

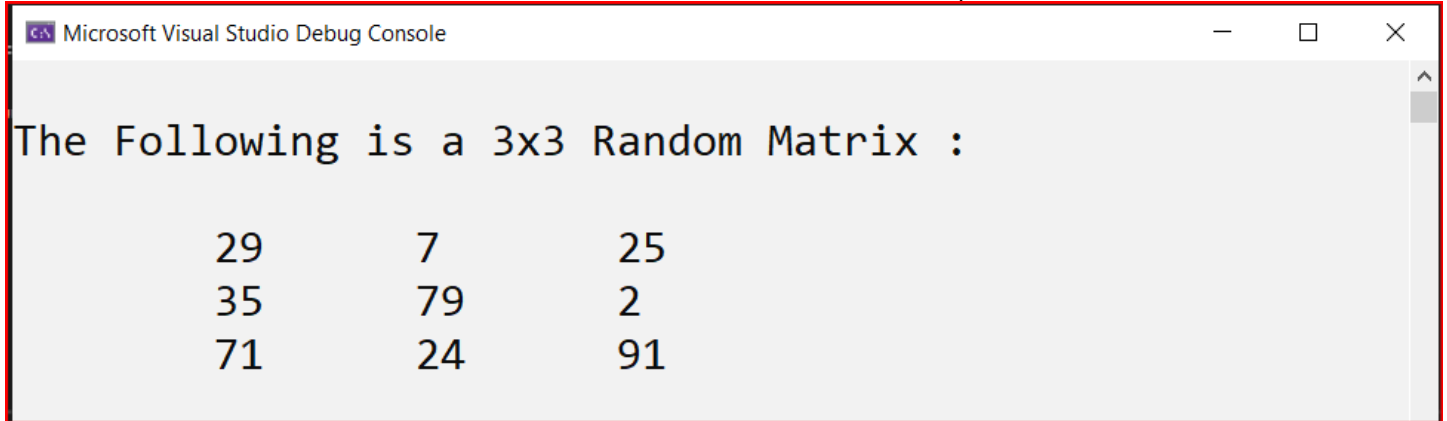
7 - Algorithms & Problem Solving Level 3

Mohammed Abu-Hadhoud

Problem #01 - 3x3 Random Matrix

Write a program to fill a 3x3 matrix with random numbers.

اكتب برنامجًا لملء مصفوفة 3x3 بأرقام عشوائية.



```
Microsoft Visual Studio Debug Console

The Following is a 3x3 Random Matrix :

    29    7    25
    35   79    2
    71   24   91
```

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

void FillElementsInArray(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Arr[Row][Col] = RandomNumber(1, 100);
        }
    }
}

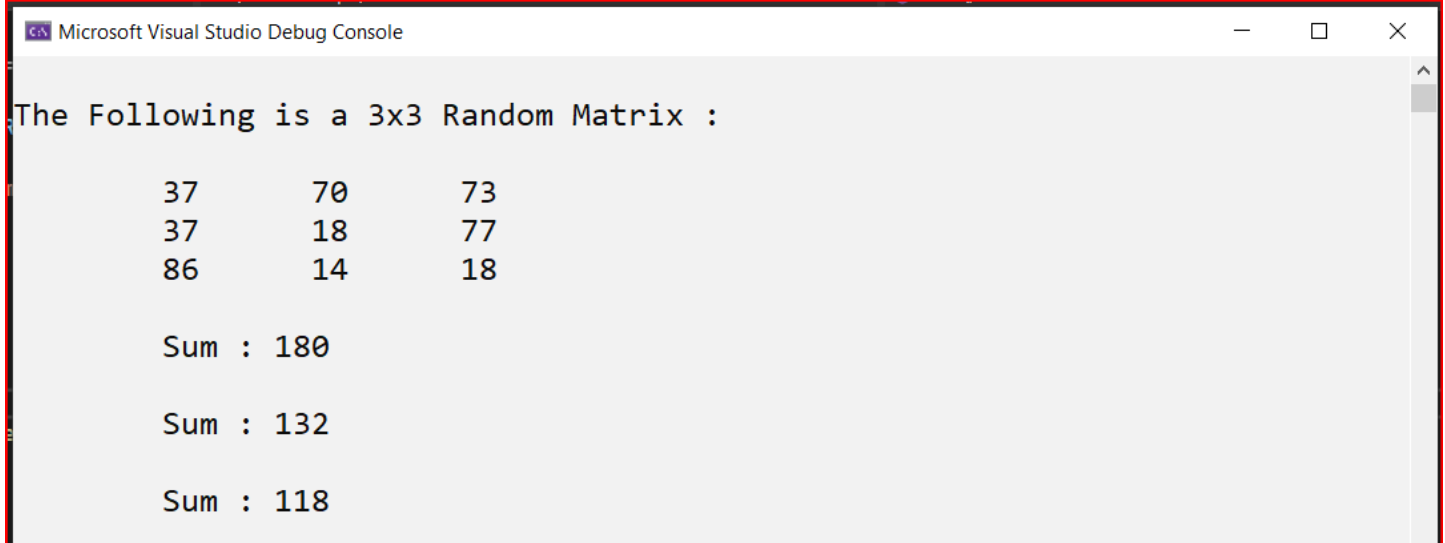
void PrintElementsFromArray(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            cout << "\t" << left << setw(3) << Arr[Row][Col] << "    ";
        }
        cout << "\n";
    }
}

int main()
{
    srand((unsigned)time(NULL));
    int Arr[3][3];
    FillElementsInArray(Arr, 3, 3);
    cout << "\n\nThe Following is a 3x3 Random Matrix : \n" << endl;
    PrintElementsFromArray(Arr, 3, 3);
    return 0;
}
```

Problem #02 - Sum Each Row in Matrix:

Write a program to fill a 3x3 matrix with random numbers, then print each row sum.

اكتب برنامجًا لملء مصفوفة 3x3 بأرقام عشوائية، ثم أطيح مجموع كل صف.



```
Microsoft Visual Studio Debug Console

The Following is a 3x3 Random Matrix :

    37      70      73
    37      18      77
    86      14      18

Sum : 180

Sum : 132

Sum : 118
```

Solving:

```
#include<iostream>
#include<iomanip>
```

```
using namespace std;
```

```
short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}
```

```
void FillElementsInArray(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Arr[Row][Col] = RandomNumber(1, 100);
        }
    }
}
```

```
void PrintElementsFromArray(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            cout << "\t" << left << setw(3) << Arr[Row][Col] << "  ";
        }
        cout << "\n";
    }
}
```

```

void PrintSumOfRows(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        int Sum = 0;

        for (short Col = 0; Col < Cols; Col++)
        {
            Sum += Arr[Row][Col];
        }
        cout << "\n\tSum : " << Sum << endl;
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Arr[3][3];
    FillElementsInArray(Arr, 3, 3);
    cout << "\nThe Following is a 3x3 Random Matrix : \n" << endl;
    PrintElementsFromArray(Arr, 3, 3);
    PrintSumOfRows(Arr, 3, 3);

    return 0;
}

```

=====

Problem #03: Sum Each Row in Matrix in Array

Write a program to fill a 3x3 matrix with random numbers, then Sum each row in separate array and print the results.

اكتب برنامجًا لملء مصفوفة 3x3 بأرقام عشوائية، ثم قم بجمع كل صف في مصفوفة منفصلة وطباعة النتائج.

