



5^ο ΦΥΛΛΑΔΙΟ ΑΣΚΗΣΕΩΝ

ΑΣΚΗΣΗ - 1

Να δημιουργήσετε μια ιεραρχία κλάσεων (δηλ. μια βασική και τις απαραίτητες απορρέουσες κλάσεις) με όνομα βασικής κλάσης `vehicle`. Οι απορρέουσες κλάσεις θα είναι : `truck`, `car`, `bus` και `motorcycle`. Όλα τα αντικείμενα του τύπου `vehicle` θα έχουν ως δεδομένα :

- Αριθμό κυκλοφορίας (`long`)
- Βάρος (`float`)

Για κάθε απορρέουσα κλάση θα υπάρχουν επιπρόσθετα τα εξής `data members`:

1. Τα φορτηγά (`trucks`) θα έχουν επιπλέον το ωφέλιμο φορτίο (`float`).
2. Τα αυτοκίνητα (`cars`) θα έχουν επιπλέον τον τύπο (έναν ακέραιο τετραψήφιο θετικό αριθμό)
3. Τα λεωφορεία (`buses`) θα έχουν επιπλέον τον αριθμό επιβατών (`int`)
4. Τα `motorcycles` δεν θα έχουν επιπλέον δεδομένα.

Σε κάθε κλάση (βασική και απορρέουσες) θα υπάρχουν:

- Ένας `constructor` με λίστα παραμέτρων για εισαγωγή δεδομένων
- Μια συνάρτηση για την εμφάνιση των δεδομένων.

Η συνάρτηση `main()` θα χρησιμοποιεί ένα αντικείμενο (`object`) από κάθε κλάση για την εισαγωγή και την εμφάνιση των δεδομένων.

ΑΣΚΗΣΗ - 2

Χρησιμοποιήστε το παρακάτω πρόγραμμα και τροποποιήστε την κλάση `manager` καθώς και τις κλάσεις `foreman` και `laborer` σύμφωνα με τις ακόλουθες οδηγίες:

Οι `managers` επιβλέπουν όλους τους εργοδηγούς (`foremen`) σε μια βιομηχανία. Όπως όλοι οι εργαζόμενοι έτσι και οι `managers` έχουν όνομα και κωδικό αριθμό και όπως οι εργοδηγοί έχουν `quotas` (ποσοστώσεις) ως `data member` που είναι ο μέσος όρος των ποσοστώσεων (`quotas`) όλων των εργοδηγών στη βιομηχανία. Επιπλέον έχουν ένα `data member` με όνομα `absenteeism` (συστηματική απουσία), τύπου `float`, που είναι η μέση τιμή % του χρόνου εργασίας που θεωρείται νεκρός χρόνος (χρόνος συστηματικής απουσίας) για όλους τους εργοδηγούς (`foremen`) καθώς και για τους εργάτες (`laborers`) που επιβλέπουν οι εργοδηγοί.

Ζητούνται :

1. Να τροποποιήσετε κατάλληλα τις κλάσεις `foreman` και `laborer`
2. Να τοποθετήσετε την κλάση `manager` στο κατάλληλο σημείο στο πρόγραμμα.
3. Η συνάρτηση `main()` θα ορίζει :
 - Για κάθε μια από τις κλάσεις `laborer` και `foreman` ένα μονοδιάστατο πίνακα από `objects` με πλήθος στοιχείων `MAX=5` και θα χρησιμοποιεί τις αντίστοιχες `member functions` για εισαγωγή και εμφάνιση δεδομένων
 - για την κλάση `scientist` ένα αντικείμενο και θα χρησιμοποιεί τις αντίστοιχες `member functions` για εισαγωγή και εμφάνιση δεδομένων.
 - Για την κλάση `manager` ένα αντικείμενο και θα χρησιμοποιεί τις αντίστοιχες `member functions` για εισαγωγή και εμφάνιση δεδομένων. Ο `manager` θα επιβλέπει όλους τους εργοδηγούς και τους εργάτες και θα χρεώνεται όλες τις αντίστοιχες ποσοστώσεις (`quotas`) και τους νεκρούς χρόνους (χρόνοι συστηματικής απουσίας κατά τους οποίους δεν εργάζονται).

```

#include <iostream>
using namespace std;
const int LEN = 80; // maximum length of names
class employee
{
private:
    char name[LEN]; // employee name
    unsigned long number; // employee number
public:
    void getdata()
    {
        cout << "\n    Enter last name: "; cin >> name;
        cout << "    Enter number: "; cin >> number;
    }
    void putdata()
    {
        cout << "\n    Name = " << name;
        cout << "\n    Number = " << number;
    }
};

class manager : public employee // management class
{
private:
    char title[LEN]; // "vice-president" etc.
    double dues; // golf club dues
public:
    void getdata()
    {
        employee::getdata();
        cout << "    Enter title: "; cin >> title;
        cout << "    Enter golf club dues: "; cin >> dues;
    }
    void putdata()
    {
        employee::putdata();
        cout << "\n    Title: " << title;
        cout << "\n    Golf club dues: " << dues;
    }
};

class scientist : public employee // scientist class
{
private:
    int pubs; // number of publications
public:
    void getdata()
    {
        employee::getdata();
        cout << "    Enter number of pubs: "; cin >> pubs;
    }
    void putdata()
    {
        employee::putdata();
        cout << "\n    Number of publications: " << pubs;
    }
};

class laborer : public employee // laborer class

```

```

{
};
class foreman : public laborer // foreman class
{
private:
    float quotas; // percent of quotas met successfully
public:
    void getdata()
    {
        laborer::getdata();
        cout << "    Enter quotas: "; cin >> quotas;
    }
    void putdata()
    {
        laborer::putdata();
        cout << "\n    Quotas: " << quotas << endl;
    }
};
void main()
{
    laborer labo;
    foreman fore;
    cout << endl;
    cout << "\nEnter data for laborer";
    labo.getdata();
    cout << "\nEnter data for foreman";
    fore.getdata();
    cout << endl;
    cout << "\nData on laborer";
    labo.putdata();
    cout << "\nData on foreman";
    fore.putdata();
}

```

ΑΣΚΗΣΗ – 3

Μια μονάδα παραγωγής διαθέτει αποθέματα από N διαφορετικά προϊόντα (να θεωρήσετε το N ως μια σταθερά στην αρχή του προγράμματος). Τα αποθέματα κάθε προϊόντος έχουν παραχθεί σταδιακά, σε διαφορετικές γενικά ημερομηνίες. Για κάθε ποσότητα αποθέματος ενός προϊόντος είναι γνωστά τα εξής :

- Κωδικός προϊόντος (επιτρεπτές τιμές 1 έως N, ακέραιες)
- Ποσότητα αποθέματος (θετικός ακέραιος αριθμός)
- Ημερομηνία παραγωγής (ακέραιος αριθμός στη μορφή MMEE όπου EE = έτος και MM = αριθμός μήνα (1-12) (π.χ. 819 → Αύγουστος 2019). Αποδεκτά έτη από 2010 έως και 2019.
- Κόστος διατήρησης της ποσότητας αποθέματος (θετικός ακέραιος αριθμός τύπου long)

Να γράψετε τον ορισμό μιας κλάσης με το όνομα stock με μέλη-δεδομένα τα παραπάνω.

Η συνάρτηση main() :

1. θα δημιουργεί έναν πίνακα αντικειμένων με όνομα *alfa* της κλάσης *stock* μεγέθους *M* (η τιμή του *M* να οριστεί ως σταθερά στην αρχή του προγράμματος). Στη συνέχεια θα εισάγει δεδομένα στον πίνακα αντικειμένων **ΜΟΝΟ** με χρήση συνάρτησης *constructor* (δομητή) **με τους απαραίτητους ελέγχους εγκυρότητας τιμών.** Συνιστάται η κατάλληλη χρήση γεννήτριας τυχαίων αριθμών.
2. θα βρίσκει και θα εμφανίζει για κάθε προϊόν ξεχωριστά το συνολικό κόστος διατήρησης καθώς και τη συνολική ποσότητα του αποθέματος.
3. η μονάδα θα πρέπει να διακόψει την παραγωγή ενός προϊόντος εφόσον η συνολική ποσότητα αποθεμάτων που έχουν παραχθεί πριν από τις 1/11/2019 υπερβαίνουν το 50% της συνολικής ποσότητας αποθεμάτων του. Χρησιμοποιώντας μια επαναληπτική διαδικασία, σε όλα τα αντικείμενα του πίνακα, να βρείτε και να εμφανίσετε (αν υπάρχουν) τους κωδικούς προϊόντων για τα οποία θα πρέπει άμεσα να διακοπεί η παραγωγή τους.

(ΥΠΟΔΕΙΞΗ : όσες συναρτήσεις – μέλη της κλάσης *stock* απαιτηθούν κατά την ανάπτυξη του προγράμματος μπορούν να γραφούν σε οποιοδήποτε σημείο με χρήση των κατάλληλων συμβολισμών)

ΣΤΗ ΣΥΝΕΧΕΙΑ ΚΑΙ ΧΩΡΙΣ ΝΑ ΞΑΝΑΓΡΑΨΕΤΕ ΚΩΔΙΚΑ ΠΟΥ ΗΔΗ ΥΠΑΡΧΕΙ!!!

