Задача 1.

Да се дефинира клас Fraction, съдържаш две член-данни (цели числа) n и d (d != 0), конструктор и функции за достъп до компонентите. Да се дефинира функция isFraction, която определя дали n не се дели целочислено на d, като :

1. Член-функция на класа
2. Външна функция
3. Приятелска функция на класа

Задача 2.

Да се реализира свързано представяне на шаблонен стек. Използвайте структурата

template <typename T>

struct elem {

T inf;

elem\* link; // elem<T>\* link;

};

Създайте template-ен клас Stack, който има поле – елемент на структурата, два private метода deleteStack() и copyStack(const Stack&) и следния интерфейс :

void push(const T&);

void pop(T&);

bool empty() const;

void print();

**Необходимо е да реализирате канонично представяне на класа Stack! (Защо?)**

Задача 3.

Да се дефинира клас Time, който определя време по зададени час и минути. Класът да съдържа подходящи конструктори, мутатор, променящ часа и минутите, мутатори, добавящи към време цяло число, означаващо час и минути, както и член-функция, която предефинира оператора за преобразуване на типовете така, че да може да се преобразува време в цяло число, означаващо времето, изразено в минути. Да се предефинират операторите +, -, \* така че да могат да се събират и изваждат две времена, както и да може да се умножава време с цяло число и цяло число с време. Да се включи дефинираният клас в програма и направят обръщения към член-функциите му и предефинират операторите.

Пояснение :

Класът Time да съдържа две полета – hours и minutes както и private метод simplify(), който да оправя минутите и часовете в случай не прехвърляне. Мутаторите имат следните сигнатури :

void reset(int = 0, int = 0)

void addMin(int)

void addHours(int)