```

The following is a 3x3 random matrix:

    6      43      33
   47      53      69
   11      10      18

The following are the sum of each row in the matrix:
Row 1 Sum = 82
Row 2 Sum = 169
Row 3 Sum = 39

```

Solving:

```

#include<iostream>
#include<iomanip>

using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + 1;
    return RandNum;
}

```

```

void FillMatrixWithRandomNumbers(intArr[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Arr[Row][Col] = RandomNumber(1, 100);
        }
    }
}

void PrintElementsOFMatrix(intArr[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            cout <<"\t"<< left << setw(3) <<Arr[Row][Col] <<" ";
        }
        cout <<"\n";
    }
}

int RowSum(intArr[3][3], shortRowNumber, shortCols)
{
    int Sum = 0;

    for (short Col = 0; Col <Cols; Col++)
    {
        Sum += Arr[RowNumber][Col];
    }
    return Sum;
}

void PrintMatrix(intArr[3][3], shortRows, shortCols)
{
    cout <<"\n";

    for (short Row = 0; Row <Rows; Row++)
    {
        for(short Col = 0; Col <Cols; Col++)
        {
            cout <<"\t"<< setw(3) <<Arr[Row][Col] <<" ";
        }
        cout << endl;
    }
}

void SumMatrixRowsInArray(intArr[3][3], intArrSum[], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        ArrSum[Row] = RowSum(Arr, Row, Cols);
    }
}

```

```

void PrintRowsSumArray(intArr[], shortRows)
{
    cout << "\nThe following are the sum of each row in the matrix:\n";
    for (short Row = 0; Row < Rows; Row++)
    {
        cout << "\tRow " << Row + 1 << " Sum = " << Arr[Row] << endl;
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Arr[3][3];
    int ArrSum[3];

    FillMatrixWithRandomNumbers(Arr, 3, 3);
    cout << "\nThe following is a 3x3 random matrix:\n";
    PrintMatrix(Arr, 3, 3);
    SumMatrixRowsInArray(Arr, ArrSum, 3, 3);
    PrintRowsSumArray(ArrSum, 3);

    system("pause>0");
    return 0;
}

```

Problem #04: Sum Each Col in Matrix

Write a program to fill a 3x3 Matrix with random numbers, then print each Col Sum.

اكتب برنامجًا لملء مصفوفة 3x3 بأرقام عشوائية، ثم أطيح مجموع كل عمود.

The following is a 3x3 random matrix:

29	17	52
1	89	76
86	49	52

The following are the sum of each row in the matrix:

Col 1 Sum = 116
 Col 2 Sum = 155
 Col 3 Sum = 180

Solving:

```

#include<iostream>
#include<iomanip>

```

```

usingnamespace std;

```

```

short RandomNumber(shortfrom, shortto)
{
    short RandNum = rand() % (to - from + 1) + 1;
    return RandNum;
}

```

```

void FillMatrixWithRandomNumbers(intArr[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Arr[Row][Col] = RandomNumber(1, 100);
        }
    }
}

int ColSum(intArr[3][3], shortRows, shortColNumber)
{
    int Sum = 0;

    for (short Row = 0; Row <Rows; Row++)
    {
        Sum += Arr[Row][ColNumber];
    }
    return Sum;
}

void PrintMatrix(intArr[3][3], shortRows, shortCols)
{
    cout <<"\n";
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            cout <<"\t"<< setw(3) <<Arr[Row][Col] <<"    ";
        }
        cout << endl;
    }
}

void PrintEachColSum(intArr[3][3], shortRows, shortCols)
{
    cout <<"\nThe following are the sum of each row in the matrix:\n";
    for (short Col = 0; Col <Cols; Col++)
    {
        cout <<"\tCol " << Col + 1 <<" Sum = " << ColSum(Arr, Rows, Col) << endl;
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Arr[3][3];
    int ArrSum[3];
    FillMatrixWithRandomNumbers(Arr, 3, 3);
    cout <<"\nThe following is a 3x3 random matrix:\n";
    PrintMatrix(Arr, 3, 3);
    PrintEachColSum(Arr, 3,3);
    system("pause>0");
    return 0;
}

```

Problem #05: Sum Each Col in Matrix in another Array

Write a program to fill a 3x3 Matrix with random numbers, then sum each Column in another array and print them.

اكتب برنامجًا لملء مصفوفة 3x3 بأرقام عشوائية، ثم جمع كل عمود في مصفوفة أخرى وطباعتها.

95	16	14
18	34	26
90	71	42

Element[0] Of Array is : 203
Element[1] Of Array is : 121
Element[2] Of Array is : 82

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

void FillMatrixWithRandomNumbers(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Arr[Row][Col] = RandomNumber(1, 100);
        }
    }
}

void PrintMatrix(int Arr[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            cout << "\t" << setw(3) << Arr[Row][Col] << " ";
        }
        cout << endl;
    }
}

short SumColumns(int Arr[3][3], short ColNumber, short Cols)
{
    short Sum = 0;
    for (short Col = 0; Col < Cols; Col++)
    {
        Sum += Arr[Col][ColNumber];
    }
    return Sum;
}
```



```

void PrintSumColumns(intArr[3][3], shortCols)
{
    for (short Col = 0; Col <Cols; Col++)
    {
        cout <<"\n\tSum Of Column ["<< Col <<"] is : "
        << SumColumns(Arr, Col, Cols);
    }
    cout << endl;
}

void SumMatrixColsInArray(intArr[3][3], intArrSum[], shortRows, shortCols)
{
    for (short i = 0; i <Rows; i++)
    {
        ArrSum[i] = SumColumns(Arr, i, Cols);
    }
}

void PrintColsSumArray(intArrSum[], shortlength)
{
    for (short i = 0; i <length; i++)
    {
        cout <<"\n\tElement["<< i <<"] Of Array is : "<<ArrSum[i];
    }
    cout << endl;
}

int main()
{
    srand((unsigned)time(NULL));

    int Arr[3][3];
    int ArrSum[3];

    FillMatrixWithRandomNumbers(Arr, 3, 3);

    PrintMatrix(Arr, 3, 3);

    SumMatrixColsInArray(Arr, ArrSum, 3, 3);

    PrintColsSumArray(ArrSum, 3);

    system("pause>0");
}

```

Problem #06: 3x3 Ordered Matrix

Write a program to fill a 3x3 Matrix with ordered numbers.

اكتب برنامجًا لملء مصفوفة 3x3 بالأرقام المرتبة.

The following is a 3x3 ordered matrix:

1	2	3
4	5	6
7	8	9

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;
```

```
void FillMatrixWithOrderedNumbers(int Arr[3][3], short Rows, short Cols)
{
    short C = 0;

    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            C++;

            Arr[Row][Col] = C;
        }
    }
}
```

```
void PrintMatrix(int Arr[3][3], short Rows, short Cols)
{
    cout << endl;
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            cout << "\t" << setw(3) << Arr[Row][Col] << " ";
        }
        cout << endl;
    }
}
```

