Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής & Υπολογιστών
https://helios.ntua.gr/course/view.php?id=869

Προγραμματιστικές Τεχνικές

Άσκηση 8 Πίνακας με μορφή σκακιέρας

Προθεσμία υποβολής στον grader: 24/4/2026

Υλοποιήστε την κλάση `ChessBoardArray` για την αναπαράσταση τετραγωνικών πινάκων, στους οποίους τα μη μηδενικά στοιχεία μπορούν να βρίσκονται μόνο στις θέσεις που σε μία σκακιέρα θα αντιστοιχούσαν σε λευκά τετράγωνα (βλ. σχήμα). Χάριν ευκολίας, θα θεωρήσουμε ότι οι πίνακες αυτοί περιέχουν ακέραιους αριθμούς και ότι ο default constructor κατασκευάζει πίνακες που έχουν παντού μηδενικά.

0		1		2		3	
	4		5		6		7
8		9		10		11	
	12		13		14		15
16		17		18		19	
	20		21		22		23
24		25		26		27	
	28		29		30		31

Η κλάση σας πρέπει να υποστηρίζει τις εξής λειτουργίες, τις οποίες πρέπει να υλοποιήσετε:

```
1  class ChessBoardArray {
2  protected:
3    class Row {
4    public:
5      Row(ChessBoardArray &a, int i);
6      int & operator [] (int i) const;
7    };
8
9    class ConstRow {
10   public:
11     ConstRow(const ChessBoardArray &a, int i);
12     int operator [] (int i) const;
13   };
14
15 public:
16   ChessBoardArray(unsigned size = 0, unsigned base = 0);
17   ChessBoardArray(const ChessBoardArray &a);
18   ~ChessBoardArray();
19
20   ChessBoardArray & operator = (const ChessBoardArray &a); {
21
22   int & select(int i, int j);
23   int select(int i, int j) const;
```

```
24
25    const Row operator [] (int i);
26    const ConstRow operator [] (int i) const;
27
28    friend ostream & operator << (ostream &out, const ChessBoardArray &a);
29  };
```

Συμβουλευτείτε τις διαφάνειες και το υλικό της διάλεξης #5 για τη σημασία των παραπάνω λειτουργιών. Ορίστε και χρησιμοποιήστε μία κατάλληλη ιδιωτική μέθοδο `loc`:

```
1   unsigned int loc(int i, int j) const;  // may throw out_of_range
```

Η εκτύπωση των πινάκων πρέπει να εμφανίζει και τα μηδενικά στοιχεία όπως φαίνεται παρακάτω. Χρησιμοποιήστε `setw(4)` για να στοιχίσετε τους αριθμούς (ορίζεται στο `<iomanip>`, ψάξτε το!).

```
1     0   0   1   0   2   0   3   0
2     0   4   0   5   0   6   0   7
3     8   0   9   0  10   0  11   0
4     0  12   0  13   0  14   0  15
5    16   0  17   0  18   0  19   0
6     0  20   0  21   0  22   0  23
7    24   0  25   0  26   0  27   0
8     0  28   0  29   0  30   0  31
```

Μπορείτε να δοκιμάσετε την υλοποίησή σας με προγράμματα όπως το εξής:

```
1   int main() {
2     ChessBoardArray a(4, 1);  // size 4x4, rows and columns numbered from 1
3     a[3][1] = 42;
4     a[4][4] = 17;
5     try { a[2][1] = 7; }
6     catch(out_of_range &e) { cout << ”a[2][1] is black” << endl; }
7     try { cout << a[1][2] << endl; }
8     catch(out_of_range &e) { cout << ”and so is a[1][2]” << endl; }
9     cout << a;
10  }
```

Η εκτέλεσή του θα πρέπει να εμφανίζει:

```
1   a[2][1] is black
2   and so is a[1][2]
3     0   0   0   0
4     0   0   0   0
5    42   0   0   0
6     0   0   0  17
```

Στο αρχείο που θα ανεβάσετε στον grader θα πρέπει να συμπεριλάβετε (μόνο) τη δήλωση της κλάσης `ChessBoardArray` και τις υλοποιήσεις των μεθόδων της.

Άσκηση 9 Πολυώνυμα με μορφή ταξινομημένης συνδεδεμένης λίστας

Προθεσμία υποβολής στον grader: 24/4/2026

Υλοποιήστε την κλάση `Polynomial` για την αναπαράσταση πολυωνύμων μίας μεταβλητής (ας την ονομάσουμε x), με ακέραιους συντελεστές. Τα πολυώνυμα θα πρέπει να αναπαρασταθούν με τη μορφή ταξινομημένης απλά συνδεδεμένης λίστας: οι κόμβοι αυτής θα πρέπει να είναι ταξινομημένοι σε *φθίνουσα σειρά του εκθέτη*. Δε θα πρέπει να υπάρχουν δύο κόμβοι με τον ίδιο εκθέτη και δε θα πρέπει να υπάρχουν κόμβοι με μηδενικούς συντελεστές.

Η κλάση σας πρέπει να υποστηρίζει τις εξής λειτουργίες, τις οποίες πρέπει να υλοποιήσετε:

```
class Polynomial {
 protected:
  class Term {
   protected:
    int exponent;
    int coefficient;
    Term *next;
    Term(int exp, int coeff, Term *n);
    friend class Polynomial;
  };

 public:
  Polynomial();
  Polynomial(const Polynomial &p);
  ~Polynomial();

  Polynomial & operator = (const Polynomial &p) {

  void addTerm(int expon, int coeff);
  double evaluate(double x);

  friend Polynomial operator+ (const Polynomial &p, const Polynomial &q);
  friend Polynomial operator* (const Polynomial &p, const Polynomial &q);

  friend ostream & operator << (ostream &out, const Polynomial &p);
};
```

Συμβουλευτείτε τις διαφάνειες των σχετικών διαλέξεων για τον τρόπο υλοποίησης συνδεδεμένων λιστών. Η εκτύπωση των πολυωνύμων πρέπει να τα εμφανίζει όπως στα παρακάτω παραδείγματα:

```
x^2 + 3x - 1
2x^5 + 1
- x^7 - 3x^3 - x
42
0
```

Μπορείτε να δοκιμάσετε την υλοποίησή σας με προγράμματα όπως το εξής:

```
int main() {
  Polynomial p;      // 0
  p.addTerm(1, 3);   // 3x
```

```
  p.addTerm(2, 1);   // x^2
  p.addTerm(0, -1);  // -1

  Polynomial q(p);   // x^2 + 3x - 1
  q.addTerm(1, -3);  // -3x

  cout << "P(x) = " << p << endl;
  cout << "P(1) = " << p.evaluate(1) << endl;
  cout << "Q(x) = " << q << endl;
  cout << "Q(1) = " << q.evaluate(1) << endl;
  cout << "(P+Q)(x) = " << p+q << endl;
  cout << "(P*Q)(x) = " << p*q << endl;
}
```

Η εκτέλεσή του θα πρέπει να εμφανίζει:

```
P(x) = x^2 + 3x - 1
P(1) = 3
Q(x) = x^2 - 1
Q(1) = 0
(P+Q)(x) = 2x^2 + 3x - 2
(P*Q)(x) = x^4 + 3x^3 - 2x^2 - 3x + 1
```

Στο αρχείο που θα ανεβάσετε στον grader θα πρέπει να συμπεριλάβετε (μόνο) τη δήλωση της κλάσης `Polynomial` και τις υλοποιήσεις των μεθόδων της.