

**Θέμα 2.**

Θεωρούμε το παρακάτω πρόγραμμα. Τι τυπώνεται κατά την εκτέλεσή του;

```
class A {
public:
    void p1() { cout << "g"; p2(); }
    virtual void p2() { cout << "a"; }
};

class B : public A {
public:
    void p1() { cout << "t"; p2(); }
    void p2() override { cout << "c"; }
};

int main() {
    A *p = new A;
    B *q = new B;
    p->p1();
    q->p1();
    p = q;
    q->p2();
    q->p1();
}
```

- A. gttcagc      B. gatccgc      **Γ. gatcctc**      Δ. κάτι άλλο

**Θέμα 3.**

Θεωρούμε το παρακάτω πρόγραμμα. Τι ισχύει για τον τελεστή == ;

```
class p {
    double x, y;
public:
    p(double _x, double _y) : x(_x), y(_y) {};
};

bool operator== (p p1, p p2) { return ((p1.x==p2.x) && (p1.y==p2.y)); }

int main() {
    p p1(1, 2), p2(1, 2);
    cout << ((p1==p2)?"equal":"different");
}
```

- A. λειτουργεί ως έχει B. πρέπει να δηλωθεί ως μέλος της κλάσης p  
Γ. πρέπει να δηλωθεί ως friend της κλάσης p      Δ. οποιοδήποτε από τα (β) ή (γ)

**Θέμα 4.**

Θεωρούμε το παρακάτω πρόγραμμα. Τι τυπώνεται κατά την εκτέλεσή του;

```
class C {
public:
    static void setX(int a) { x -= a; }
    C(int b): y(b) {}
    void status() const {
        cout << " " << x << " " << y;
    }
private:
    static int x;
    int y;
};

int C::x = 7;

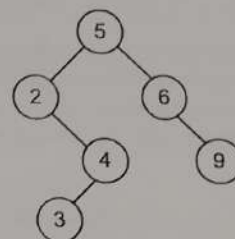
int main() {
    C c1(0);
    c1.status();
    c1.setX(1); c1.setX(2);
    c1.status();
    C c2(3); c2.status();
    c2.setX(4); c2.status();
}
```

- A. 7 0 4 0 7 3 3 3      B. 7 0 2 0 2 3 4 3      **Γ. 7 0 4 0 4 3 0 3**      Δ. Κανένα από τα προηγούμενα

**Θέμα 5.**

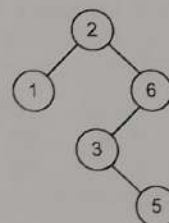
Δίνεται το δένδρο του διπλανού σχήματος. Ποιο είναι το τρίτο στοιχείο που θα επεξεργαστεί ο αλγόριθμος ενδοδιατεταγμένης (in-order) διάσχισης κατά βάθος;

- A. Το 2.
- B. Το 3.
- ☒ Γ. Το 4.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.

**Θέμα 6.**

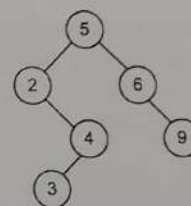
Δίνεται το δένδρο του παρακάτω σχήματος. Πόσα διαφορετικά μονοπάτια έχει αυτό το δένδρο;

- A. 2.
- ☒ B. 7.
- Γ. 12.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.

**Θέμα 7.**

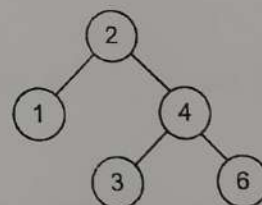
Δίνεται το δυαδικό δένδρο αναζήτησης του διπλανού σχήματος, από το οποίο διαγράφεται το κλειδί 5. Μετά την εφαρμογή του αλγόριθμου, τελικά το αριστερό παιδί της ρίζας θα έχει κλειδί:

- A. 2.
- B. 3.
- ☒ Γ. 4.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.