να δημιουργήσετε μια απορρέουσα κλάση με όνομα *new_stock* που θα περιλαμβάνει ως **επιπλέον private μέλη – δεδομένα :**

- ημερομηνία λήξης της ποσότητας παραγωγής, στην ίδια μορφή ΜΜΕΕ
- κόστος παραγωγής της μονάδας προϊόντος (θετικός αριθμός τύπου *float*)

Στη συνέχεια:

1. να γράψετε τη συνάρτηση του *constructor* (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης *new_stock* **ΜΕ ΤΟΝ ΑΠΑΡΑΙΤΗΤΟ ΕΛΕΓΧΟ εγκυρότητας τιμών.**
2. να γράψετε τη συνάρτηση – μέλος της κλάσης *new_stock* που θα επιστρέφει για ένα προϊόν τη συνολική ποσότητα παραγωγής που έχει λήξει με βάση τον τρέχοντα μήνα (1219) μαζί με το αντίστοιχο κόστος παραγωγής. Ο κωδικός προϊόντος θα αποτελεί όρισμα εισόδου για τη συνάρτηση.
3. Να ορίσετε στο συνάρτηση *main()* ένα πίνακα αντικειμένων με όνομα *beta* της κλάσης *new_stock* μεγέθους *L* (η τιμή του *L* να οριστεί ως σταθερά στην αρχή του προγράμματος, πρέπει να είναι $L \leq M$).
4. Να κάνετε εισαγωγή δεδομένων στον πίνακα *beta* με χρήση *constructor* χρησιμοποιώντας (αντιγράφοντας) τα *L* πρώτα αντικείμενα του πίνακα *alfa* καθώς και τη γεννήτρια των τυχαίων αριθμών για τα επιπλέον δεδομένα.
5. Χρησιμοποιώντας μια επαναληπτική διαδικασία, σε όλα τα αντικείμενα του πίνακα *beta*, να βρείτε και να εμφανίσετε (αν υπάρχουν), για κάθε προϊόν ξεχωριστά, τη συνολική ποσότητα που έχει λήξει με βάση τον τρέχοντα μήνα (1219) καθώς και το αντίστοιχο συνολικό κόστος παραγωγής.

ΑΣΚΗΣΗ – 4

Ένας αγωγός μεταφοράς Φυσικού Αερίου (ΦΑ) διαθέτει κατά μήκος του σημεία ελέγχου του δικτύου. Σε τακτά χρονικά διαστήματα, που διαφέρουν ανάλογα με το σημείο ελέγχου, διενεργούνται προληπτικοί έλεγχοι για την διαπίστωση της κατάστασης του αγωγού, μέσω μετρητικών οργάνων που είναι κοινά σε όλα τα σημεία ελέγχου. Για κάθε σημείο ελέγχου τα δεδομένα είναι :

- Κωδικός σημείου ελέγχου – ακέραιος θετικός αριθμός τύπου int [όρια 1-999]
- Συχνότητα προληπτικών ελέγχων σε ημέρες - ακέραιος θετικός αριθμός τύπου int [όρια 1-365]
- Κατάσταση καθενός από τα 10 διαφορετικά μετρητικά όργανα του σημείου ελέγχου – επιτρεπτές τιμές 1 ή 2 ή 3 ή 4.

Να γραφεί μια κλάση με όνομα checkpoint και private μέλη – δεδομένα τα προαναφερόμενα δεδομένα.

Στη συνάρτηση main() :

1. να οριστεί ένας πίνακας αντικειμένων της κλάσης checkpoint N θέσεων ($N =$ γνωστό, $N \leq 99$) και να αρχικοποιηθούν όλες οι θέσεις του πίνακα με χρήση default constructor, θέτοντας μηδενικές τιμές για όλα τα δεδομένα.
2. Να γίνει εισαγωγή δεδομένων στον πίνακα αντικειμένων της checkpoint μέσω δομητή (constructor) της κλάσης, χρησιμοποιώντας μόνον τυχαίους αριθμούς, στα όρια τιμών που έχουν δοθεί.
3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα:
 - a. Να βρείτε και να εμφανίσετε σε πόσα σημεία ελέγχου καθένα από τα μετρητικά όργανα βρίσκεται σε κατάσταση 1 (δηλ. 10 τιμές, μία για κάθε μετρητικό όργανο).
 - b. Να δημιουργήσετε και να εμφανίσετε ένα δυναμικό πίνακα που θα περιλαμβάνει τους κωδικούς των σημείων ελέγχου που έχουν συχνότητα προληπτικών ελέγχων μικρότερη των 30 ημερών και τουλάχιστον 5 μετρητικά όργανα με κατάσταση 4.
4. ΣΤΗ ΣΥΝΕΧΕΙΑ ΚΑΙ ΧΩΡΙΣ ΝΑ ΞΑΝΑΓΡΑΨΕΤΕ ΚΩΔΙΚΑ ΠΟΥ ΗΔΗ ΥΠΑΡΧΕΙ!!! να δημιουργήσετε μια απορρέουσα κλάση με όνομα extended_checkpoint που θα περιλαμβάνει ως **επιπλέον private μέλη – δεδομένα:**
 - Το κόστος του προληπτικού ελέγχου κάθε μετρητικού οργάνου (θετικοί αριθμοί τύπου double)
 - Τον προβλεπόμενο χρόνο εκτέλεσης του προληπτικού ελέγχου κάθε μετρητικού οργάνου (θετικοί αριθμοί τύπου double)

Στη συνέχεια να γράψετε :

τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της κλάσης extended_checkpoint χρησιμοποιώντας μόνον τυχαίους αριθμούς **και θέτοντας μόνοι σας τα αντίστοιχα άνω όρια τιμών.**

ΑΣΚΗΣΗ – 5

Ένα δίκτυο αγωγών μεταφοράς φυσικού αερίου τροφοδοτεί εγκαταστάσεις υψηλής πίεσης και εγκαταστάσεις μέσης πίεσης. Υπάρχουν συνολικά 5 εγκαταστάσεις υψηλής πίεσης και 5 εγκαταστάσεις μέσης πίεσης. Σε ένα σειριακό αρχείο με όνομα **gas.txt** περιέχονται δεδομένα για τις ποσότητες μεταφοράς για κάθε ημέρα.

Κάθε γραμμή του αρχείου περιέχει τις εξής πληροφορίες :

- Ημερομηνία στη μορφή HHMM όπου HH = ημέρα [1,30] και MM = μήνας [1,12]
- Είδος εγκατάστασης (0=υψηλής πίεσης, 1=μέσης πίεσης)
- Ποσότητα τροφοδοσίας σε m³ για κάθε μία από τις εγκαταστάσεις (5 αριθμοί τύπου float, >=0.0)

Μια τυχαία γραμμή του αρχείου είναι η εξής (οι επικεφαλίδες δεν υπάρχουν στο αρχείο):

ΗΜΕΡΟΜΗΝΙΑ	ΕΙΔΟΣ ΕΓΚΑΤΑΣΤ.	ΠΟΣΟΤΗΤΑ
1405	1	17.2 16.9 25.7 8.0 27.7

Ζητούνται να υλοποιηθούν τα παρακάτω :

1. Να γραφεί ο ορισμός μιας κλάσης με όνομα `station` που θα περιέχει ως private μέλη – δεδομένα όλα τα στοιχεία μιας γραμμής του αρχείου.
2. Στη συνάρτηση `main()` να οριστεί ένας πίνακας αντικειμένων της κλάσης `station` `N` θέσεων (`N=γνωστό`). Να αρχικοποιηθούν όλες οι θέσεις του πίνακα με χρήση `default constructor`, σύμφωνα με τα παραπάνω, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα δεδομένα.
3. Να γραφεί ένας δομητής (`constructor`) της κλάσης `station` για την εισαγωγή δεδομένων στον πίνακα αντικειμένων της κλάσης. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου).
4. Για να εισαχθεί μια γραμμή του αρχείου στον πίνακα αντικειμένων θα πρέπει η συνολική ποσότητα και για τις 5 εγκαταστάσεις να είναι ≥ 10.0 . Αν ο συνολικός αριθμός γραμμών του αρχείου **gas.txt** που θα εισαχθούν στον πίνακα είναι μεγαλύτερος του `N` τότε, για τις γραμμές που ικανοποιούν τον παραπάνω περιορισμό και δεν θα καταχωρηθούν στον πίνακα : το είδος της εγκατάστασης και η αντίστοιχη συνολική ποσότητα και για τις 5 εγκαταστάσεις θα καταχωρούνται αντιστοίχως σε δύο μονοδιάστατους δυναμικούς πίνακες.
5. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα να βρεθούν και να εμφανιστούν οι συνολικές ποσότητες ανά μήνα και είδος εγκατάστασης (δηλ. συνολικά 24 τιμές, 12 για υψηλή πίεση και 12 για μέση πίεση).
6. Χρησιμοποιώντας τα στοιχεία των δυναμικών πινάκων να βρεθούν οι συνολικές ποσότητες που έχουν καταχωρηθεί για κάθε είδος εγκατάστασης (δηλ. δύο τελικά σύνολα ποσοτήτων).

