**Tasks**

# Problem 1:

Create a class Money that represents a money valuer (combination of dollars and cents).

Overload the following operator. Functions:

1. **Money()** // Default Constructor Initializes dollars and cents to zero
2. **Money(int dollar, int cents)** // Parameterized Constructor Update dollar and cents accordingly
3. **Money& operator= (const Money& right)** Overload Assignment operator to assign Money objects to each other
4. **bool operator== (const Money& right)** Overload Equal operator to check if Money objects are equal or not
5. **Money& operator+= (const Money& right)** Overload Addition operator to Add Money objects to each other
6. **Money& operator-= (const Money& right)** Overload Subtraction operator to Subtract smaller Money object from larger Money object.
7. **Money& operator\*= (int a)** Overload Multiplication operator to multiply Money object with an integer number
8. **Money& operator/= (int a)** Overload Division operator to divide Money object with an integer number
9. **Money& operator++ ():** Overload the ++ operator for pre increment in the dollars(i.e add 1 to dollars).
10. **Money& operator++ (int a):** Overload the ++ operator for post increment in the dollars (i.e add 1 to dollars).
11. **~Money()**

# Problem 2:

Design a class String having a data member as char \*s, int size, and define the appropriate function members,

you task is to

1. **Overload binary operator ‘+’ to concatenate two strings.** For example: If you pass string “hello” and “world” for s1 and s2 respectively, it will concatenate both into s3, and display the output as “helloworld”.
2. **Overload comparison operators (< , >, ==);**

# Problem 3:

Implement 2-dimensional matrix class, and overload the Sum ‘+’ operator to add two matrices.

Your solution will have following methods:

* 1. Default Constructor
  2. Parameterized Constructor
  3. Copy Constructor
  4. Destructor
  5. To insert data in matrix
  6. Overload + operator to add two matrix objects