**Θέμα 8.**

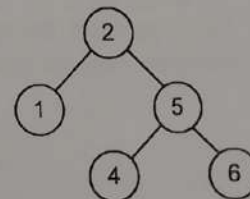
Στο δένδρο AVL του παρακάτω σχήματος διαγράφεται το κλειδί 1. Ποια πρόταση που αφορά τον κόμβο με το κλειδί 4 είναι ορθή (ελέγχουμε την ετικέτα του κόμβου που έχει το κλειδί 4 στο αρχικό AVL δένδρο και στο τελικό AVL δένδρο);

- ☒ A. Θα αλλάξει ετικέτα, θα γίνει αριστερά ψηλός.
- B. Θα αλλάξει ετικέτα, θα γίνει δεξιά ψηλός.
- Γ. Θα παραμείνει με την ίδια ετικέτα.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.

**Θέμα 9.**

Στο δένδρο AVL του διπλανού σχήματος εισάγεται το κλειδί 3. Ποια πρόταση που αφορά τον κόμβο με το κλειδί 5 είναι ορθή (ελέγχουμε την ετικέτα του κόμβου που έχει το κλειδί 5 στο αρχικό AVL δένδρο και στο τελικό AVL δένδρο);

- A. Θα αλλάξει ετικέτα, θα γίνει αριστερά ψηλός.
- B. Θα αλλάξει ετικέτα, θα γίνει δεξιά ψηλός.
- Γ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.
- Δ. Θα παραμείνει με την ίδια ετικέτα.



**Θέμα 10.**

Σε ένα δένδρο AVL, δίνεται η ακολουθία εντολών  $\text{ins}(4)$ ,  $\text{ins}(5)$ ,  $\text{ins}(6)$ ,  $\text{ins}(3)$ ,  $\text{ins}(4)$ ,  $\text{ins}(2)$ ,  $\text{del}(5)$ ,  $\text{del}(4)$ ,  $\text{del}(6)$ ,  $\text{del}(3)$ ,  $\text{del}(2)$ . Κατά την εφαρμογή του γνωστού αλγορίθμου εισαγωγής και διαγραφής στοιχείων σε AVL δένδρα, συνολικά θα γίνουν:

- A. 2 αριστερές περιστροφές.
- B. 1 αριστερή και 1 δεξιά περιστροφή.
- Γ. Περισσότερες από 2 περιστροφές.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.

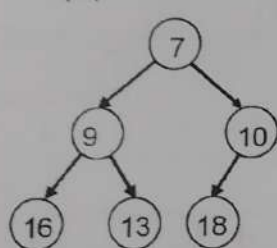
**Θέμα 11.**

Δίνεται ένας πίνακας κατακερματισμού 25 θέσεων, για τον οποίο επιλέγεται η μέθοδος της διαιρέσης ( $\text{mod } 25$ ) ως μέθοδος κατακερματισμού με αλυσίδωση. Έστω ότι επιθυμούμε να αποθηκεύσουμε κλειδιά που μπορούν να πάρουν τιμές στο διάστημα των ακεραίων  $[1,100]$ . Ποια από τις παρακάτω ακολουθίες εισαγωγής κλειδιών θα οδηγήσει σε σχηματισμό αλυσίδας μήκους τουλάχιστον 3 σε κάποια θέση του πίνακα;

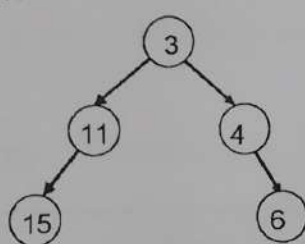
- A. 3, 6, 9, 12, ..., 75
- B. 5, 6, 7, 8, ..., 54
- Γ. 1, 3, 5, 7, ..., 75
- Δ. καμία από τις παραπάνω

**Θέμα 12.**

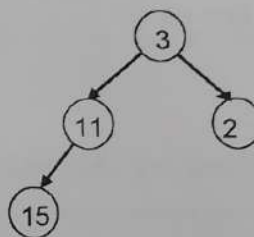
Δίνονται τα παρακάτω δένδρα. Ποια από αυτά είναι σωροί ελαχίστου (στην απάντηση πρέπει να αναφέρονται όλα όσα είναι);



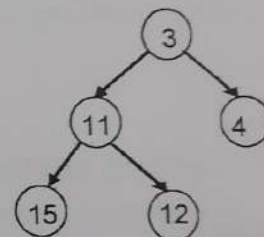
1



2



3



4

A. Τα 1 και 4.

B. Τα 2 και 4.

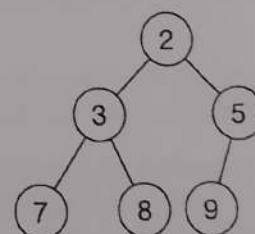
Γ. Μόνο το 4.