ΣΤΗ ΣΥΝΕΧΕΙΑ ΚΑΙ ΧΩΡΙΣ ΝΑ ΞΑΝΑΓΡΑΨΕΤΕ ΚΩΔΙΚΑ ΠΟΥ ΗΔΗ ΥΠΑΡΧΕΙ!!!

Να δημιουργήσετε μια απορρέουσα κλάση με όνομα `new_station` που θα περιλαμβάνει ως επιπλέον private μέλη – δεδομένα:

- Το κόστος λειτουργίας κάθε εγκατάστασης για τη συγκεκριμένη ημερομηνία (πέντε τιμές τύπου float ≥ 0.0)

Στη συνέχεια να γράψετε :

τη συνάρτηση του `constructor` (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της κλάσης `new_station` **ΜΕ ΤΟΝ ΑΠΑΡΑΙΤΗΤΟ ΕΛΕΓΧΟ εγκυρότητας τιμών.**

ΑΣΚΗΣΗ – 6

Μια βιομηχανική μονάδα επιθεωρεί το μηχανολογικό της εξοπλισμό σύμφωνα με τις οδηγίες των κατασκευαστών των μηχανών. Για κάθε μηχανή και για κάθε επιθεώρηση που γίνεται από το έτος 2010 μέχρι και το έτος 2019 είναι γνωστά τα εξής δεδομένα: **ΥΠΟΔΕΙΞΗ:** για κάθε μηχανή υπάρχουν πολλές επιθεωρήσεις σε κάθε έτος)

- Κωδικός μηχανής – ακέραιος θετικός αριθμός τύπου int [όρια 1-99] (**ΥΠΟΔΕΙΞΙΣ**: 1. Δεν υπάρχουν όλοι οι κωδικοί από το 1-99, δηλ. το ακριβές πλήθος των μηχανών δεν είναι γνωστό και εάν απαιτηθεί πρέπει να προσδιοριστεί. 2. Επίσης, εάν ζητηθεί, θα πρέπει να προσδιοριστεί και το σύνολο των υπαρχόντων κωδικών μηχανών).
- Ημερομηνία ελέγχου στη μορφή ΜΜΕΕ (ΜΜ = μήνας, ΕΕ = έτος). Επιτρεπτές τιμές: $1 \leq MM \leq 12$ και για το έτος το εύρος [2010-2019].
- Κατάσταση της μηχανής σε 10 διαφορετικά σημεία επιθεώρησης με επιτρεπτές τιμές 0 και 1 (0 → καλή κατάσταση και 1 → κακή κατάσταση).

Να γραφεί μια κλάση με όνομα `i_mach` και private μέλη – δεδομένα τα παραπάνω.

Στη συνάρτηση `main()`:

1. να οριστεί ένας πίνακας αντικειμένων της κλάσης `i_mach` N θέσεων ($N =$ γνωστό, $100 \leq N \leq 999$) και να αρχικοποιηθούν όλες οι θέσεις του πίνακα με χρήση `default constructor`, θέτοντας μηδενικές τιμές για όλα τα αριθμητικά δεδομένα.
2. Να γίνει εισαγωγή δεδομένων στον πίνακα αντικειμένων της `i_mach` μέσω δομητή (`constructor`) της κλάσης, χρησιμοποιώντας μόνον τυχαίους αριθμούς, στα όρια τιμών που έχουν δοθεί. Επειδή οι επιθεωρήσεις είναι συχνές κάθε μηχανή θα εμφανίζεται περισσότερες από μία φορές ανά έτος.
3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα:
 - a. Να βρείτε και να εμφανίσετε το πλήθος των επιθεωρήσεων για κάθε κωδικό μηχανής.
 - b. Να βρείτε και να εμφανίσετε, για κάθε έτος το πλήθος των επιθεωρήσεων για **μία** μηχανή καθώς και το πλήθος των 1 που έλαβε και στα 10 διαφορετικά σημεία επιθεώρησης στο έτος αυτό. Ο κωδικός της μηχανής εισάγεται μέσω της εντολής `cin` στη `main()`.
 - c. Να δημιουργήσετε και να εμφανίσετε έναν μονοδιάστατο δυναμικό πίνακα που θα περιλαμβάνει, σε μορφή ζευγών, τον κωδικό της μηχανής και το έτος επιθεώρησης κάθε φορά που το πλήθος των 0 είναι μεγαλύτερο του 5.

ΣΤΗ ΣΥΝΕΧΕΙΑ ΚΑΙ ΧΩΡΙΣ ΝΑ ΞΑΝΑΓΡΑΨΕΤΕ ΚΩΔΙΚΑ ΠΟΥ ΗΔΗ ΥΠΑΡΧΕΙ!!!

Να δημιουργήσετε μια απορρέουσα κλάση με όνομα `new_i_mach` που θα περιλαμβάνει ως **επιπλέον private μέλη – δεδομένα:**

- Το κόστος επιθεώρησης της κατάστασης καθενός από τα 10 σημεία της κάθε μηχανής (θετικοί αριθμοί τύπου `double`).

Στη συνέχεια να γράψετε :

τη συνάρτηση του `constructor` (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της κλάσης `new_i_mach` χρησιμοποιώντας μόνον τυχαίους αριθμούς **και θέτοντας μόνοι σας τα αντίστοιχα άνω όρια τιμών της συνάρτησης `rand()`.**

ΑΣΚΗΣΗ – 7

Για την παραγωγή ενός προϊόντος απαιτείται ένα προκαθορισμένο πλήθος ολοκληρωμένων φάσεων κατεργασίας (ή διεργασιών). Για τα προϊόντα που παράγονται στη μονάδα παραγωγής το μέγιστο πλήθος των διεργασιών είναι Z ($Z = \text{γνωστό}$), πλήθος που δεν ισχύει για όλα τα προϊόντα (δηλ. κάποια έχουν λιγότερες από Z διεργασίες). Μια διεργασία μπορεί να είναι κοινή σε περισσότερα από ένα προϊόντα. Επίσης μια διεργασία μπορεί να εμφανίζεται περισσότερες από μία φορές, για το ίδιο προϊόν.

Οι διεργασίες (φάσεις κατεργασίας) που εκτελούνται εφοδιάζονται με έναν αριθμό ταυτοποίησης (*id*, ακέραιος αριθμός από 1 – 999) και την κατάσταση (*status*) της διεργασίας με τιμές 0 (σημαίνει ότι η διεργασία δεν έχει ξεκινήσει να εκτελείται), 1 (σημαίνει ότι η διεργασία είναι σε εξέλιξη) ή 2 (η διεργασία έχει ολοκληρωθεί).

Να γραφεί μια κλάση με όνομα *product* και *private* μέλη - δεδομένα:

α) κωδικός προϊόντος (*code*, τύπου *int* με τιμές από 1-99)

β) το πλήθος των διεργασιών, έστω m (τύπου *int*) που απαιτούνται για να ολοκληρωθεί η παραγωγή του προϊόντος (είναι πάντοτε $0 < m \leq Z$)

γ) τα m σε πλήθος ζεύγη (αριθμός διεργασίας, κατάσταση διεργασίας) που αποτελούν στοιχεία ενός αριθμητικού πίνακα τύπου *int*.

Στη συνάρτηση *main()* :

- να οριστεί ένας πίνακας αντικειμένων της κλάσης *product* N θέσεων ($N = \text{γνωστό}$, $N \leq 99$).
- να αρχικοποιηθούν όλες οι θέσεις του πίνακα με χρήση *default constructor*, θέτοντας μηδενικές τιμές για όλα τα δεδομένα.
- Να γίνει εισαγωγή δεδομένων στον πίνακα αντικειμένων της κλάσης *product* μέσω ενός δομητή (*constructor*) της κλάσης, χρησιμοποιώντας μόνον τυχαίους αριθμούς, στα όρια τιμών που έχουν δοθεί.
- Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα:
 - να βρεθούν και να εμφανιστούν όλα τα προϊόντα η παραγωγή των οποίων έχει ολοκληρωθεί.
 - να βρεθούν και να εμφανιστούν όλα τα προϊόντα που έχουν το μέγιστο πλήθος φάσεων κατεργασίας σε εξέλιξη
 - να δημιουργηθεί και να εμφανιστεί ένας δυναμικός πίνακας που θα περιλαμβάνει τους κωδικούς των προϊόντων που έχουν κοινή μια συγκεκριμένη φάση κατεργασίας, ο κωδικός της οποίας θα δίνεται από το χρήστη κατά την εκτέλεση του προγράμματος.

ΣΤΗ ΣΥΝΕΧΕΙΑ ΚΑΙ ΧΩΡΙΣ ΝΑ ΞΑΝΑΓΡΑΨΕΤΕ ΚΩΔΙΚΑ ΠΟΥ ΗΔΗ ΥΠΑΡΧΕΙ!!!