```
int main()
{
    srand((unsigned)time(NULL));

    int Arr[3][3];
    FillMatrixWithOrderedNumbers(Arr, 3, 3);
    cout << "\n\tThe following is a 3x3 ordered matrix:\n";
    PrintMatrix(Arr, 3, 3);

    system("pause>0");
}
```

Problem #07: Transpose Matrix

Write a program to fill a 3x3 Matrix with ordered numbers and print it, then transpose matrix and print it.

اكتب برنامجًا لملء مصفوفة 3x3 بالأرقام المرتبة وطباعتها، ثم تبديل موضع المصفوفة وطباعتها.

The Following is a 3x3 Ordered Matrix:

1	2	3
4	5	6
7	8	9

The Following is the Transposed Matrix:

1	4	7
2	5	8
3	6	9

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;
```

```
void FillMatrixWithOrderedNumbers(intArr[3][3], shortRows, shortCols)
{
    short C = 0;

    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            C++;
            Arr[Row][Col] = C;
        }
    }
}
```

```
void PrintMatrix(intArr[3][3], shortRows, shortCols)
{
    cout << "\n\t";

    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            cout << Arr[Row][Col] << " ";
        }
        cout << "\n\t";
    }
}
```

```

void TransposeMatrix(intArr[3][3], intArrTransposed[3][3], shortRows,
shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            ArrTransposed[Col][Row] = Arr[Row][Col];
        }
    }
}

int main()
{
    int Arr[3][3], ArrTransposed[3][3];

    FillMatrixWithOrderedNumbers(Arr, 3, 3);

    cout <<"\n\tThe Following is a 3x3 Ordered Matrix: \n";
    PrintMatrix(Arr, 3, 3);

    cout <<"\n\t-----\n";

    TransposeMatrix(Arr, ArrTransposed, 3, 3);
    cout <<"\n\tThe Following is the Transposed Matrix: \n";
    PrintMatrix(ArrTransposed, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #08: Multiply Two Matrices

Write a program to fill two 3x3 Matrix with random numbers and print them, then multiply elements them and fill them result into a 3rd matrix and print it.

اكتب برنامجاً لملء مصفوفتين 3x3 بأرقام عشوائية وطباعتهما ثم ضرب عناصرهما وتعبئة الناتج في مصفوفة ثالثة وطباعتها.

Matrix 1:

04	03	10
07	09	04
08	05	08

=====

Matrix 2:

05	03	04
02	05	06
05	07	07

=====

Result :

20	09	40
14	45	24
40	35	56

Solving:

```
#include<iostream>
```

```
#include<iomanip>
```

```
using namespace std;
```

```
short RandomNumber(short from, short to)
```

```
{  
    short RandNum = rand() % (to - from + 1) + from;  
    return RandNum;  
}
```

```
void FillMatrixWithRandomNumbers(int Matrix1[3][3], short Rows, short Cols)
```

```
{  
    for (short Row = 0; Row < Rows; Row++)  
    {  
        for (short Col = 0; Col < Cols; Col++)  
        {  
            Matrix1[Row][Col] = RandomNumber(1, 10);  
        }  
    }  
}
```

```

void PrintMatrix(intMatrix1[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            printf("\t%02d", Matrix1[Row][Col]);
        }
        cout << endl;
    }
}

void MultiplyMatrix(intMatrix1[3][3], intMatrix2[3][3], intMatrixResult[3][3],
shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            MatrixResult[Row][Col] = Matrix1[Row][Col] * Matrix2[Row][Col];
        }
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix1[3][3], Matrix2[3][3], MatrixResult[3][3];
    cout <<"\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix1, 3, 3);
    PrintMatrix(Matrix1, 3, 3);

    cout <<"\n\t=====\n";
    cout <<"\n\tMatrix 2: \n\n";
    FillMatrixWithRandomNumbers(Matrix2, 3, 3);
    PrintMatrix(Matrix2, 3, 3);

    cout <<"\n\t=====\n";
    cout <<"\n\tResult : \n\n";
    MultiplyMatrix(Matrix1, Matrix2, MatrixResult, 3, 3);
    PrintMatrix(MatrixResult, 3, 3);

    system("pause>0");
}

```

Problem #09: Print Middle Row and Col of Matrix

Write a program to fill a 3x3 Matrix with random numbers and print it, then print the middle row and middle col.

اكتب برنامجاً لملء مصفوفة 3x3 بأرقام عشوائية ثم قم بطباعتها، ثم أطبع الصف الأوسط والعمود الأوسط.

Matrix 1:

09	02	08
04	10	09
03	09	07

=====

Middle Row:

04	10	09
----	----	----

=====

Middle Col:

02	10	09
----	----	----

Solving:

```
#include<iostream>
#include<iomanip>
```

```
using namespace std;
```

```
short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}
```

```
void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 10);
        }
    }
}
```

```

void PrintMatrix(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            printf("\t%02d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}

void PrintMiddleRowOfMatrix(intMatrix[3][3], shortRows, shortCols)
{
    short MidlleRows = Rows / 2;

    for (short Col = 0; Col <Cols; Col++)
    {
        printf("\t%02d", Matrix[MidlleRows][Col]);
    }
}

void PrintMiddleColOfMatrix(intMatrix[3][3], shortRows, shortCols)
{
    short MidlleCols = Cols / 2;

    for (short Row = 0; Row <Rows; Row++)
    {
        printf("\t%02d", Matrix[Row][MidlleCols]);
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix[3][3];
    cout <<"\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix, 3, 3);
    PrintMatrix(Matrix, 3, 3);

    cout <<"\n\t===== \n\n";
    cout <<"\tMiddle Row: \n\n";

    PrintMiddleRowOfMatrix(Matrix, 3, 3);

    cout <<"\n\n\t===== \n\n";
    cout <<"\tMiddle Col: \n\n";

    PrintMiddleColOfMatrix(Matrix, 3, 3);

    cout << endl;
    system("pause>0");
}

```


Problem #10: Sum of Matrix

Write a program to fill a 3x3 Matrix with random numbers and print it, then write a function to sum all numbers in this Matrix and print it.

اكتب برنامجًا لملء مصفوفة 3x3 بأرقام عشوائية ثم قم بطباعتها، ثم اكتب دالة لجمع كل الأرقام الموجودة في هذه المصفوفة و قم بطباعتها.

Matrix 1:

10	08	05
01	10	06
06	02	07

=====

Sum Numbers Of Matrix: 55

=====

Matrix 2:

05	04	05
09	09	05
04	05	03

=====

Sum Numbers Of Matrix: 49

Solving:

```
#include<iostream>
```

```
#include<iomanip>
```

```
using namespace std;
```

```
short RandomNumber(short from, short to)
```

```
{
```

```
    short RandNum = rand() % (to - from + 1) + from;
```

```
    return RandNum;
```

```
}
```

```
void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)
```

```
{
```

```
    for (short Row = 0; Row < Rows; Row++)
```

```
    {
```

```
        for (short Col = 0; Col < Cols; Col++)
```

```
        {
```

```
            Matrix[Row][Col] = RandomNumber(1, 10);
```

```
        }
```

```
    }
```

```
}
```

```

void PrintMatrix(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            printf("\t%02d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}

int SumNumbersOfMatrix(intMatrix[3][3], shortRows, shortCols)
{
    short Sum = 0;

    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Sum += Matrix[Row][Col];
        }
    }
    return Sum;
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix1[3][3], Matrix2[3][3];

    cout <<"\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix1, 3, 3);
    PrintMatrix(Matrix1, 3, 3);

    cout <<"\n\t===== \n\n";

    cout <<"\tSum Numbers Of Matrix: " << SumNumbersOfMatrix(Matrix1, 3, 3) << endl;

    cout <<"\n\t===== \n\n";

    cout <<"\n\tMatrix 2: \n\n";
    FillMatrixWithRandomNumbers(Matrix2, 3, 3);
    PrintMatrix(Matrix2, 3, 3);

    cout <<"\n\t===== \n\n";

    cout <<"\tSum Numbers Of Matrix: " << SumNumbersOfMatrix(Matrix2, 3, 3) << endl;

    cout << endl;
    system("pause>0");
}

```

Problem #11: Check Matrices Equality

Write a program to fill two 3x3 Matrices with random numbers and print them, then write a function to sum all numbers in those two matrices and compare sum of them to check if they are equal or not.

اكتب برنامجًا لملء مصفوفتين 3x3 بأرقام عشوائية وطباعتهما، ثم اكتب دالة لجمع كل الأرقام في هاتين المصفوفتين ومقارنة مجموعهما لمعرفة ما إذا كانت متساوية أم لا.