Δ. Τα 1 και 3.

**Θέμα 13.**

Δίνεται ο σωρός ελαχίστου του διπλανού σχήματος. Αν εφαρμόσουμε τον αλγόριθμο εισαγωγής για το στοιχείο 1, τι από τα παρακάτω θα συμβεί;

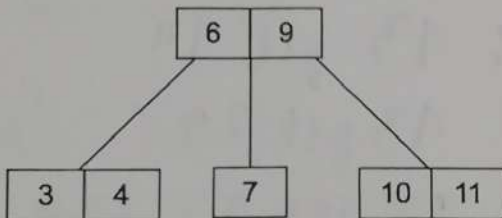
- A. Το στοιχείο 1 θα βρεθεί στη θέση του στοιχείου 2.
- B. Το στοιχείο 1 θα βρεθεί στη θέση του στοιχείου 5.
- Γ. Το στοιχείο 5 θα βρεθεί στη θέση του στοιχείου 9.
- Δ. Κανένα από τα προηγούμενα.



#### Θέμα 14.

Δίνεται το B-δένδρο 3 οδεύσεων του παρακάτω σχήματος, στο οποίο γνωρίζουμε ότι εισάγεται το κλειδί 12 και διαγράφεται το κλειδί 9, χρησιμοποιώντας τους γνωστούς αλγορίθμους, δεν γνωρίζουμε όμως με ποια σειρά γίνονται οι δύο αυτές λειτουργίες. Ποια από τις παρακάτω προτάσεις που το αφορούν είναι σωστή;

- A. Το τελικό δένδρο θα είναι το ίδιο, με όποια σειρά και να γίνουν οι λειτουργίες αυτές.
- ☒ B. Τα δύο τελικά δένδρα (ανάλογα με τη σειρά με την οποία θα γίνουν οι λειτουργίες) θα είναι διαφορετικά αλλά θα έχουν το ίδιο ύψος.
- Γ. Τα δύο τελικά δένδρα (ανάλογα με τη σειρά με την οποία θα γίνουν οι λειτουργίες) θα έχουν διαφορετικό ύψος.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.



#### Θέμα 15.

Σε ένα B-δένδρο τριών οδεύσεων, δίνεται η ακολουθία εντολών  $\text{ins}(5)$ ,  $\text{ins}(3)$ ,  $\text{ins}(4)$ ,  $\text{ins}(6)$ ,  $\text{del}(4)$ ,  $\text{ins}(4)$ ,  $\text{ins}(7)$ . Κατά την εφαρμογή του γνωστού αλγορίθμου εισαγωγής και διαγραφής στοιχείων σε B-δένδρα, συνολικά για την εξισορρόπηση θα χρειαστεί να γίνουν:

- A. Ένας διαχωρισμός κόμβου και μία περιστροφή.
- B. Μόνο ένας διαχωρισμός.
- Γ. Δύο περιστροφές.
- Δ. Καμία από τις άλλες προτάσεις δεν είναι σωστή.



### Μέρος Β. Ελεύθερα ερωτήματα

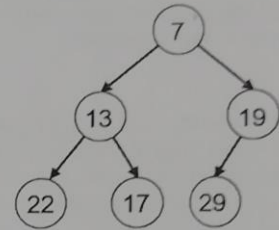
Απαντήστε στο κενό κάτω από το ερώτημα.