Να δημιουργήσετε μια απορρέουσα κλάση με όνομα *extended_product* που θα περιλαμβάνει ως **επιπλέον private**

μέλη – δεδομένα:

- Το κόστος κάθε φάσης κατεργασίας του προϊόντος (θετικοί αριθμοί τύπου *double*)
- Τον προβλεπόμενο χρόνο εκτέλεσης κάθε φάσης κατεργασίας του προϊόντος (θετικοί αριθμοί τύπου *double*)

Στη συνέχεια να γράψετε τη συνάρτηση του *constructor* (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της κλάσης *extended_product* χρησιμοποιώντας μόνον τυχαίους αριθμούς **και θέτοντας μόνοι σας τα αντίστοιχα άνω όρια τιμών.**

ΑΣΚΗΣΗ – 8

Μία παραγωγική μονάδα χρησιμοποιεί 10 συνεργεία συντήρησης, με κωδικούς 1-10, για τη συντήρηση των εργαλειομηχανών της. Η μονάδα διαθέτει συνολικά 20 εργαλειομηχανές με κωδικούς 1-20. Το σειριακό αρχείο **service.txt** περιέχει πληροφορίες για τους χρόνους απασχόλησης κάθε συνεργείου για μία ημέρα του ιδίου έτους. Κάθε συνεργείο απασχολείται μόνον μία φορά την ημέρα σε κάποια εργαλειομηχανή. Κάθε γραμμή του αρχείου αφορά μόνον μία ημέρα και περιλαμβάνει:

- α) την ημερομηνία στη μορφή HHMM, αριθμός τύπου int
- β) το πλήθος των συνεργείων που απασχολήθηκαν, αριθμός τύπου int
- γ) υπό μορφή τριάδων: τον κωδικό του συνεργείου, τον κωδικό της εργαλειομηχανής και τον αντίστοιχο χρόνο απασχόλησης (σε πρώτα λεπτά, αριθμός τύπου int). (Προφανώς το μέγιστο πλήθος των ζευγών είναι 10.)

Το αρχείο δεν είναι ταξινομημένο ημερολογιακά και κάθε ημερομηνία υπάρχει μόνον μία φορά στο αρχείο. Π.χ. (οι παρενθέσεις δεν υπάρχουν στο αρχείο και παρατίθενται για διευκόλυνσή σας).

```
1107, 3, (5,7,100), (4,11,45), (8,13,90)
1405, 1, (7,10,80)
2903 2, (1,4,120), (9,17,250)
.....
```

Να γραφεί μια κλάση με όνομα **repair** που θα διαχειρίζεται τα δεδομένα από το αρχείο **service.txt**. Τα private μέλη – δεδομένα της κλάσης είναι τα δεδομένα του αρχείου.

Η κλάση πρέπει να περιέχει:

- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της
- Έναν constructor για την εκχώρηση τιμών από την ανάγνωση του αρχείου. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου).

Στη συνάρτηση `main()` :

1. να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **repair** με **N** θέσεις (**N** = γνωστό) και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα. Η τιμή του **N** μπορεί να είναι μικρότερη, μεγαλύτερη ή και ίση με το πλήθος των εγγραφών του αρχείου **service.txt**. Στη συνέχεια να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor.
2. Να βρεθεί (αν υπάρχει) το πλήθος των γραμμών από το αρχείο που δεν περιλαμβάνονται στον πίνακα αντικειμένων.
3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα:
 - a. Να βρείτε και να εμφανίσετε τον συνολικό χρόνο που αφιερώθηκε για τη συντήρηση κάθε μιας από τις εργαλειομηχανές, δηλ. 20 τιμές

- b. Να βρείτε και να εμφανίσετε τον συνολικό χρόνο απασχόλησης κάθε συνεργείου για κάθε μήνα του έτους για τη συντήρηση όλων των εργαλειομηχανών, δηλ. συνολικά 120 τιμές
4. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_service**, που θα περιλαμβάνει ως επιπλέον private μέλος – δεδομένο το αντίστοιχο κόστος (θετικός ακέραιος αριθμός) συντήρησης για κάθε συνεργείο και κάθε εργαλειομηχανή (δηλ. θα υπάρχουν για κάθε ημέρα τόσα κόστη όσα και τα συνεργεία που απασχολήθηκαν). Στη συνέχεια:
- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_service**.
- b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_service**, μεγέθους $N = \text{γνωστό}$.
- c. Χρησιμοποιώντας τον δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **repair** που έχετε δημιουργήσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα ποσά για το νέο μέλος-δεδομένο σε κάθε θέση του νέου πίνακα.
- (ΥΠΟΔΕΙΞΗ : όσες συναρτήσεις – μέλη των κλάσεων **service** και **new_service** απαιτηθούν κατά την ανάπτυξη του προγράμματος μπορούν να γραφούν σε οποιοδήποτε σημείο με χρήση των κατάλληλων συμβολισμών)

ΑΣΚΗΣΗ – 9

Ένα αιολικό πάρκο αποτελείται από 20 πανομοιότυπες ανεμογεννήτριες με κωδικούς 1-20. Κάθε ανεμογεννήτρια περιλαμβάνει 10 κύρια υποσυστήματα (π.χ. ανεμόμετρο, πτερύγιο, κέλυφος κλπ) με κωδικούς 1-10. Το σειριακό αρχείο **malfunction.txt** περιέχει πληροφορίες για τους χρόνους κατά τους οποίους μια ανεμογεννήτρια τίθεται εκτός λειτουργίας λόγω βλάβης σε κάποιο από τα κύρια υποσυστήματά της για τις ημέρες του ίδιου έτους. Να θεωρήσετε ότι κάθε τέτοια βλάβη οφείλεται κάθε φορά, κατά κύριο λόγο, σε ένα μόνον από αυτά τα υποσυστήματα. Μπορούν να υπάρχουν περισσότερες από μία βλάβες κάθε ημέρα από το ίδιο υποσύστημα, για την ίδια ή για διαφορετικές ανεμογεννήτριες. Κάθε γραμμή του αρχείου αφορά μόνον μία ημέρα και περιλαμβάνει:

α) την ημερομηνία στη μορφή HHMM, αριθμός τύπου int (HH = ημέρα, MM = μήνας)

β) το πλήθος των βλαβών που εντοπίστηκαν σε όλες τις ανεμογεννήτριες (τύπου int), με μέγιστο πλήθος 20.

γ) υπό μορφή αριθμητικών τριάδων, τύπου int: τον κωδικό του υποσυστήματος, τον κωδικό της ανεμογεννήτριας και τον αντίστοιχο χρόνο κατά τον οποίο η ανεμογεννήτρια τέθηκε εκτός λειτουργίας (σε πρώτα λεπτά, αριθμός τύπου int). (Προφανώς το μέγιστο πλήθος των τριάδων είναι 20.)

Το αρχείο δεν είναι ταξινομημένο ημερολογιακά και κάθε ημερομηνία υπάρχει μόνον μία φορά στο αρχείο. Π.χ. (οι παρενθέσεις δεν υπάρχουν στο αρχείο και παρατίθενται για διευκόλυνσή σας).

```
1405, 6, (9, 7, 50), (2, 11, 40), (7, 13, 60), (2, 7, 30), (2, 11, 30), (7, 10, 60)
2706, 3, (3, 14, 15), (9, 18, 110), (2, 8, 17)
1102, 4, (1, 10, 45), (6, 15, 20), (6, 11, 77) (5, 13, 34)
.....
```

Να γραφεί μια κλάση με όνομα **damage** που θα διαχειρίζεται τα δεδομένα από το αρχείο **malfunction.txt**. Τα private μέλη – δεδομένα της κλάσης είναι :

1. Η ημερομηνία στη μορφή HHMM, αριθμός τύπου int
2. Το πλήθος των διαφορετικών υποσυστημάτων που προκάλεσαν βλάβη (μέγιστο πλήθος = 10)
3. Υπό μορφή αριθμητικών τριάδων τύπου int, ταξινομημένων κατά αύξοντα αριθμό υποσυστήματος:
 - τον κωδικό του υποσυστήματος,
 - το πλήθος των διαφορετικών ανεμογεννητριών στις οποίες προέκυψε βλάβη και
 - τον αντίστοιχο ημερήσιο συνολικό χρόνο εκτός λειτουργίας (σε πρώτα λεπτά, αριθμός τύπου int).

ΠΑΡΑΔΕΙΓΜΑ : Το 1^ο αντικείμενο της κλάσης **damage** για την 1^η γραμμή του αρχείου θα έχει ως δεδομένα:

```
1405 3 2,2,100 7,2,120 9,1,50
```

Η κλάση πρέπει να περιέχει:

- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της
- Έναν constructor για την εκχώρηση τιμών από την ανάγνωση του αρχείου. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου).

Στη συνάρτηση `main()` να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **damage** με N θέσεις ($N =$ γνωστό) και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα. Η τιμή του N μπορεί να είναι μικρότερη, μεγαλύτερη ή και ίση με το πλήθος των εγγραφών του αρχείου **malfunction.txt**.