```
E:\Courses\Programming Advices Courses_M. AbuHadhoud\ملفات مشاريع\Visual Studio 2022\repos\Test1\x64\Debug\...

Matrix 1:

03      07      07
08      02      01
09      08      04

=====

Matrix 2:

04      07      05
05      03      07
07      06      09

=====

No : Matrices are NOT Equal.
```

Solving:

```
#include<iostream>
#include<iomanip>
```

```
using namespace std;
```

```
short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}
```

```
void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 10);
        }
    }
}
```

```

void PrintMatrix(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            printf("\t%02d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}

int SumNumbersOfMatrix(intMatrix1[3][3], shortRows, shortCols)
{
    short Sum = 0;

    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Sum += Matrix1[Row][Col];
        }
    }
    return Sum;
}

bool AreMatricesEqual(intMatrix1[3][3], intMatrix2[3][3], shortRows, shortCols)
{
    return SumNumbersOfMatrix(Matrix1, Rows, Cols) == SumNumbersOfMatrix(Matrix2, Rows, Cols);
}

void PrintResult(intMatrix1[3][3], intMatrix2[3][3], shortRows, shortCols)
{
    if (AreMatricesEqual(Matrix1, Matrix2, Rows, Cols))
    {
        cout << "\n\tYes: Both Matrices are Equal.\n";
    }
    else
    {
        cout << "\n\tNo : Matrices are NOT Equal.\n";
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix1[3][3], Matrix2[3][3];
    cout << "\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix1, 3, 3);
    PrintMatrix(Matrix1, 3, 3);

    cout << "\n\t===== \n";
    cout << "\n\tMatrix 2: \n\n";
    FillMatrixWithRandomNumbers(Matrix2, 3, 3);
    PrintMatrix(Matrix2, 3, 3);

    cout << "\n\t===== \n";
    PrintResult(Matrix1, Matrix2, 3, 3);
    cout << endl;
    system("pause>0");
}

```

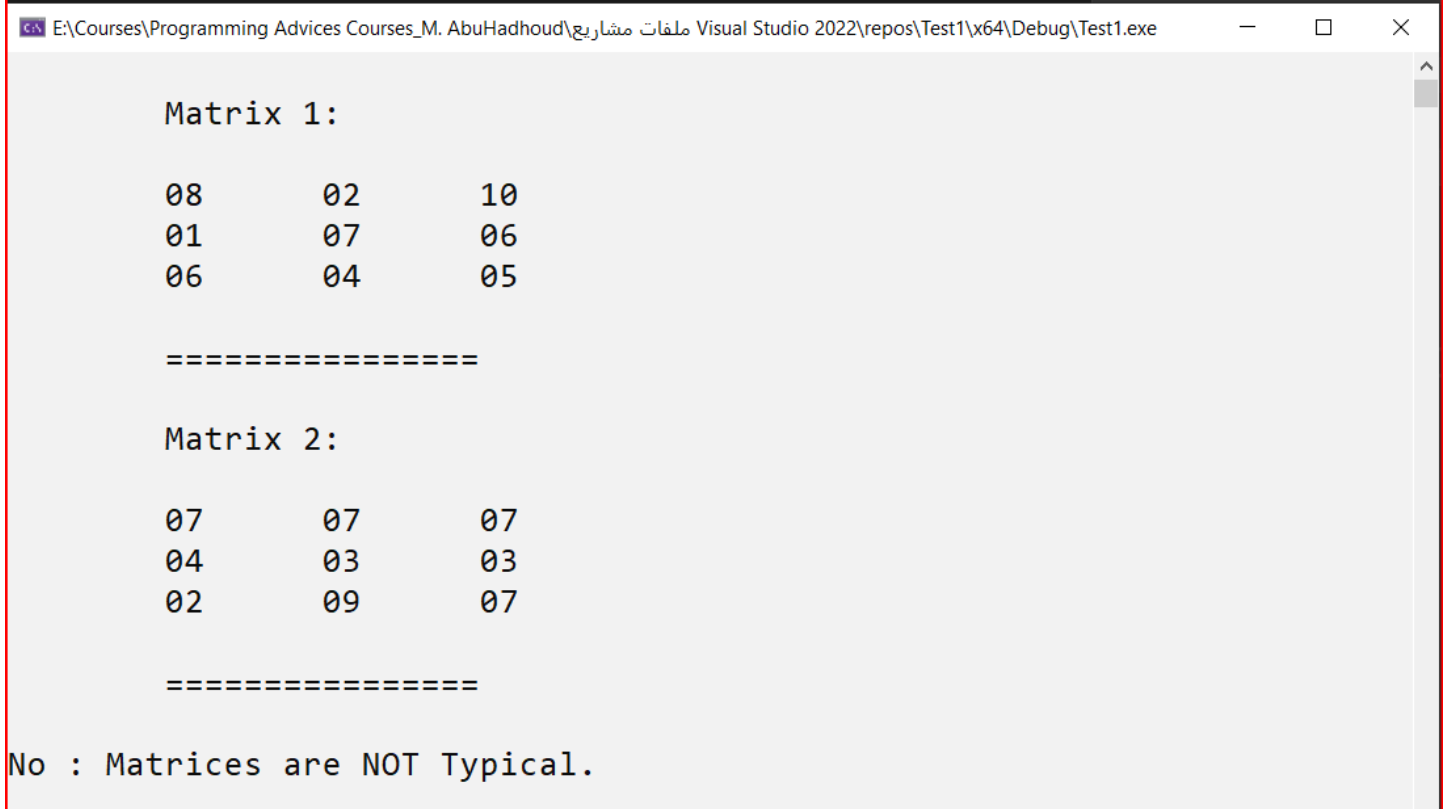
Problem #12: Check Typical Matrices

Write a program to fill two 3x3 Matrices with random numbers and print them, then check if they are Typical or not.

Yes: Matrices are Typical.

No : Matrices are NOT Typical.

اكتب برنامجًا لملء مصفوفتين 3x3 بأرقام عشوائية ثم قم بطباعتهما ثم تحقق مما إذا كانتا نموذجيتين أم لا.



```
Matrix 1:

08      02      10
01      07      06
06      04      05

=====

Matrix 2:

07      07      07
04      03      03
02      09      07

=====

No : Matrices are NOT Typical.
```

Solving:

```
#include<iostream>
#include<iomanip>
```

```
usingnamespace std;
```

```
short RandomNumber(shortfrom, shortto)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}
```

```
void FillMatrixWithRandomNumbers(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 10);
        }
    }
}
```

```

void PrintMatrix(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            printf("\t%02d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}

bool CheckTypicalMatrices(intMatrix1[3][3], intMatrix2[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            if (Matrix1[Row][Col] != Matrix2[Row][Col])
            {
                return 0;
            }
        }
    }
    return 1;
}

void PrintResultOfCheck(intMatrix1[3][3], intMatrix2[3][3], shortRows, shortCols)
{
    if (CheckTypicalMatrices(Matrix1, Matrix2, Rows, Cols))
        cout << "\t\nYes: Matrices are Typical.\n";

    else
        cout << "\t\nNo : Matrices are NOT Typical.\n";
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix1[3][3], Matrix2[3][3];

    cout << "\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix1, 3, 3);
    PrintMatrix(Matrix1, 3, 3);
    cout << "\n\t===== \n";
    cout << "\n\tMatrix 2: \n\n";
    FillMatrixWithRandomNumbers(Matrix2, 3, 3);
    PrintMatrix(Matrix2, 3, 3);
    cout << "\n\t===== \n";

    PrintResultOfCheck(Matrix1, Matrix2, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #13: Check Identity Matrix

Write a program to check if the matrix is identity or not.

اكتب برنامجًا للتحقق مما إذا كانت المصفوفة 3×3 هي متجانسة (متماثلة) أم لا.

