/\*Task=01 : \*/

#include <iostream>

#include <string>

using namespace std;

class RomanNumeral

{

private:

string numeral;

public:

RomanNumeral(string num)

{

numeral = num;

}

int toDecimal()

{

int decimal = 0;

int prevDigit = 0;

for (int i = numeral.length() - 1; i >= 0; i--)

{

int digit;

switch (numeral[i])

{

case 'I':

digit = 1;

break;

case 'V':

digit = 5;

break;

case 'X':

digit = 10;

break;

case 'L':

digit = 50;

break;

case 'C':

digit = 100;

break;

case 'D':

digit = 500;

break;

case 'M':

digit = 1000;

break;

default:

cout << "Invalid Roman numeral entered." << endl;

return -1;

}

if (digit < prevDigit)

{

decimal -= digit;

}

{

decimal = decimal + digit;

}

prevDigit = digit;

}

return decimal;

}

};

int main()

{

string numeral;

cout << "Enter a Roman numeral: ";

cin >> numeral;

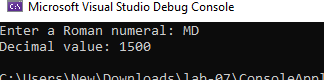
RomanNumeral roman(numeral);

int decimal = roman.toDecimal();

cout << "Decimal value: " << decimal << endl;

return 0;

}



/\*Task-04 : \*/

#include <iostream>

#include <string>

using namespace std;

class bankAccount

{

private:

string name;

int accountNumber;

string accountType;

double balance;

double interestRate;

static int nextAccountNumber;

public:

bankAccount(string n, string t, double b, double r)

{

name = n;

accountNumber = nextAccountNumber++;

accountType = t;

balance = b;

interestRate = r;

}

// setter functions

void setInfo(string n, string t, double b, double r)

{

name = n;

accountType = t;

balance = b;

interestRate = r;

}

// getter functions

string getName()

{

return name;

}

int getAccountNumber()

{

return accountNumber;

}

string getAccountType()

{

return accountType;

}

double getBalance()

{

return balance;

}

double getInterestRate()

{

return interestRate;

}

void deposit(double amount)

{

balance = balance + amount;

}

void withdraw(double amount)

{

balance = balance - amount;

}

double calculateInterest()

{

return balance \* interestRate;

}

void printAccountInfo()

{

cout << "Account Holder's Name: " << name << endl;

cout << "Account Number: " << accountNumber << endl;

cout << "Account Type: " << accountType << endl;

cout << "Balance: $" << balance << endl;

cout << "Interest Rate: " << interestRate \* 100 << "%" << endl;

}

};

int bankAccount::nextAccountNumber = 1;

int main()

{

bankAccount accounts[10] =

{

bankAccount("Ali", "checking", 1000.0, 0.01),

bankAccount("Ibrahim", "saving", 5000.0, 0.02),

bankAccount("Hello", "checking", 2000.0, 0.01),

bankAccount("Jerry", "saving", 8000.0, 0.03),

bankAccount("zafar supaari", "checking", 1500.0, 0.01),

bankAccount("sheroz bhai", "saving", 6000.0, 0.02),

bankAccount("sheroz sab", "checking", 3000.0, 0.01),

bankAccount("sheroz ali", "saving", 7000.0, 0.03),

bankAccount("Dr sheroz", "checking", 2500.0, 0.01),

bankAccount("Imran", "saving", 9000.0, 0.02)

};

cout << "before manipulation" << endl;

accounts[0].printAccountInfo();

cout << endl;

accounts[0].setInfo("Ali", "checking", 20000.0, 0.01);

cout << "after manipulation" << endl;

accounts[0].printAccountInfo();

cout << endl;

for (int i = 0; i < 10; i++)

{

for (int i = 0; i < 10; i++)

{

cout << "Account " << i + 1 << ":" << endl;

accounts[i].printAccountInfo();

cout << endl;

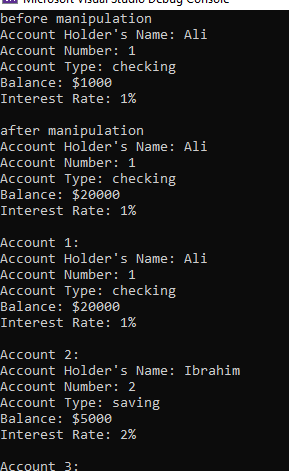
}

cout << endl;

return 0;

}

}



/\*Task-05 : \*/

#include <iostream>

using namespace std;

class Matrix

{

private:

int\*\* matrix;

int rows, cols;

public:

Matrix(int rows, int cols)

{

this->rows = rows;

this->cols = cols;

matrix = new int\* [rows];

for (int i = 0; i < rows; i++)

{

matrix[i] = new int[cols];

}

}

~Matrix()

{

for (int i = 0; i < rows; i++)

{

delete[] matrix[i];

}

delete[] matrix;

}

void setMatrix(int\*\* arr)

{

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

matrix[i][j] = arr[i][j];

}

}

}

int\*\* getMatrix() const

{

return matrix;

}

int getRows() const

{

return rows;

}

int getCols() const

{

return cols;

}

void printMatrix() const

{

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cout << matrix[i][j] << " ";

}

cout << endl;

}

}

};

int main()

{

int rows, cols;

cout << "Enter number of rows: ";

cin >> rows;

cout << "Enter number of columns: ";

cin >> cols;

Matrix matrix(rows, cols);

int\*\* arr = new int\* [rows];

for (int i = 0; i < rows; i++)

{

arr[i] = new int[cols];

}

cout << "Enter elements of matrix: " << endl;

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

cin >> arr[i][j];

}

}

matrix.setMatrix(arr);

cout << "The matrix is:" << endl;

matrix.printMatrix();

for (int i = 0; i < rows; i++)

{

delete[] arr[i];

}

delete[] arr;

return 0;

}