Στη συνέχεια μέσω της προσπέλασης του αρχείου **malfunction.txt** :

1. Να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor.
2. Να γίνει εισαγωγή σε έναν δυναμικό πίνακα, από κάθε εγγραφή του αρχείου, ζευγών που περιλαμβάνουν τον αύξοντα αριθμό της ανεμογεννήτριας και το χρόνο εκτός λειτουργίας (π.χ. για την 1^η εγγραφή του αρχείου ο δυναμικός πίνακας θα περιλαμβάνει τα ζεύγη : (7, 50), (11, 40), (13, 60), (7,30), (11, 30), (10,60)).
3. Να βρεθεί (αν υπάρχει) το πλήθος των γραμμών από το αρχείο που δεν περιλαμβάνονται στον πίνακα αντικειμένων.
4. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα αντικειμένων της κλάσης **damage**:
Να βρείτε και να εμφανίσετε τον συνολικό χρόνο που οι ανεμογεννήτριες τέθηκαν εκτός λειτουργίας λόγω βλάβης καθενός υποσυστήματος, για κάθε έναν από τους 12 μήνες του έτους, δηλ. συνολικά $10 \times 12 = 120$ τιμές.
5. Χρησιμοποιώντας όλα τα στοιχεία του δυναμικού πίνακα να βρείτε και να εμφανίσετε το συνολικό χρόνο εκτός λειτουργίας κάθε εργαλειομηχανής (MONΑΔΕΣ 1,0)
6. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_damage**, που θα περιλαμβάνει ως επιπλέον private μέλη – δεδομένα τα αντίστοιχα κόστη αποκατάστασης (τύπου int) λόγω των βλαβών των υποσυστημάτων για κάθε ημερομηνία (π.χ. για το 1^ο αντικείμενο της κλάσης **damage** με δεδομένα:

1405 3 2, 2, 100 7, 2, 120 9, 1, 50

θα πρέπει να υπάρχουν ως νέα δεδομένα τρεις ακέραιοι αριθμοί που θα αντιστοιχούν στα κόστη για τα υποσυστήματα 2, 7 και 9)

Στη συνέχεια:

- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_damage**.
- b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_damage**, μεγέθους $N =$ γνωστό.
- c. Χρησιμοποιώντας τον δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **damage** που έχετε δημιουργήσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα ποσά για τα νέα μέλη - δεδομένα σε κάθε θέση του νέου πίνακα.

ΑΣΚΗΣΗ – 10

Μία παραγωγική μονάδα παράγει 10 διαφορετικά εξαρτήματα μηχανών, με κωδικούς 1-10. Το σειριακό αρχείο **data.txt** περιέχει πληροφορίες για τα εξαρτήματα αυτά. Κάθε γραμμή του αρχείου αφορά μια ημερομηνία παραγωγής μέσα στο ίδιο έτος και περιλαμβάνει :

- ημερομηνία παραγωγής στη μορφή HHMM (ημέρα και μήνας) με τιμές ημέρας 1-30 και μήνα 1-12.
- ημερήσιες ποσότητες παραγωγής για κάθε ένα από τα 10 εξαρτήματα (int). Για ορισμένα από τα εξαρτήματα η παραγωγή μπορεί να είναι μηδενική.

Να θεωρήσετε ότι κάθε ημερομηνία υπάρχει μόνον μία φορά στο αρχείο. Το αρχείο δεν είναι ταξινομημένο ημερολογιακά. Π.χ.

```
1104    20   0  14  15  67  0  44  56  0  28
1705    21  11   0  16   0  9  33  0  9  14
2901     7  11   0   0   3  0   4  15  20  0
.....
```

Να γραφεί μια κλάση με όνομα **parts** που θα διαχειρίζεται τα δεδομένα από το αρχείο **data.txt**. Τα private μέλη – δεδομένα της κλάσης είναι :

- ημερομηνία παραγωγής στη μορφή HHMM , όπως παραπάνω
- ημερήσιες ποσότητες παραγωγής για κάθε ένα από τα 10 εξαρτήματα (int).

Η κλάση πρέπει να περιέχει:

- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της
- Έναν constructor για την εκχώρηση τιμών από την ανάγνωση του αρχείου. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου).

Στη συνάρτηση main() :

1. να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **parts** με **N** θέσεις (**N** = γνωστό) και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα. Η τιμή του **N** μπορεί να είναι μικρότερη, μεγαλύτερη ή και ίση με το πλήθος των εγγραφών του αρχείου **_data.txt**. Στη συνέχεια να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor.
2. Να βρεθεί (αν υπάρχει) το πλήθος των γραμμών από το αρχείο που δεν περιλαμβάνονται στον πίνακα αντικειμένων.
3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα:
 - a. Να βρείτε και να εμφανίσετε τη συνολική ποσότητα παραγωγής κάθε μήνα του έτους της παραγωγικής μονάδας για κάθε εξάρτημα (δηλ. συνολικά $12 \times 10 = 120$ τιμές).
 - b. Να δημιουργήσετε και να εμφανίσετε έναν μονοδιάστατο δυναμικό πίνακα που θα περιλαμβάνει, σε μορφή ζευγών, για κάθε κωδικό εξαρτήματος το πλήθος των ημερών του έτους με μηδενική παραγωγή.

4. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_parts**, που θα περιλαμβάνει ως επιπλέον private μέλος – δεδομένο το κόστος παραγωγής για κάθε ποσότητα παραγωγής και για κάθε εξάρτημα ανά ημέρα. Εφόσον η ημερήσια παραγωγή του εξαρτήματος είναι μηδενική θα πρέπει και το αντίστοιχο κόστος παραγωγής να είναι μηδενικό. Στη συνέχεια:
- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_parts**.
 - b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_parts**, μεγέθους N = γνωστό.
 - c. Χρησιμοποιώντας τον δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **parts** που έχετε δημιουργήσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα ποσά για το νέο μέλος-δεδομένο σε κάθε θέση του νέου πίνακα.

ΑΣΚΗΣΗ – 11

Μία μονάδα παραγωγής κατασκευάζει ένα προϊόν. Η ημερήσια ποσότητα παραγωγής του προϊόντος σε τεμάχια, η ημερομηνία σε μορφή HHMM (HH = ημέρα, MM = μήνας), καθώς και η ημερήσια ζήτηση για το προϊόν αυτό από τα 10 διαφορετικά σημεία διανομής που διαθέτει η παραγωγική μονάδα υπάρχουν στο σειριακό αρχείο **orders.txt**. Για κάθε διαφορετική ημερομηνία υπάρχει μόνον 1 εγγραφή στο αρχείο, δηλ. κάθε γραμμή του αρχείου αφορά μόνον μία ημέρα, και το αρχείο δεν είναι ταξινομημένο ημερολογιακά. Παράδειγμα: (ημ/νία 26/03, ποσότητα παραγωγής 2000, ημερήσια ζήτηση οι υπόλοιπες 10 τιμές)

```
2603  2000  [ 50 | 100 | 240 | 80 | 350 | 90 | 100 | 70 | 200 | 320 ]
.....
```

Να γραφεί μια κλάση με όνομα **alfa** που θα διαχειρίζεται τα δεδομένα από το αρχείο **orders.txt**. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου). Τα private μέλη – δεδομένα της κλάσης είναι :

1. Ο κωδικός μήνα, τιμές int 1-12
2. Ο κωδικός του σημείου διανομής, τιμές int 1-10
3. Η συνολική μηνιαία ζήτηση για το προϊόν από αυτό το σημείο διανομής

Η κλάση πρέπει να περιέχει:

- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της
- Έναν constructor για την εκχώρηση τιμών στα αντικείμενα της κλάσης μέσω της ανάγνωσης (προσπέλασης) του αρχείου.

Στη συνάρτηση main() να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **alfa** μεγέθους $N = 120$ θέσεων και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα.

Στη συνέχεια μέσω της προσπέλασης του αρχείου **orders.txt** :

1. Να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor. Οι τιμές θα εισαχθούν κατά αύξουσα σειρά του σημείου διανομής και κατά αύξουσα σειρά του αριθμού μήνα, δηλ. πρώτα όλα τα μηνιαία σύνολα για το σημείο διανομής 1, έπειτα όλα τα μηνιαία σύνολα για το σημείο διανομής 2 κ.ο.κ. (ΥΠΟΔΕΙΞΗ : να χρησιμοποιήσετε έναν ενδιάμεσο πίνακα τύπου int, μεγέθους 10 X 12 θέσεων).
2. Να γίνει εισαγωγή σε έναν δυναμικό πίνακα, από κάθε εγγραφή του αρχείου, ζευγών τιμών που περιλαμβάνουν τον αριθμό του μήνα και την υπολειπόμενη ή την πλεονάζουσα ημερήσια ποσότητα παραγωγής (δηλ. τη διαφορά της συνολικής ημερήσιας ζήτησης από την ημερήσια ποσότητα παραγωγής (π.χ. για την 1^η εγγραφή του αρχείου ο δυναμικός πίνακας θα περιλαμβάνει το ζεύγος : (3, 400) $\left(400 = 2000 - \sum_{i=1}^{10} x_i = 2000 - 1600 \right)$).
3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα αντικειμένων της κλάσης **alfa** να βρείτε και να εμφανίσετε τη συνολική ετήσια ζήτηση του προϊόντος από το κάθε ένα από τα 10 σημεία διανομής (ΜΟΝΑΔΕΣ 1,5)

4. Χρησιμοποιώντας όλα τα στοιχεία του δυναμικού πίνακα να βρείτε και να εμφανίσετε, με βάση τη ζήτηση, πόσες ημέρες καθενός από τους 12 μήνες η παραγωγή είναι ελλειμματική και πόσες ημέρες είναι πλεονάζουσα (ΜΟΝΑΔΕΣ 2,0)
5. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_alfa**, που θα περιλαμβάνει ως επιπλέον private μέλη – δεδομένα, το κόστος παραγωγής και το κόστος αποστολής της μηνιαίας ποσότητας παραγωγής του προϊόντος για το σημείο διανομής, δύο τιμές τύπου double.