Identity (متجانس او متماثل) يجب أن يكون الdiagonal (قطري) عبارة عن رقم 1 وباقي الأرقام 0 مثل:

```
1  0  0
0  1  0
0  0  1
```

```
E:\Courses\Programming Advices Courses_M. AbuHadhoud\مشاريع Visual St...
Matrix :

1      2      3
4      5      6
7      8      9

=====

No, this matrix is Not Identity.
```

```
E:\Courses\Programming Advices Courses_M. AbuHadhoud\مشاريع Visual Stur
Matrix :

1      0      0
0      1      0
0      0      1

=====

Yes, this matrix is Identity.
```

Solving:

```
#include<iostream>
#include<iomanip>
```

```
using namespace std;
```

```
void PrintMatrix(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            printf("\t%d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}
```

```
bool IsIdentityMatrix(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            if ((Row == Col) && Matrix[Row][Col] != 1)
            {
                return false;
            }
            elseif ((Row != Col) && Matrix[Row][Col] != 0)
            {
                return false;
            }
        }
    }
    return true;
}
```

```

void PrintResult(intMatrix[3][3], shortRows, shortCols)
{
    if (IsIdentityMatrix(Matrix, Rows, Cols))
        cout <<"\n\tYes, this matrix is Identity.";
    else
        cout <<"\n\tNo, this matrix is Not Identity.";
}

int main()
{
    int Matrix[3][3] = { {1,2,3},{4,5,6},{7,8,9} };
    //int Matrix[3][3] = { {1,0,0},{0,1,0},{0,0,1} };
    cout <<"\n\tMatrix : \n\n";

    PrintMatrix(Matrix, 3, 3);
    cout <<"\n\t=====\\n";

    PrintResult(Matrix, 3, 3);
    cout << endl;
    system("pause>0");
}

```

Problem #14: Check Scalar Matrix

Write a program to check if the matrix is Scalar or not..

اكتب برنامجًا للتحقق مما إذا كانت المصفوفة عددية أم لا.

الـ Scalar (عددية) يجب أن يكون الـ diagonal (قطري) متساوي بغض النظر عن الرقم الموجود فيه وباقي الأرقام 0

Matrix :

6	0	0
0	6	0
0	0	6

=====

Yes, this matrix is Scalar.

Solving:

```

#include<iostream>
#include<iomanip>

```

```

usingnamespace std;

```

```

void PrintMatrix(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            printf("\t%d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}

```



```

bool IsScalarMatrix(intMatrix[3][3], shortRows, shortCols)
{
    int FirstElementInMatrix = Matrix[0][0];

    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            if ((Row == Col) &&Matrix[Row][Col] != FirstElementInMatrix)
            {
                returnfalse;
            }
            elseif ((Row != Col) &&Matrix[Row][Col] != 0)
            {
                returnfalse;
            }
        }
    }
    returntrue;
}

void PrintResult(intMatrix[3][3], shortRows, shortCols)
{
    if (IsScalarMatrix(Matrix, Rows, Cols))
        cout <<"\n\tYes, this matrix is Scalar.";
    else
        cout <<"\n\tNo, this matrix is Not Scalar.";
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix[3][3] = { {6,0,0},{0,6,0},{0,0,6} };

    cout <<"\n\tMatrix : \n\n";
    PrintMatrix(Matrix, 3, 3);

    cout <<"\n\t=====\\n";

    PrintResult(Matrix, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #15: Count Number in Matrix

Write a program to count given number in matrix.

اكتب برنامجًا لحساب كم مرة يظهر الرقم الذي ادخله المستخدم من داخل المصفوفة .
مثل: المستخدم أدخل رقم 2 .. حساب كم عدد الرقم 2 الذي ادخله المستخدم داخل المصفوفة:

```
Matrix 1:

    20      01      13
    16      02      19
    15      20      06

=====
Enter the number to count in matrix: 2

Number 2 count in Matrix is : 1
```

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

short ReadNumber()
{
    short Num;
    cout << "Enter the number to count in matrix: ";
    cin >> Num;
    return Num;
}

void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 20);
        }
    }
}

void PrintMatrix(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            printf("\t%02d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}
```

```

short CountNumberInMatrix(intMatrix[3][3], shortNum, shortRows, shortCols)
{
    short Sum = 0;

    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            if (Matrix[Row][Col] == Num)
            {
                Sum++;
            }
        }
    }
    return Sum;
}

```

```

int main()
{
    srand((unsigned)time(NULL));

    int Matrix[3][3];

    cout <<"\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix, 3, 3);
    PrintMatrix(Matrix, 3, 3);

    cout <<"\n\t=====\\n";

    short Num = ReadNumber();
    cout <<"\t\\nNumber " << Num <<" count in Matrix is : "
    << CountNumberInMatrix(Matrix, Num, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #16: Check Sparse Matrix

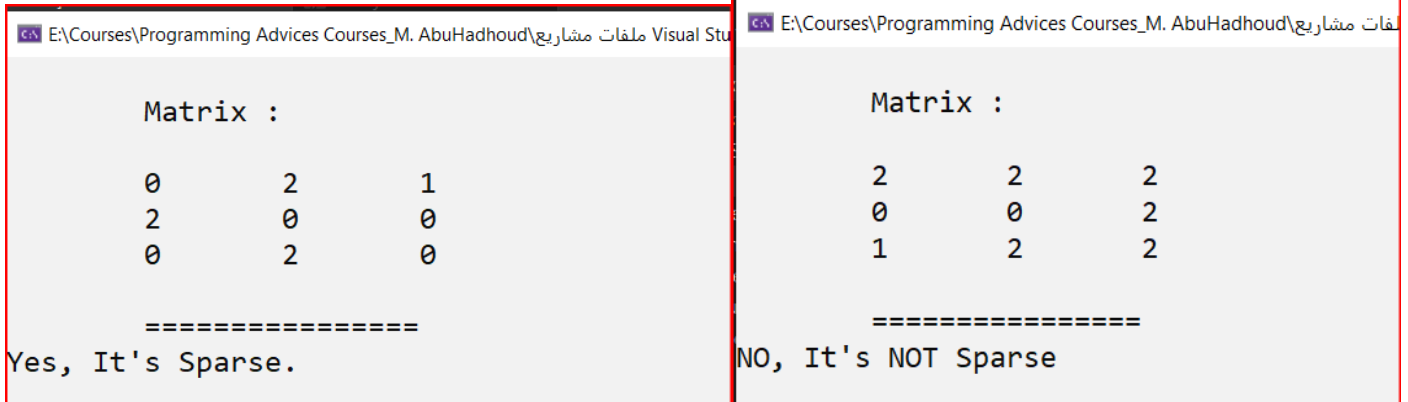
Write a program to check if the matrix is **Sparse** or not..

اكتب برنامجًا للتحقق مما إذا كانت المصفوفة متفرقة أم لا.

"matrix is sparse" تشير إلى مصفوفة تحتوي على عدد كبير من العناصر الفارغة (zero) مقارنةً بعناصرها المملوءة.

The Matrix is **Sparse**

The Matrix is **NOT Sparse**



Solving:

```
#include<iostream>  
#include<iomanip>  
  
using namespace std;  
  
short RandomNumber(short from, short to)  
{  
    short RandNum = rand() % (to - from + 1) + from;  
    return RandNum;  
}  
  
void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)  
{  
    for (short Row = 0; Row < Rows; Row++)  
    {  
        for (short Col = 0; Col < Cols; Col++)  
        {  
            Matrix[Row][Col] = RandomNumber(0, 3);  
        }  
    }  
}  
  
void PrintMatrix(int Matrix[3][3], short Rows, short Cols)  
{  
    for (short Row = 0; Row < Rows; Row++)  
    {  
        for (short Col = 0; Col < Cols; Col++)  
        {  
            printf("\t%d", Matrix[Row][Col]);  
        }  
        cout << endl;  
    }  
}
```

