Lab 10

Question 1

Implement the matrix class is implemented with three members

- o Int **matrix
- o Int rows
- o Int cols

Implement Constructor(s), destructor, allocation, deallocation, initialization and printing of a matrix. The main function is also given, you are not allowed to modify that.

```
int main() {

    // Create two matrices
    Matrix mat1(2, 3);
    Matrix mat2(3, 2);

    // Set values for mat1
    mat1.initalizeMatrix(1); // starting value = 1, initialize matrix with 1..6
    mat2.initalizeMatrix(7); // starting value = 7, initialize matrix with 7..12

    // Print matrices
    cout << "Matrix 1:" << endl;
    mat1.print();

    cout << "Matrix 2:" << endl;
    mat2.print();

    Matrix mat4;
    mat4 = mat1;

    cout << "Matrix 4:" << endl;
    mat4.print();

    Matrix mat5 = mat2;
    cout << "Matrix 5:" << endl;
    mat5.print();

    return 0;
}
```

Question 2

Create a C++ class that models a basic Bank Account. The class should include functionalities such as depositing money, withdrawing money, and displaying account details.

Requirements:

1. Define a class called BankAccount:
    - Data members:
        - accountNumber (string)
        - accountHolderName (string)
        - balance (float)
    - Constructor: Initializes account with the account number, holder's name, and initial balance.
    - Destructor: Displays a message when the account object is destroyed.
    - Member functions:
        - deposit(float amount): Adds money to the balance.
        - withdraw(float amount): Subtracts money from the balance if there are sufficient funds, else print an error message.
        - displayAccountDetails(): Displays account number, holder's name, and current balance.
2. Create a main function where:
    - You create an object of BankAccount.
    - Perform deposit and withdrawal operations.
    - Display the account details after each operation.

Question 3

Your task is to implement findPairsWithSum(int targetValue) method to display pairs (with in a matrix class) that sum to a given target value. The main function is given below. You are NOT allowed to make a temporary array, it means don't copy matrix items in a 1D or any temporary storage, then make & print the combinations easily. Don't even think about that. Make combinations by traversing a matrix.

```
int main() {

    Matrix mat1(4,4);
    mat1.initalizeMatrix(1);      // initialize matrix with 1..16
    cout << endl << "Displaying the matrix " << endl;
    mat1.printMatrix();

    // TASK: Display all pairs that add up to a target value
    cout << endl << "testing code for target value = 12" << endl;
    mat1.findPairsWithSum(12); //target value = 12

    cout << endl << "testing code for target value = 7" << endl;
    mat1.findPairsWithSum(7); //target value = 7

    cout << endl << "testing code for target value = 18" << endl;
    mat1.findPairsWithSum(18); //target value = 18

    return 0;
}
```