#### Θέμα 16. Σωροί (5 μονάδες)

Δίνεται ο σωρός ελαχίστου του διπλανού σχήματος:

α) Δώστε τη μορφή του σωρού μετά την διαγραφή του ελάχιστου στοιχείου. Ποιες συγκρίσεις θα πραγματοποιηθούν; (θα πρέπει να αναφέρονται με τη σειρά που θα γίνουν)

β) Δώστε τη μορφή του σωρού μετά την εισαγωγή στοιχείου με τιμή 7 (αφού προηγηθεί η διαγραφή του ερωτήματος α). Ποιες συγκρίσεις θα πραγματοποιηθούν; (θα πρέπει να αναφέρονται με τη σειρά που θα γίνουν)

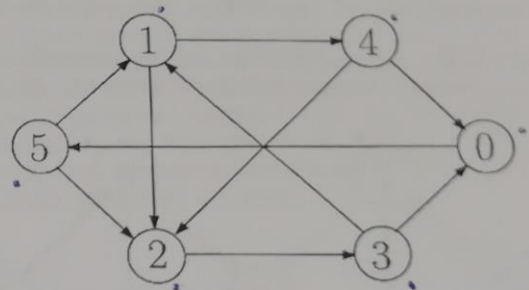


#### Θέμα 17. Αλγόριθμοι BFS/DFS (5 μονάδες)

Δίνεται ο γράφος του παρακάτω σχήματος.

α) Εκτελέσετε τον αλγόριθμο εξερεύνησης κατά πλάτος (BFS) στον γράφο αυτό με αφετηρία τον κόμβο  $x = (3 * AM\_last + 1) \bmod 6$ . [Σημείωση:  $AM\_last$  = το τελευταίο ψηφίο του Αρ. Μητρώου σας]

Δείξτε την σειρά επίσκεψης των κόμβων (θεωρήστε ότι οι ισοπαλίες επιλύονται αλφαβητικά). Δείξτε επίσης τα περιεχόμενα της ουράς μετά την εξερεύνηση του τρίτου κατά σειρά κόμβου.

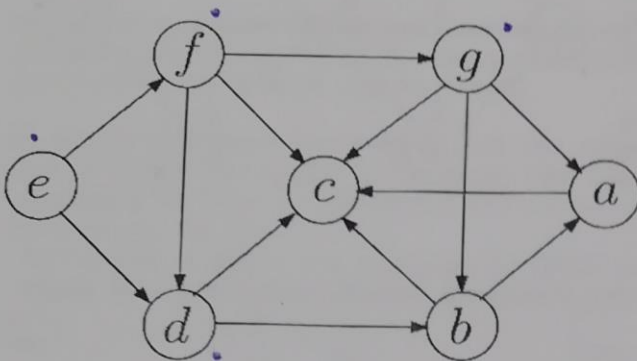


β) Εκτελέστε τον αλγόριθμο εξερεύνησης κατά βάθος (DFS) με αφετηρία τον κόμβο  $x$ . Δείξτε την σειρά επίσκεψης των κόμβων (θεωρήστε ότι οι ισοπαλίες επιλύονται αλφαβητικά). Ποιες back edges θα βρει ο αλγόριθμος;

#### Θέμα 18. Τοπολογική ταξινόμηση (5 μονάδες)

Δώστε μία τοπολογική ταξινόμηση για τον γράφο του σχήματος. Αναφέρετε τη μέθοδο που ακολουθήσατε και εξηγήστε με λίγα λόγια γιατί δίνει το αποτέλεσμα που βρήκατε.

ΠΡΟΣΟΧΗ: σε όποια μέθοδο και αν ακολουθήσετε, οποτεδήποτε υπάρχει δυνατότητα επιλογής μεταξύ δύο ή περισσότερων κόμβων θα πρέπει να επιλέξετε αυτόν που προηγείται αλφαβητικά.