```

short CountNumberInMatrix(intMatrix[3][3], shortNum, shortRows, shortCols)
{
    short Sum = 0;

    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            if (Matrix[Row][Col] == Num)
            {
                Sum++;
            }
        }
    }
    return Sum;
}

bool IsSparseMatrix(intMatrix[3][3], shortRows, shortCols)
{
    short MatrixSize = Rows * Cols;

    return CountNumberInMatrix(Matrix, 0, Rows, Cols) > MatrixSize / 2;
}

void PrintResult(intMatrix[3][3], shortRows, shortCols)
{
    if (IsSparseMatrix(Matrix, Rows, Cols))
        cout <<"Yes, It's Sparse.";
    else
        cout <<"NO, It's NOT Sparse";
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix[3][3];

    //int Matrix[3][3] = {{0,2,2}, {4,0,0},{1,9,0}};

    cout <<"\n\tMatrix : \n\n";

    FillMatrixWithRandomNumbers(Matrix, 3, 3);
    PrintMatrix(Matrix, 3, 3);

    cout <<"\n\t===== \n";

    PrintResult(Matrix, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #17: Number Exists in Matrix

Write a program to check if a given number exists in matrix or not.

كتابة برنامج للتحقق من وجود رقم معين في المصفوفة أم لا.

Matrix :

32	46	43
72	28	62
87	26	45

=====

Please enter the number to look for in matrix ? 62

Yes, It's There.

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

short ReadNumber()
{
    short Number;
    cout << "\nPlease enter the number to look for in matrix ? ";
    cin >> Number;
    return Number;
}

void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 100);
        }
    }
}

void PrintMatrix(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            printf("\t%d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}
```

```

bool IsNumberInMatrix(intMatrix[3][3], shortNum, shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            if (Matrix[Row][Col] == Num)
            {
                return true;
            }
        }
    }
    return false;
}

void PrintResult(intMatrix[3][3], shortNum, shortRows, shortCols)
{
    if (IsNumberInMatrix(Matrix, Num, Rows, Cols))
        cout <<"Yes, It's There.";
    else
        cout <<"NO, It's NOT There";
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix[3][3];

    //int Matrix[3][3] = {{0,2,2}, {4,0,0},{1,9,0}};

    cout <<"\n\tMatrix : \n\n";

    FillMatrixWithRandomNumbers(Matrix, 3, 3);
    PrintMatrix(Matrix, 3, 3);

    cout <<"\n\t===== \n";
    short Number = ReadNumber();

    PrintResult(Matrix, Number, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #18: Intersected Number in Matrices

Write a program to print the intersected numbers in two given matrices.

اكتب برنامجاً لطباعة الأعداد المتقاطعة (المشتركة) في مصفوفتين معطيتين.

```
Matrix 1:

6      9      7
5      3     10
10     6      1

=====

Matrix 2:

1      2      7
10     10     10
4      1      5

The Intersected Numbers are : 7  5  10  10  1
```

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

short ReadNumber()
{
    short Number;
    cout << "\nPlease enter the number to look for in matrix ? ";
    cin >> Number;
    return Number;
}

void FillMatrixWithRandomNumbers(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 10);
        }
    }
}

void PrintMatrix(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            printf("\t%d", Matrix[Row][Col]);
        }
        cout << endl;
    }
}
```



```

bool IsNumberInMatrix(intMatrix1[3][3], shortNumber, shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            if (Matrix1[Row][Col] == Number)
                returntrue;
        }
    }
    returnfalse;
}

void FindIntersectedNumbers(intMatrix1[3][3], intMatrix2[3][3], shortRows,
shortCols)
{
    short Number = 0;

    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Number = Matrix1[Row][Col];

            if (IsNumberInMatrix(Matrix2, Number, Rows, Cols))
            {
                cout << Number <<" ";
            }
        }
    }
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix1[3][3], Matrix2[3][3];
    //int Matrix[3][3] = {{0,2,2}, {4,0,0},{1,9,0}};

    cout <<"\n\tMatrix 1: \n\n";
    FillMatrixWithRandomNumbers(Matrix1, 3, 3);
    PrintMatrix(Matrix1, 3, 3);

    cout <<"\n\t===== \n";

    cout <<"\n\tMatrix 2: \n\n";
    FillMatrixWithRandomNumbers(Matrix2, 3, 3);
    PrintMatrix(Matrix2, 3, 3);

    cout << endl;
    cout <<"The Intersected Numbers are : ";
    FindIntersectedNumbers(Matrix1, Matrix2, 3, 3);
    //PrintResult(Matrix, Number, 3, 3);

    cout << endl;
    system("pause>0");
}

```

Problem #19: Min-Max Number in Matrix

Write a program to print the Minimum and Maximum Numbers in Matrix.

اكتب برنامجًا لطباعة الرقم الأدنى والرقم الأقصى في المصفوفة.

```
41  63  76
41  97  52
86  31  11
```

The Minimum Number In Matrix is : 11

The Maxmum Number In Matrix is : 97

Solving:

```
#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

void FillMatrixWithRandomNumber(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 100);
        }
    }
}

void PrintMatrix(int Matrix[3][3], short Rows, short Cols)
{
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            cout << setw(3) << Matrix[Row][Col] << " ";
        }
        cout << endl;
    }
}

int MinNumberInMatrix(int Matrix[3][3], short Rows, short Cols)
{
    short Min = Matrix[0][0];
    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            if (Matrix[Row][Col] < Min)
                Min = Matrix[Row][Col];
        }
    }
    return Min;
}
```

```

int MaxNumberInMatrix(int Matrix[3][3], short Rows, short Cols)
{
    short Max = Matrix[0][0];

    for (short Row = 0; Row < Rows; Row++)
    {
        for (short Col = 0; Col < Cols; Col++)
        {
            if (Matrix[Row][Col] > Max)
            {
                Max = Matrix[Row][Col];
            }
        }
    }
    return Max;
}

int main()
{
    srand((unsigned)time(NULL));

    int Matrix[3][3];

    FillMatrixWithRandomNumber(Matrix, 3, 3);
    PrintMatrix(Matrix, 3, 3);

    cout << "\n The Minimum Number In Matrix is : "
    << MinNumberInMatrix(Matrix, 3, 3) << endl;

    cout << "\n The Maxmum Number In Matrix is : "
    << MaxNumberInMatrix(Matrix, 3, 3);

    cout << "\n";
    system("pause>0");
}

```

Problem #20: Palindrome Matrix

Write a program to print the intersected numbers in two given matrices.

اكتب برنامجًا للتأكد من أن المصفوفة متناظرة أم لا.

3	8	8	3
8	3	3	8
2	9	3	2
No, Matrix is NOT Palindrome.			

3	8	8	3
8	3	3	8
2	9	9	2
Yes, Matrix is Palindrome.			

Solving:

```

#include<iostream>
#include<iomanip>
using namespace std;

short RandomNumber(short from, short to)
{
    short RandNum = rand() % (to - from + 1) + from;
    return RandNum;
}

```

