτες σε int, όπου  $N = \text{rand}() \% 1000 + 1$ . Σε κάθε κελί  $i = 0 \dots N - 1$  του πίνακα, δεσμεύστε έναν πίνακα 10000 θέσεων από δεδομένα τύπου int και αρχικοποιήστε τα στην τιμή 1.

Στην ίδια συνάρτηση, αποδεσμεύστε τη μνήμη που δεσμεύσατε.

### Θέμα Β – 1.5 μονάδες

Δημιουργήστε μια συνάρτηση void TestFileSave(). Σε αυτή, γράψτε με δυαδικό τρόπο σε ένα αρχείο με όνομα exam.bin α) την ημέρα των εξετάσεων σε μορφή unsigned short, β) το μήνα της εξέτασης σε μορφή unsigned short, γ) το έτος της εξέτασης σε μορφή unsigned short και δ) το ονοματεπώνυμό σας με Λατινικούς χαρακτήρες.

Στη συνέχεια, δημιουργήστε μια συνάρτηση TestFileLoad() στην οποία εξάγετε τα δεδομένα που γράψατε στο αρχείο exam.bin και τα τυπώνετε στην κονσόλα.

### Θέμα Γ – 2.0 μονάδες

Δημιουργήστε μια template κλάση Pipeline με template argument T που να υλοποιεί μια απλή σωλήνωση διεργασιών, στην οποία ισχύουν τα ακόλουθα:

- Εισάγουμε ένα στοιχείο τύπου T με τον τελεστή <<, αφού πρώτα πολλαπλασιάσουμε την τιμή του δεδομένου X 2.
- Εξάγουμε ένα στοιχείο τύπου T με τον τελεστή >>. Το στοιχείο αφαιρείται από το Pipeline.
- Τα δεδομένα εξέρχονται ταξινομημένα, ανεξάρτητα από τη σειρά που μπήκαν και υπάρχουν μόνο μια φορά μέσα στη σωλήνωση κάθε δεδομένη στιγμή.
- Εξάγεται πρώτο το μικρότερο στοιχείο.
- Ο τελεστής >> διαχειρίζεται ομαλά την μη ύπαρξη κάποιου στοιχείου.

Γράψτε μια συνάρτηση ελέγχου της παραπάνω λειτουργικότητας με υπογραφή void TestPipeline() που:

- Να δηλώνει μια μεταβλητή τύπου Pipeline που να περιέχει στοιχεία τύπου int.
- Να εισάγει τα στοιχεία 2, 4, 1, 2 σε αυτή.
- Να εξάγει και τυπώνει ένα ένα τα στοιχεία που προκύπτουν.

### Θέμα Δ – 1.0 μονάδα

Πραγματοποιήστε οποιαδήποτε απαραίτητη αλλαγή στην κλάση Pipeline και επιπλέον δημιουργήστε μια παράγωγη κλάση από την Pipeline με όνομα GenericPipeline που να την εξειδικεύει για T=float και επιπλέον να παρέχει τη δυνατότητα να εφαρμόζει οποιαδήποτε πράξη πάνω στα στοιχεία που εισέρχονται σε αυτή με τον τελεστή <<. Η πράξη αυτή θα πρέπει να περνιέται στο στιγμιότυπο μιας GenericPipeline, με μια κατάλληλη μέθοδο, ως std::function F που να δέχεται και επιστρέφει δεδομένα συμβατά με το περιεχόμενο της GenericPipeline. Η F εφαρμόζεται πριν αποθηκευτεί εσωτερικά ένα στοιχείο x στα δεδομένα που κρατάει η Pipeline, άρα αποθηκεύει το F(x).

Γράψτε μια συνάρτηση ελέγχου της παραπάνω λειτουργικότητας με υπογραφή void TestGenericPipeline() που:

- Να δηλώνει μια μεταβλητή τύπου `GenericPipeline`.
- Να περνάει ως συνάρτηση `F` την `fabsf()` (υπάρχει στο header `math.h`).
- Να εισάγει τα στοιχεία `-4.0f`, `2.0f`, `4.0f`, `-2.0f` σε αυτή.
- Να εξάγει και τυπώνει ένα ένα τα στοιχεία που προκύπτουν.

Η `main` συνάρτηση του προγράμματος θα πρέπει να καλεί όλες τις παραπάνω συναρτήσεις που φτιάξατε:

```
int main()
{
    TestMemory();
    TestFileSave();
    TestFileLoad();
    TestPipeline();
    TestGenericPipeline();
    return 0;
}
```