### Θέμα 19. Ταξινομημένα διανύσματα (10 μονάδες)

Υλοποιήστε την κλάση `sorted_vector` η οποία θα περιλαμβάνει **ταξινομημένα** στοιχεία με τις εξής επιπλέον προδιαγραφές: α) Θα πρέπει να αποθηκεύει γενικού τύπου στοιχεία, β) Επειδή θα ζητείται συχνά το άθροισμα των στοιχείων, θα πρέπει να υποστηρίζεται κατάλληλη μέθοδος που να το επιστρέφει σε σταθερό χρόνο ( $O(1)$ ), γ) Θα πρέπει να δίνεται η δυνατότητα τυχράς ανάγνωσης ενός στοιχείου μέσω του `operator []` (π.χ. `int x = v[5]`) αλλά να μην επιτρέπεται η εγγραφή του στοιχείου (π.χ. να μην επιτρέπεται `v[5] = x`) γιατί έτσι θα υπήρχε κίνδυνος να καταλήξουμε σε μη ταξινομημένο διάνυσμα, δ) να υποστηρίζεται το άθροισμα δύο διανυσμάτων μέσω του `operator +` αν τα διανύσματα έχουν το ίδιο μέγεθος (επιλέξτε ελεύθερα τι θα συμβαίνει αν τα διανύσματα δεν έχουν το ίδιο μέγεθος), ε) Να υποστηρίζεται εκτύπωση του διανύσματος (των στοιχείων του και του αθροίσματός τους) μέσω του `operator <<`, στ) Να υποστηρίζεται προσθήκη ενός νέου στοιχείου.

### Θέμα 20. Ταμειακή μηχανή (10 μονάδες)

Σε μια ταμειακή μηχανή καταχωρούνται μέσω αναγνώστη γραμμωτού κώδικα το όνομα και η τιμή κάθε είδους που σκανάρεται στο ταμείο με τυχαία σειρά. Αν το ίδιο είδος έχει αγοραστεί περισσότερες από μία φορές, τότε σκανάρεται χωριστά κάθε τεμάχιο. Ωστόσο στο λογαριασμό που εκτυπώνεται, για κάθε είδος εμφανίζεται μία φορά το όνομά του, η ποσότητα, η τιμή μονάδας και το σύνολο, με τη σειρά που σκανάρονται τα είδη. Γράψτε ένα κομψό και αποδοτικό πρόγραμμα που διαβάζει από την τυπική είσοδο (`standard input`) σε κάθε γραμμή το όνομα και την τιμή κάθε είδους που σκανάρεται και εμφανίζει στην τυπική έξοδο (`standard output`) τον τελικό λογαριασμό, ο οποίος περιλαμβάνει και το σύνολο, όπως φαίνεται στο παράδειγμα.

```
$ cat input1.txt
chair 24
chair 24
desk 80
bed 140
chair 24
table 100
sofa 300
table 100
chair 24
bed 500
table 100
carpet 184
```

```
$ ./a.out < input1.txt
chair 4 x 24 = 96
desk 1 x 80 = 80
bed 2 x 140 = 640
table 3 x 100 = 300
sofa 1 x 300 = 300
carpet 1 x 184 = 184
Total: 1600
```

Υπόδειξη: Να κατασκευάσετε μια κλάση **purchase**, η οποία να περιέχει κατάλληλη κλάση της βιβλιοθήκης STL για την καταχώρηση κάθε γραμμής του τελικού λογαριασμού, καθώς και μεθόδους για το σκανάρισμα είδους και την εμφάνιση του τελικού λογαριασμού. Θα χρειαστεί να χρησιμοποιήσετε αναζήτηση (**find**) έχοντας υπερφορτώσει κατάλληλα τον τελεστή `"=="`.

### Θέμα 21. Δυαδικό δένδρο (10 μονάδες)

Ορίστε τον τύπο **node** του κόμβου δυαδικού δένδρου που περιέχει ως πληροφορία δεδομένα τύπου **int**.

Στη συνέχεια, γράψτε μία κομψή και αποδοτική συνάρτηση η οποία να δέχεται ως παράμετρο ένα δείκτη προς τη ρίζα ενός τέτοιου δένδρου **t** και μία τιμή κλειδιού **k**. Η συνάρτησή σας θα πρέπει να επιστρέφει `true` αν το δέντρο έχει ένα μονοπάτι από τη ρίζα σε κάποιο φύλλο, τέτοιο ώστε το άθροισμα των τιμών στο μονοπάτι (συμπεριλαμβανομένης της ρίζας και του φύλλου) να είναι ίσο με την τιμή **k**, διαφορετικά `false`.

Παράδειγμα: Για το δένδρο του διπλανού σχήματος και για  $k=36$ , η συνάρτησή σας πρέπει να επιστρέφει `true`.