```

void FillMatrixWithRandomNumber(intMatrix[3][3], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            Matrix[Row][Col] = RandomNumber(1, 100);
        }
    }
}

void PrintMatrix(intMatrix[3][4], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols; Col++)
        {
            cout << setw(3) <<Matrix[Row][Col] <<" ";
        }
        cout << endl;
    }
}

bool IsPalindromeMatrix(intMatrix[3][4], shortRows, shortCols)
{
    for (short Row = 0; Row <Rows; Row++)
    {
        for (short Col = 0; Col <Cols / 2; Col++)
        {
            if (Matrix[Row][Col] != Matrix[Row][Cols - 1 - Col])
                returnfalse;
        }
    }
    returntrue;
}

void PrintResult(intMatrix[3][4], shortRows, shortCols)
{
    if (IsPalindromeMatrix(Matrix, Rows, Cols))
    {
        cout <<"Yes, Matrix is Palindrome."<< endl;
    }
    else
        cout <<"No, Matrix is NOT Palindrome."<< endl;
}

int main()
{
    srand((unsigned)time(NULL));

    //int Matrix[3][3];
    int Matrix[3][4] = { {3,8,8,3},{8,3,3,8},{2,9,3,2} };
    //FillMatrixWithRandomNumber(Matrix, 3, 3);
    PrintMatrix(Matrix, 3, 4);
    PrintResult(Matrix, 3, 4);
    cout <<"\n";
    system("pause>0");
}

```

Problem #21: Fibonacci Series

Write a program to print Fibonacci Series of 10.

اكتب برنامجًا لطباعة سلسلة فيبوناتشي المكونة من 10.

ما هي متتالية فيبوناتشي؟

هي متتالية من الأرقام حيث يكون كل رقم هو مجموع الرقمين السابقين. تبدأ المتتالية بـ 0 و 1، ثم 1 (0 + 1)، ثم 2 (1 + 1)، ثم 3 (2 + 1)، ثم 5 (3 + 2)، وهكذا.

1	1	2	3	5	8	13	21	34	55
---	---	---	---	---	---	----	----	----	----

Solving: Using Loop.

```
#include<iostream>
```

```
using namespace std;
```

```
void PrintFibonacciUsingLoop(short Number)
{
```

```
    int FibNumber = 0;
```

```
    int Prev1 = 1, Prev2 = 0;
```

```
    cout << "\n\t1    ";
```

```
    for (short i = 2; i <= Number; i++)
    {
```

```
        FibNumber = Prev1 + Prev2;
```

```
        cout << FibNumber << "    ";
```

```
        Prev2 = Prev1;
```

```
        Prev1 = FibNumber;
```

```
    }
```

```
}
```

```
int main()
```

```
{
```

```
    PrintFibonacciUsingLoop(10);
```

```
    cout << endl;
```

```
    system("pause>0");
```

```
}
```

Problem #22: Fibonacci Series with Recursion

Write a program to print Fibonacci Series of 10.

اكتب برنامجًا لطباعة سلسلة فيبوناتشي المكونة من 10.

1	1	2	3	5	8	13	21	34	55
---	---	---	---	---	---	----	----	----	----

Solving: Using Recursion.

```
#include<iostream>
```

```
using namespace std;
```

```
void PrintFibonacciUsingRecursion(short Number, int Prev1, int Prev2)
```

```
{
```

```
    int FibNumber = 0;
```

```
    if (Number > 0)
```

```
    {
```

```

FibNumber = Prev1 + Prev2;
//      0  +  1
//      1  +  0
//      1  +  1
//      2  +  1
//      3  +  2
//      5  +  3
//      8  +  5
//     13  +  8
//     21  + 13
//     34  + 21
cout << FibNumber << " ";
//      1
//      1
//      2
//      3
//      5
//      8
//     13
//     21
//     34
//     55
Prev2 = Prev1;
//  =  0
//  =  1
//  =  1
//  =  2
//  =  3
//  =  5
//  =  8
//  = 13
//  = 21
//  = 34
Prev1 = FibNumber;
//  =  1
//  =  1
//  =  2
//  =  3
//  =  5
//  =  8
//  = 13
//  = 21
//  = 34
//  = 55
PrintFibonccUsingRecursion(Number - 1, Prev1, Prev2);
//      9      1      0
//      8      1      1
//      7      2      1
//      6      3      2
//      5      5      3
//      4      8      5
//      3     13      8
//      2     21     13
//      1     34     21
//      0     55     34

```

```

}
}

```

```
int main()
{
    PrintFibonacciUsingRecursion(10, 0, 1);

    cout << endl;
    system("pause>0");
}
```

Problem #23: Print First Letter Of Each Word

Write a program to read a string then print the first letter of each word in that string.

اكتب برنامجًا لقراءة سلسلة ثم أطبع الحرف الأول من كل كلمة في تلك السلسلة.

```
Please Enter Your String : Ahmad ElSayed Mohamed AbdelRahim
A
E
M
A
```

Solving:

```
#include<iostream>
#include<string>

usingnamespace std;

string ReadText()
{
    string Word;
    cout <<"Please Enter Your String : ";
    getline(cin, Word);

    return Word;
}

void PrintFirstLetterOfEachWord(stringSt)
{
    bool IsFirstLetter = true;

    for (short i = 0; i <St.length(); i++)
    {
        if (St[i] != ' ' && IsFirstLetter)
        {
            cout <<St[i]<< endl;
        }
        IsFirstLetter = (St[i] == ' ' ? true : false);
    }
}

int main()
{
    PrintFirstLetterOfEachWord(ReadText());

    cout << endl;
    system("pause>0");
}
```

Problem #24: Upper First Letter Of Each Word

Write a program to read a string then Uppercase the first letter of each word in that string.

اكتب برنامجًا لقراءة سلسلة ثم قم بكتابة الحرف الأول من كل كلمة في تلك السلسلة بحروف كبيرة.

Please Enter Your Text : mahmoud ahmad elsayed mohamed

String after conversion :

Mahmoud Ahmad Elsayed Mohamed

Solving:

```
#include<iostream>
#include<string>

using namespace std;

string ReadText()
{
    string St;
    cout << "Please Enter Your Text : ";
    getline(cin, St);
    return St;
}

string UpperFirstLetterOfEachWord(string St)
{
    bool isFirstLetter = true;

    for (short i = 0; i < St.length(); i++)
    {
        if (St[i] != ' ' && isFirstLetter)
        {
            St[i] = toupper(St[i]);
        }

        isFirstLetter = St[i] == ' ' ? true : false;
    }
    return St;
}

int main()
{
    string St = ReadText();

    cout << "\n\tString after conversion : " << endl;

    St = UpperFirstLetterOfEachWord(St);

    cout << "\n\t" << St << endl;

    cout << "\n";
    system("pause>0");
}
```


Problem #25: Lower First Letter Of Each Word

Write a program to read a string then Lowercase the first letter of each word in that string.

اكتب برنامجًا لقراءة سلسلة ثم قم بتصغير الحرف الأول من كل كلمة في تلك السلسلة.

```
Precision Specification of 3.1
Precision Specification of 3.14
Precision Specification of 3.142
Precision Specification of 3.1416
Precision Specification of 3.14159

The float division is: 6.200 / 4.400 = 1.409

The value number is: 34.320
The value number is: 34.3200
```

Solving:

```
#include<iostream>
#include<cstdio>
using namespace std;

int main()
{
    // printf المطلوب لتشغيل
    // Place Holder حاجز للمكان = (" ")
    // Specifiers محددات - مخصصات = %d , %f, %c or %s

    double PI = 3.1415926535898;

    // 1 طريقة
    // %. *f - (خانة) بعد النقطة العشرية
    // نضع الرقم المطلوب للخانات كبرامير

    // Precision Specification مواصفات الدقة
    printf("Precision Specification of %. *f \n", 1, PI);
    printf("Precision Specification of %. *f \n", 2, PI);
    printf("Precision Specification of %. *f \n", 3, PI);
    printf("Precision Specification of %. *f \n", 4, PI);
    printf("Precision Specification of %. *f \n\n", 5, PI);

    // 2 طريقة
    // %.3f - (خانة) بعد النقطة العشرية
    float Num1 = 6.2, Num2 = 4.4;
    printf("The float division is: %.3f / %.3f = %.3f \n\n", Num1, Num2, Num1 / Num2);
    // division = تقسيم

    double D = 34.32;
    printf("The value number is: %.3f \n", D);
    printf("The value number is: %.4f \n\n", D);

    // يسمى هذا فرمات والذي سيظهر على الشاشة اما القيم فهي ثابتة لا تختلف في الكود
    // %. *f or %.3f = used with float and double

    system("pause");
}
```

Problem #26: Upper-Lower All Letter Of a String

Write a program to read a string then uppercase all letters, then lowercase all letters and print them.

اكتب برنامجًا لقراءة سلسلة ثم كتابة الحروف الكبيرة ثم كتابة الحروف الصغيرة وطباعتها.