Στη συνέχεια:

- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_alfa**.
- b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_alfa**, μεγέθους N .
- c. Χρησιμοποιώντας το δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **alfa** που έχετε γεμίσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα ποσά για τα νέα μέλη - δεδομένα σε κάθε θέση του νέου πίνακα.

ΑΣΚΗΣΗ – 12

A. Σε πολλές εφαρμογές, η δημιουργία κωδικών ασφαλείας χρησιμοποιεί πρώτους (primes) αριθμούς. Ένας αριθμός είναι πρώτος (prime) όταν οι μοναδικοί του διαιρέτες είναι το 1 και ο ίδιος ο αριθμός. Στο τέλος δίνεται ένα πλήρες πρόγραμμα σε C++ που βρίσκει εάν ένας αριθμός είναι prime ή όχι. Ο πρώτος prime είναι ο 2.

B. Ένας ακέραιος θετικός αριθμός μπορεί να γραφεί ως άθροισμα δύο διαφορετικών πρώτων αριθμών. Παράδειγμα για τον αριθμό 16:

$16 = 2 + 14$	Το 2 είναι πρώτος, το 14 όχι
$16 = 3 + 13$	Και οι δύο είναι πρώτοι
$16 = 4 + 12$	Κανένας δεν είναι πρώτος
$16 = 5 + 11$	Και οι δύο είναι πρώτοι
$16 = 6 + 10$	Κανένας δεν είναι πρώτος
$16 = 7 + 9$	Το 7 είναι πρώτος, το 9 όχι
$16 = 8 + 8$	Οι αριθμοί είναι ΙΔΙΟΙ !!!!

Γ. Δίνεται ένα σειριακό αρχείο με όνομα **primes.txt** που περιέχει πρώτους (primes) αριθμούς, όλους σε μία γραμμή και σε αύξουσα διάταξη. Οι αριθμοί αυτοί δεν είναι όλοι οι primes που υπάρχουν, όλοι είναι μικρότεροι του 1000. Παράδειγμα για τα περιεχόμενα του αρχείου: 19, 71, 89, 107, 113, 127, 277, 331, 359, 397, 431,

Το πλήθος των αριθμών αυτών δεν είναι μεγαλύτερο από μία γνωστή τιμή N, η οποία δίνεται ως σταθερά στο πρόγραμμα. Στη συνάρτηση main(), μέσω προσπέλασης του σειριακού αρχείου να εισάγετε αυτούς τους αριθμούς σε έναν μονοδιάστατο πίνακα ακεραίων, N θέσεων, που θα έχει όνομα p.

Να ορίσετε μία κλάση με όνομα **test_prime** με μέλη – δεδομένα:

- Έναν θετικό ακέραιο αριθμό στα όρια [2 - 5000]. Ο αριθμός δεν πρέπει να είναι πρώτος (prime).
- Διαδοχικά ζεύγη πρώτων αριθμών το άθροισμα των οποίων είναι ίσο με τον θετικό ακέραιο αριθμό (ΥΠΟΔΕΙΞΗ : Να θεωρήσετε ότι το μέγιστο πλήθος αυτών των ζευγών δεν θα είναι μεγαλύτερο από 10)
- Πλήθος των ζευγών αυτών

Η κλάση πρέπει να περιέχει:

- μία συνάρτηση – μέλος, με όνομα find_prime, που θα δέχεται ως όρισμα εισόδου έναν θετικό ακέραιο αριθμό, μεγαλύτερο του 1, και θα επιστρέφει εάν ο αριθμός είναι πρώτος (prime) ή όχι. Η συνάρτηση θα προκύψει αξιοποιώντας τον κώδικα που δίνεται στο τέλος.
- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της.
- Έναν constructor για την ανάθεση τιμών στα αντικείμενα της κλάσης. Οι τιμές θα μεταβιβάζονται μέσω κατάλληλης κλήσης του constructor στη συνάρτηση main(). Συγκεκριμένα :
 - ο θετικός ακέραιος θα δημιουργείται με κατάλληλη χρήση της συνάρτησης rand().
 - Τα διαδοχικά ζεύγη θα δημιουργούνται χρησιμοποιώντας :
 - ΜΟΝΟΝ τους πρώτους αριθμούς που έχουν διαβαστεί από το αρχείο **primes.txt** και βρίσκονται στον πίνακα p, καθώς και
 - την αλγοριθμική διαδικασία που περιγράφεται στη συνέχεια.

ΑΛΓΟΡΙΘΜΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Για κάθε αντικείμενο του πίνακα αντικειμένων:

- Χρησιμοποίησε τον θετικό ακέραιο αριθμό (1^ο μέλος της κλάσης στην οποία ανήκει το αντικείμενο)
 - Για κάθε πρώτο (prime) αριθμό που υπάρχει στον πίνακα p
 - Έλεγχος αν το άθροισμα αυτού και του επόμενου του πρώτου αριθμού στον πίνακα p ισούται με τον θετικό ακέραιο αριθμό. Εάν αυτό ισχύει δημιουργήσε το σχετικό ζεύγος και να το εισάγεις στην αλληλουχία των διαδοχικών ζευγών, αυξάνοντας και το πλήθος αυτών των ζευγών κατά 1.

Στη συνάρτηση `main()` να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **test_prime** μεγέθους M = γνωστό, και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα.

Στη συνέχεια, χρησιμοποιώντας όλα τα στοιχεία του πίνακα αντικειμένων:

1. Να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor.
2. Να βρείτε και να εμφανίσετε σε ποια αντικείμενα του πίνακα αντικειμένων ο θετικός ακέραιος είναι prime και το πλήθος των ζευγών primes είναι μηδενικό.
3. Να βρείτε και να εμφανίσετε τη συχνότητα κάθε αριθμού του πίνακα p σε όλα τα αντικείμενα του πίνακα αντικειμένων.
4. Να γίνει εισαγωγή σε έναν δυναμικό πίνακα, τύπου `int`, από κάθε αντικείμενο του πίνακα αντικειμένων, και για όσα ζεύγη πρώτων αριθμών υπάρχουν στο κάθε αντικείμενο του πίνακα αντικειμένων, τριάδων ακέραιων θετικών τιμών. Κάθε τριάδα θα περιλαμβάνει:
 - i. Τον θετικό ακέραιο αριθμό
 - ii. Τον πρώτο prime αριθμό του ζεύγους
 - iii. Τον μέσο όρο (ακέραια τιμή) του ζεύγους των πρώτων (primes) αριθμών

Στο τέλος να εμφανίσετε όλες τις τριάδες του δυναμικού πίνακα, κάθε τριάδα σε μία γραμμή.

5. Να δημιουργήσετε μία απορρέουσα κλάση με όνομα **new_test_prime**, που θα περιλαμβάνει ως επιπλέον private μέλος – δεδομένο (τύπου `int`), τον μεγαλύτερο prime αριθμό που είναι μικρότερος του θετικού ακέραιου αριθμού (δηλ του 1^{ου} μέλους της κλάσης). Η εύρεση αυτού του αριθμού prime πρέπει να γίνει μέσα στο δομητή (constructor) της απορρέουσας κλάσης

Στη συνέχεια:

- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_test_prime**.
- b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_test_prime**, μεγέθους M .
- c. Χρησιμοποιώντας το δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **test_prime** που έχετε γεμίσει στο ερώτημα 1.

