Tema 1

Precizari generale

Fiecare tema trebuie insotita de un mic exemplu care sa demonstreze utilizarea functionalitatilor implementate.
Urmatoarele cerinte sunt valabile pentru oricare dintre temele listate:

- Alocarea dinamica a memoriei;
- Separarea fiecarei clase in fisiere individuale de tip header (.h), respectiv sursa (.cpp);
- Utilizarea constructorilor cu parametri, fara parametri si cel de copiere;
- Supraincarcarea operatorilor de intrare si iesire (<<, respectiv >>);
- Supraincarcarea operatorului =;
- Utilizarea destructorului;
- Utilizarea unei conventii de denumire a metodelor si a variabilelor;
- Indentarea adecvata a codului;
- Comentarii in cod care sa ofere detalii despre modul de implementare.

* Nota: Functiile/metodele identificate in cerinte (fie ca este vorba de supraincarcari de operatori, fie de alte functionalitati), pot fi implementate ca functii friend in loc de metode ale claselor respective, daca se considera ca aceasta alegere este mai naturala.

Termen predare: 24 martie, ora 23:59

Se accepta o depasire a termenului limita (de maxim 2 zile) cu penalitate -1p/zi

Teme

1. Clasa pentru un numar complex, care sa ofere urmatoarele functionalitati:

- supraincarcarea operatorului * pentru inmultirea a doua numere complexe, sau a unui numar complex cu un real;
- supraincarcarea operatorului / pentru impartirea a doua numere complexe, sau a unui numar complex cu un real;
- supraincarcarea operatorului ==, pentru testarea egalitatii a doua numere complexe, sau a unui complex cu un real;
- obtinerea partii reale;
- obtinerea partii imaginare;
- functie pentru obtinerea modulului (abs), respectiv a normei (norm).

2. Clasa pentru un numar rational, care sa ofere urmatoarele functionalitati:

- metoda interna pentru simplificare; numerele se vor memora in forma canonica (numaratorul si numitorul sa fie prime intre ele, numitorul > 0);

- supraincarcarea operatorului * pentru inmultirea a doua numere rationale, sau a unui numar rational cu un intreg;
- supraincarcarea operatorului / pentru impartirea a doua numere rationale, sau a unui numar rational cu un intreg;
- supraincarcarea operatorului ^ pentru ridicarea la putere naturala a unui numar rational;
- supraincarcarea operatorului <, respectiv >, pentru compararea a doua numere rationale;
- supraincarcarea operatorului ==, pentru testarea egalitatii a doua numere rationale;
- obtinerea numaratorului;
- obtinerea numitorului.

3. Clasa pentru un string (sir de caractere), care sa ofere urmatoarele functionalitati:

- un singur caracter (char) poate fi vazut ca un string de un element;
- setarea sirului de caractere;
- supraincarcarea operatorului ==, pentru testarea egalitatii caracterelor a doua stringuri;
- supraincarcarea operatorului <, respectiv >, pentru compararea lexicografica a doua stringuri;
- supraincarcarea operatorului + pentru concatenarea a doua stringuri;
- supraincarcarea operatorului [ ] pentru obtinerea caracterului de pe pozitia i;
- obtinerea numarului de caractere.

4. Clasa pentru o stiva, care sa ofere urmatoarele functionalitati:

- inserarea (push), stergerea (pop) si interogarea elementului din varful stivei (top);
- supraincarcarea operatorului ==, pentru testarea egalitatii elementelor a doua stive;
- supraincarcarea operatorului <, respectiv >, pentru compararea lexicografica a doua stive;
- stabilirea daca stiva este vida;
- obtinerea numarului de elemente;
- functie pentru inversarea unei stive folosind numai metodele publice.

5. Clasa pentru o coada, care sa ofere urmatoarele functionalitati:

- inserarea (put), stergerea (get) si interogarea elementului din varful cozii (front), respectiv din fundul conzii (back);
- supraincarcarea operatorului ==, pentru testarea egalitatii elementelor a doua cozi;
- supraincarcarea operatorului <, respectiv >, pentru compararea lexicografica a doua cozi;
- stabilirea daca coada este vida;
- obtinerea numarului de elemente;

- functie pentru inversarea unei cozi folosind numai metodele publice.

6. Clasa pentru un arbore binar de cautare, care sa ofere urmatoarele functionalitati:

- inserarea, stergerea si cautarea unui element;
- parcurgerea in inordine, preordine, respectiv postordine;
- obtinerea elementului minim, respectiv maxim;
- supraincarcarea operatorilor <, respectiv >, pentru compararea inaltimii a doi arbori;
- obtinerea inaltimii arborelui;
- obtinerea numarului de elemente.

7. Clasa pentru o multime (ordonata crescator), care sa ofere urmatoarele functionalitati:

- fiecare element sa apara o singura data;
- inserarea, stergerea si cautarea unui element;
- supraincarcarea operatorului + pentru reuniunea a doua multimi;
- supraincarcarea operatorului - pentru diferenta a doua multimi;
- supraincarcarea operatorilor <, respectiv >, pentru compararea cardinalului a doua multimi;
- supraincarcarea operatorului [ ] pentru obtinerea elementului de pe pozitia i;
- obtinerea numarului de elemente.

8. Clasa pentru un graf neorientat, care sa ofere urmatoarele functionalitati:

- parcurgerea in latime si adancime pornind dintr-un nod dat;
- obtinerea distantei dintre doua noduri date;
- obtinerea listei de adiacenta a unui nod dat;
- supraincarcarea operatorului + pentru adunarea a doua grafuri avand acelasi nr. de noduri, rezultand astfel un graf cu acelasi nr. de noduri, dar cu multimea muchiilor egala cu reuniunea multimilor muchiilor acelor doua grafuri;
- spraincarcarea operatorului - (cu un intreg) pentru eliminarea unui nod;
- stabilirea daca graful este arbore;
- stabilirea daca graful este conex;
- obtinerea numarului de noduri, respectiv muchii;

9. Clasa pentru o matrice, care sa ofere urmatoarele functionalitati:

- supraincarcarea operatorului + pentru adunarea a doua matrici;
- supraincarcarea operatorului * pentru inmultirea a doua matrici, sau a unei matrici cu un real;
- supraincarcarea operatorului - pentru scaderea a doua matrici;
- supraincarcarea operatorului [ ] pentru obtinerea (sau modificarea) liniei i;
- obtinerea determinantului matricei;

- obtinerea numarului de linii, respectiv coloane.

10. Clasa pentru un polinom (reprezentat ca tablou unidimensional), care sa ofere urmatoarele functionalitati:

- calcularea valorii polinumului intr-un anumit punct;
- supraincarcarea operatorului + pentru adunarea a doua polinoame;
- supraincarcarea operatorului * pentru inmultirea a doua polinoame, sau a unui polinom cu un real;
- supraincarcarea operatorului / pentru impartirea a doua polinoame;
- supraincarcarea operatorului [ ] pentru obtinerea (sau setarea) termenului cu gradul i;
- obtinerea gradului ponilomului.

11. Clasa pentru polinom (reprezentat ca lista inlantuita), oferind aceleasi functionalitati ca la pct. anterior.