ΑΣΚΗΣΗ – 13

Μία βιομηχανική μονάδα διατηρεί το σειριακό αρχείο **malfun.txt** για τις βλάβες των εργαλειομηχανών της. Η σημαντικότητα κάθε βλάβης προσδιορίζεται σε δεκαβάθμια κλίμακα, δηλ. κάθε βλάβη κατηγοριοποιείται με μία τιμή 1-10. Κάθε γραμμή του αρχείου, το οποίο δεν είναι ταξινομημένο ημερολογιακά, περιέχει τα εξής στοιχεία:

- Ημερομηνία βλάβης στη μορφή MMEE (MM = μήνας (1-12), EE = έτος (10 - 19 → 2010 – 2019)), τύπου int
- Πλήθος εργαλειομηχανών στις οποίες παρουσιάστηκε βλάβη το συγκεκριμένο μήνα (δυναμικές τιμές 0-50), τύπου int
- Τριάδες τιμών τύπου int, ίσες σε πλήθος με το αμέσως προηγούμενο δεδομένο του αρχείου. Κάθε τριάδα περιλαμβάνει τα εξής στοιχεία για κάθε μία βλάβη που εντοπίστηκε σε μια εργαλειομηχανή:
 1. Κωδικός εργαλειομηχανής, τύπου int, περιοχή τιμών 1-15, δηλ. υπάρχουν 15 διαφορετικές εργαλειομηχανές
 2. Χρόνος αποκατάστασης της βλάβης σε ώρες, τύπου double, τιμές θετικές ή μηδενικές
 3. Σημαντικότητα της βλάβης, αριθμός τύπου int, δυναμικές τιμές για μία βλάβη 1 - 10

Για κάθε διαφορετική ημερομηνία υπάρχει μόνον 1 εγγραφή στο αρχείο, δηλ. κάθε γραμμή του αρχείου αφορά μόνον ένα μήνα ενός έτους. Μία εργαλειομηχανή είναι πιθανό να παρουσιάσει βλάβη περισσότερες από μία φορές μέσα στον ίδιο μήνα του ίδιου έτους. Παράδειγμα (ημ/νία 514 (5^{ος} του 2014), πλήθος 5, και ακολουθούν οι τριάδες τιμών)

514 5

7	3.2	7	14	1.8	2	11	7.6	8	3	0.4	1	3	2.7	6
---	-----	---	----	-----	---	----	-----	---	---	-----	---	---	-----	---

.....

Να γραφεί μια κλάση με όνομα **malff** που θα διαχειρίζεται και θα ομαδοποιεί τα δεδομένα από το αρχείο **malfun.txt**. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου). Τα private μέλη – δεδομένα της κλάσης είναι :

1. κωδικός εργαλειομηχανής, τιμές int 1-15
2. ετήσιος συνολικός χρόνος αποκατάστασης βλαβών για κάθε ένα από τα δέκα έτη 2010 - 2019, δηλ. 10 τιμές τύπου double, που αφορούν τον κωδικό εργαλειομηχανής
3. πλήθος βλαβών ανά κατηγορία σημαντικότητας (συνολικά για όλα τα έτη) που αφορούν τον κωδικό εργαλειομηχανής

Η κλάση πρέπει να περιέχει:

- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της
- Έναν constructor για την εκχώρηση τιμών στα αντικείμενα της κλάσης μέσω της ανάγνωσης (προσπέλασης) του αρχείου.

Στη συνάρτηση main() να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **malff** μεγέθους $N = 15$ θέσεων, όσες και οι εργαλειομηχανές, και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα.

Στη συνέχεια μέσω της προσπέλασης του αρχείου **malfun.txt**:

1. Να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor. Οι τιμές θα εισαχθούν κατά αύξουσα σειρά του κωδικού εργαλειομηχανής (ΜΟΝΑΔΕΣ 3,5). (ΥΠΟΔΕΙΞΗ : να χρησιμοποιήσετε κατάλληλες δισδιάστατες δομές δεδομένων).
2. Παράλληλα με το ερώτημα 1 : Να γίνει εισαγωγή σε έναν δυναμικό πίνακα τύπου int, ζευγών τιμών. Κάθε ζεύγος θα περιλαμβάνει :
 - a. Την ημερομηνία,
 - b. Το πλήθος των βλαβών για μία κατηγορία σημαντικότητας, η τιμή της οποίας θα εισάγεται (ή θα δημιουργείται) κατά την εκτέλεση του προγράμματος

Με την ολοκλήρωση της προσπέλασης του αρχείου **malfun.txt** το πρόγραμμα θα πρέπει να εμφανίζει τον δυναμικό πίνακα καθώς και το ετήσιο πλήθος της συγκεκριμένης κατηγορίας βλαβών για καθένα από τα έτη 2010-2019, κατά αύξουσα σειρά του έτους

3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα αντικειμένων της κλάσης **malf**, να βρείτε και να εμφανίσετε
 - a. το συνολικό χρόνο αποκατάστασης βλαβών για καθένα από τα 10 έτη, δηλ. το συνολικό κόστος αποκατάστασης για τα έτη 2010, 2011, ..., 2019
 - b. το συνολικό πλήθος βλαβών για κάθε μία κατηγορία σημαντικότητας, δηλ. για τις κατηγορίες σημαντικότητας 1, 2, 3, ..., 10
4. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_malf**, που θα περιλαμβάνει ως επιπλέον private μέλη – δεδομένα, τύπου double, τα αντίστοιχα ετήσια κόστη αποκατάστασης των βλαβών της εργαλειομηχανής για καθένα από τα έτη 2010 - 2019.

Στη συνέχεια:

- d. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_malf**.
- e. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_malf**, μεγέθους N .
- f. Χρησιμοποιώντας το δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **malf** που έχετε δημιουργήσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα αριθμητικά δεδομένα για τα νέα μέλη - δεδομένα σε κάθε θέση του νέου πίνακα.

ΑΣΚΗΣΗ – 14

Μία μονάδα παραγωγής κατασκευάζει N διαφορετικά προϊόντα. Οι κωδικοί των προϊόντων είναι ακέραιοι αριθμοί $[1, 999]$. Ωστόσο ορισμένα προϊόντα είναι δυνατόν να μην κατασκευαστούν κατά τη διάρκεια ενός ημερολογιακού έτους (για έναν ή περισσότερους μήνες ή και για όλο το έτος).

Για το έτος 2018 υπάρχουν δύο σειριακά αρχεία με ονόματα, αντίστοιχα, **alpha.txt** και **beta.txt** που αφορούν την παραγωγή των N διαφορετικών προϊόντων για όλη τη διάρκεια του έτους.

Το αρχείο **alpha.txt** περιέχει σε κάθε γραμμή του, με τη σειρά που παρατίθενται στη συνέχεια, τρεις τιμές:

- **Κωδικός προϊόντος, αριθμός μήνα (1-12), παραχθείσα ποσότητα (θετική τιμή τύπου longint)**

Ένα προϊόν μπορεί να παραχθεί περισσότερες από μία φορές μέσα στον ίδιο μήνα.

Το αρχείο **alpha.txt** ΔΕΝ ΕΙΝΑΙ ταξινομημένο ως προς κάποια από τις τιμές κάθε γραμμής.

Παράδειγμα ορισμένων γραμμών από το αρχείο **alpha.txt**:

```
211, 4, 1770
33, 11, 500
727, 2, 100
.....
```

Το αρχείο **beta.txt** περιέχει μία γραμμή για κάθε κωδικό που υπάρχει σε μία τουλάχιστον γραμμή του αρχείου **alpha.txt**. Κάθε γραμμή του **beta.txt** περιέχει δύο τιμές, τον κωδικό του προϊόντος και τη συνολική ετήσια παραγωγή. Το αρχείο ΔΕΝ ΕΙΝΑΙ ΤΑΞΙΝΟΜΗΜΕΝΟ ως προς τον κωδικό προϊόντος (κάθε κωδικός προϊόντος εμφανίζεται μόνον μία φορά στο αρχείο **beta.txt**). Παράδειγμα ορισμένων γραμμών από το αρχείο **beta.txt**:

```
400, 25490
33, 18700
.....
```

Η μονάδα παραγωγής επιθυμεί να επαληθεύσει εάν η συνολική ετήσια παραγωγή ανά κωδικό προϊόντος του αρχείου **beta.txt** είναι ίση με το άθροισμα των μηνιαίων ποσοτήτων παραγωγής, ανά κωδικό προϊόντος, από το αρχείο **alpha.txt**. Ο CIO (Chief Information Officer) της μονάδας σας ζητά τα εξής, σε γλώσσα C++ και με τη χρήση κλάσεων και αντικειμένων:

1. Να ορίσετε μία κλάση με όνομα **check** και μέλη – δεδομένα :
 - a. Κωδικός προϊόντος $[1,999]$
 - b. Πλήθος μηνών που υπάρχει παραγωγή στο αρχείο **alpha.txt**
 - c. Αλληλουχία των μηνιαίων τιμών, σε αύξουσα σειρά του αριθμού μήνα, σε μορφή ζευγών (α/α μήνα, μηνιαία παραχθείσα ποσότητα). Το πλήθος αυτών των ζευγών προφανώς θα είναι για κάθε προϊόν ίσο με το μέλος (b).
 - d. Συνολική ετήσια παραγωγή που προκύπτει από το αρχείο **alpha.txt**
 - e. Συνολική ετήσια παραγωγή όπως αναφέρεται στο αρχείο **beta.txt**
2. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **check**, N θέσεων, με όνομα πίνακα **a**.
3. Να γίνει εισαγωγή δεδομένων στον πίνακα **a**, μέσω προσπέλασης από τα δύο αρχεία (απαιτείται έλεγχος για την ύπαρξη των αρχείων). Για την εισαγωγή των δεδομένων από τα αρχεία απαιτούνται :

- a. Default constructor της κλάσης **check**, για την απόδοση, αρχικά, μηδενικών τιμών στα μέλη των αντικειμένων της κλάσης
- b. Constructor της κλάσης **check**, με λίστα παραμέτρων, για την εκχώρηση τιμών στα αντικείμενα της κλάσης μέσω της ανάγνωσης (προσπέλασης) των δύο αρχείων.

Οι τιμές στον πίνακα a θα εισαχθούν κατά αύξουσα σειρά του κωδικού προϊόντος.

ΥΠΟΔΕΙΞΕΙΣ :

1. Να χρησιμοποιήσετε κατάλληλη (κατάλληλες) δισδιάστατη (δισδιάστατες) δομή (δομές) δεδομένων
2. Εάν χρησιμοποιήσετε μία κατάλληλα οργανωμένη δομή δεδομένων για να διακρατήσετε τα δεδομένα του αρχείου **beta.txt** θα απλοποιήσετε σε μεγάλο βαθμό τη διαδικασία εισαγωγής δεδομένων στον πίνακα **a**.
3. Η κλάση ΔΕΝ ΘΑ ΠΕΡΙΕΧΕΙ ΣΥΝΑΡΤΗΣΕΙΣ *print()*.
4. Να δημιουργηθεί δυναμικός πίνακας ζευγών ακεραίων αριθμών. Κάθε ζεύγος θα περιλαμβάνει: τον κωδικό του προϊόντος και τον αριθμό του μήνα, για κάθε μήνα του έτους για τον οποίο δεν υπάρχει παραγωγή αυτού του προϊόντος. Τα στοιχεία του δυναμικού πίνακα θα εισάγονται κατά αύξουσα σειρά του κωδικού προϊόντος και κατά αύξουσα σειρά του αριθμού μήνα. Όταν ολοκληρωθεί η συμπλήρωση του δυναμικού πίνακα να εμφανίσετε για κάθε προϊόν που περιλαμβάνεται στο δυναμικό πίνακα το πλήθος των μηνών για τους οποίους δεν υπάρχει παραγωγή (ΜΟΝΑΔΕΣ 2,5).
5. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα **a**, να βρείτε και να εμφανίσετε:
 - a. Σε ποιους κωδικούς προϊόντων υπάρχει διαφορά των δύο συνόλων παραγωγής (διαφορά των μελών (d) και (e) κάθε αντικειμένου, δηλ. (d) – (e)). Να εμφανίσετε για κάθε τέτοιο κωδικό προϊόντος, σε μία γραμμή της οθόνης για κάθε προϊόν, τον κωδικό του προϊόντος μαζί με τη διαφορά αυτή.
 - b. Τη συνολική, ανά μήνα, παραγωγή της παραγωγικής μονάδας συμπεριλαμβάνοντας όλους τους κωδικούς προϊόντων.
6. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_check**, που θα περιλαμβάνει ως επιπλέον private μέλη – δεδομένα, για κάθε μήνα για τον οποίο υπάρχει παραγωγή του προϊόντος:
 - Τη (μηνιαία) ποσότητα παραγωγής που πέρασε επιτυχώς τον ποιοτικό έλεγχο (θα είναι το πολύ ίση με τη μηνιαία παραχθείσα ποσότητα)

Στη συνέχεια :

- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_check**.
- b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_check**, μεγέθους *N* .
- c. Χρησιμοποιώντας το δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **check** που έχετε δημιουργήσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα αριθμητικά δεδομένα για τα νέα μέλη - δεδομένα σε κάθε θέση του νέου πίνακα.

ΑΣΚΗΣΗ – 15

Μία βιομηχανική μονάδα παράγει 10 διαφορετικά προϊόντα, με κωδικούς 1-10. Το σειριακό αρχείο **prod2018.txt** περιέχει σε κάθε γραμμή του τα εξής στοιχεία για την ημερήσια παραγωγή του έτους 2018 και τις παραγγελίες των προϊόντων, χωρίς να είναι ταξινομημένο ημερολογιακά:

- Ημερομηνία στη μορφή HHMM (HH = ημέρα, MM = μήνας), τύπου int
- Παραχθείσα ημερήσια ποσότητα για καθένα από τα 10 διαφορετικά προϊόντα (10 τιμές), τύπου longint
- Πλήθος ημερήσιων παραγγελιών, τύπου int
- Ζεύγη τιμών τύπου int, ίσες σε πλήθος με το προηγούμενο δεδομένο του αρχείου. Κάθε ζεύγος περιλαμβάνει τα εξής στοιχεία για την παραγγελία ενός μόνον προϊόντος:
 - Κωδικός προϊόντος
 - Ποσότητα παραγγελίας

Για κάθε προϊόν μπορεί να μην υπάρχει παραγωγή ή παραγγελία κάποια ημέρα, ενώ η παραχθείσα ημερήσια ποσότητα μπορεί να μην αρκεί για να καλύψει την ημερήσια ζήτηση των παραγγελιών. Παράδειγμα μιας γραμμής από το αρχείο :

2701

100	150	30	0	0	45	200	20	0	90
-----	-----	----	---	---	----	-----	----	---	----

 4

2	50	3	50	8	20	10	80
---	----	---	----	---	----	----	----

.....

Να γραφεί μια κλάση με όνομα **prod** που θα διαχειρίζεται και θα ομαδοποιεί τα δεδομένα από το αρχείο **prod2018.txt**. Τα δεδομένα θα διαβάζονται σειριακά από το αρχείο (απαιτείται έλεγχος για την ύπαρξη του αρχείου). Τα private μέλη – δεδομένα της κλάσης είναι :

1. κωδικός προϊόντος (1-10), τύπου int
2. μήνας (MM = μήνας (1-12)), τύπου int
3. συνολική μηνιαία παραχθείσα ποσότητα
4. συνολική μηνιαία ποσότητα παραγγελιών

Η κλάση πρέπει να περιέχει:

- Έναν constructor για την απόδοση μηδενικών τιμών στα μέλη των αντικειμένων της
- Έναν constructor για την εκχώρηση τιμών στα αντικείμενα της κλάσης μέσω της ανάγνωσης (προσπέλασης) του αρχείου.

Στη συνάρτηση main() να δηλώσετε έναν πίνακα αντικειμένων της κλάσης **prod** μεγέθους $N = 120$ θέσεων (10 προϊόντα X 12 μήνες), και να αρχικοποιήσετε όλες τις θέσεις του πίνακα με χρήση του κατάλληλου constructor, θέτοντας ως εξ ορισμού μηδενικές τιμές για όλα τα μέλη - δεδομένα.

Στη συνέχεια μέσω της προσπέλασης του αρχείου **prod2018.txt**:

1. Να γίνει εισαγωγή τιμών στα αντικείμενα του πίνακα, με τη χρήση του κατάλληλου constructor. Οι τιμές θα εισαχθούν α) κατά αύξουσα σειρά του κωδικού προϊόντος και β) κατά αύξουσα σειρά του μήνα. (ΥΠΟΔΕΙΞΗ : να χρησιμοποιήσετε έναν κατάλληλο δισδιάστατο ενδιάμεσο πίνακα).

2. Παράλληλα με το ερώτημα 1, να γίνει εισαγωγή σε έναν δυναμικό πίνακα τύπου `int`, τριάδων τιμών. Κάθε τριάδα θα περιλαμβάνει :
 - τον αριθμό του μήνα,
 - τον κωδικό προϊόντος και
 - την υπολειπόμενη ποσότητα παραγγελιών, όταν η ημερήσια παραχθείσα ποσότητα είναι μικρότερη από την ποσότητα που παραγγέλθηκε
3. Χρησιμοποιώντας όλα τα αντικείμενα του πίνακα αντικειμένων της κλάσης **prod**, να βρείτε και να εμφανίσετε τη συνολική ετήσια παραγωγή για κάθε προϊόν καθώς και τη συνολική ετήσια ποσότητα παραγγελιών για κάθε προϊόν.
4. Χρησιμοποιώντας όλα τα στοιχεία του δυναμικού πίνακα να βρείτε και να εμφανίσετε, για ένα προϊόν του οποίου ο κωδικός εισάγεται από το χρήστη: τη συνολική ανά μήνα υπολειπόμενη ποσότητα παραγγελιών.
5. Να δημιουργήσετε μια απορρέουσα κλάση με όνομα **new_prod**, που θα περιλαμβάνει ως επιπλέον private μέλη – δεδομένα, 10 διαφορετικές τιμές κόστους που σχετίζονται με την παραγωγή του προϊόντος, τύπου `double`.

Στη συνέχεια:

- a. Να γράψετε τη συνάρτηση του constructor (δομητή) για την εισαγωγή των δεδομένων σε ένα αντικείμενο της απορρέουσας κλάσης **new_prod**.
- b. Να ορίσετε έναν πίνακα αντικειμένων της κλάσης **new_prod**, μεγέθους N .
- c. Χρησιμοποιώντας το δομητή της απορρέουσας κλάσης, να αντιγράψετε σε αυτόν τον νέο πίνακα τα δεδομένα από τον πίνακα αντικειμένων της κλάσης **prod** που έχετε δημιουργήσει στο ερώτημα 1, δημιουργώντας τυχαία έγκυρα ποσά για τα νέα μέλη - δεδομένα σε κάθε θέση του νέου πίνακα.