```
Please enter your text : ahmad elsayed mohamed abdelrahim
```

```
Upper Letters : AHMAD ELSAYED MOHAMED ABDELRAHIM
```

```
Lower Letters : ahmad elsayed mohamed abdelrahim
```

Solving:

```
#include<iostream>
```

```
#include<string>
```

```
usingnamespace std;
```

```
string ReadText()
```

```
{
    string Word;
    cout <<"Please enter your text : ";
    getline(cin, Word);
    return Word;
}
```

```
string UpperLetterOfText(stringSt)
```

```
{
    for (short i = 0; i <St.length(); i++)
    {
        St[i] = toupper(St[i]);
    }
    returnSt;
}
```

```
string LowerLetterOfText(stringSt)
```

```
{
    string StU = UpperLetterOfText(St);
    for (short i = 0; i <St.length(); i++)
    {
        St[i] = tolower(StU[i]);
    }
    returnSt;
}
```

```
int main()
```

```
{
    string St = ReadText();

    cout <<"\nUpper Letters : "<< UpperLetterOfText(St);

    cout <<"\nLower Letters : "<< LowerLetterOfText(St);

    cout << endl;
    system("pause>0");
}
```

Problem #27: Invert Character Case

Write a program to read a character then invert its case and print it.

كتابة برنامج لقراءة الحرف ثم قلب حالته وطباعته.

```
Please enter your text : Q
```

```
=====
```

```
q
```

Solving:

```
#include<iostream>
#include<string>
usingnamespace std;

char ReadCharacter()
{
    char C;
    cout <<"Please enter your text : ";
    cin >> C;
    return C;
}

char InvertLatterCase1(charC)
{
    if (isupper(C))
    {
        C = tolower(C);
    }
    else
    {
        C = toupper(C);
    }
    returnC;
}

char InvertLatterCase2(charC)
{
    isupper(C) ? C = tolower(C) : C = toupper(C);

    returnC;
}

int main()
{
    char C = ReadCharacter();

    //C = InvertLatterCase1(C);

    //cout << C << endl;

    cout <<"\n=====\\n";

    cout << InvertLatterCase2(C) << endl;

    system("pause>0");
}
```

Problem #28: Invert All Letters Case

Write a program to read a string then invert all its letter's case and print it.

اكتب برنامجًا لقراءة سلسلة ثم اقلب جميع حروفها ثم قم بطباعتها.

```
Please enter a character : mahmoud ahmad elsayed mohamed
```

```
MAHMOUD AHMAD ELSAYED MOHAMED
```

Solving:

```
#include<iostream>
```

```
#include<string>
```

```
using namespace std;
```

```
string ReadCharacters()
```

```
{
    string Ch;
    cout << "\n\tPlease enter a character : ";
    getline(cin, Ch);
    return Ch;
}
```

```
string InvertStringCase(string Ch)
```

```
{
    for (short i = 0; i < Ch.length(); i++)
    {
        (char)isupper(Ch[i]) ? Ch[i] = tolower(Ch[i]) : Ch[i] = toupper(Ch[i]);
    }
    return Ch;
}
```

```
int main()
```

```
{
    string Ch = ReadCharacters();

    Ch = InvertStringCase(Ch);

    cout << "\n\t" << Ch << endl;

    system("pause>0");
}
```

Problem #29: Count Small-Capital Letters

Write a program to read a string then count small / capital letters in that string.

اكتب برنامجًا لقراءة سلسلة ثم قم بعد الحروف الصغيرة/الكبيرة في تلك السلسلة.

```
Please enter a character : Ahmed ElSayed Mohamed AbdelRahim
```

```
String Length = 32
```

```
Capital Letters Count = 6
```

```
Small Letters Count = 23
```

Solving:

```
#include<iostream>
#include<string>

using namespace std;

string ReadCharacters()
{
    string Ch;
    cout << "\n\tPlease enter a character : ";
    getline(cin, Ch);
    return Ch;
}

void PrintCountCapitalLetters(string Ch)
{
    int Sum = 0;

    for (short i = 0; i < Ch.length(); i++)
    {
        if (isupper(Ch[i]))
        {
            Sum++;
        }
    }
    cout << "\n\tCapital Letters Count = " << Sum << endl;
}

void PrintCountSmallLetters(string Ch)
{
    int Sum = 0;

    for (short i = 0; i < Ch.length(); i++)
    {
        if (islower(Ch[i]))
        {
            Sum++;
        }
    }
    cout << "\n\tSmall Letters Count = " << Sum << endl;
}

int main()
{
    string Ch = ReadCharacters();

    int L = Ch.length();

    cout << "\n\tString Length = " << L << endl;
    PrintCountCapitalLetters(Ch);
    PrintCountSmallLetters(Ch);

    system("pause>0");
}
```

Problem #30: Count Letters

Write a program to read a string and read a character then count the character in that string.

اكتب برنامجًا لقراءة سلسلة وقراءة حرف ثم قم بعدد الحرف الموجود في تلك السلسلة.

```
Please Enter Your String ? I Love Programming, So I learn some of C++
```

```
Please Enter a Character ? m
```

```
Letter 'm' Count = 3
```

Solving:

```
#include<iostream>
#include<string>
using namespace std;
```

```
string ReadText()
{
    string T;
    cout << "\n\tPlease Enter Your String ? ";
    getline(cin, T);
    return T;
}
```

```
char ReadChar()
{
    char Ch;
    cout << "\n\tPlease Enter a Character ? ";
    cin >> Ch;
    return Ch;
}
```

```
short FindCharater(charCh, stringT)
{
    short Sum = 0;

    for (short i = 0; i < T.length(); i++)
    {
        if (Ch == T[i])
        {
            Sum++;
        }
    }
    return Sum;
}
```

```
int main()
{
    string T = ReadText();
    char Ch = ReadChar();
    cout << "\n\tLetter '" << Ch << "' Count = " << FindCharater(Ch, T) << endl;

    system("pause>0");
}
```

Problem #31: Count Letters (Match Case)

Write a program to read a string and read a character then count the character in that string (Match Case or Not).

اكتب برنامجًا لقراءة سلسلة وقراءة حرف ثم قم بعدد الحرف الموجود في تلك السلسلة (تطابق حالة الأحرف أم لا).

```
Please Enter Your String ? Ahmad Elsayed Mohamed AbdelRahim

Please Enter a Character ? a

Text Length : 32

Letter 'a' Count = 4

Letter 'a' Or 'A' Count = 9
```

Solving:

```
#include<iostream>
#include<string>
usingnamespace std;

string ReadText()
{
    string T;
    cout <<"\n\tPlease Enter Your String ? ";
    getline(cin, T);
    return T;
}

char ReadChar()
{
    char Ch;
    cout <<"\n\tPlease Enter a Character ? ";
    cin >> Ch;
    return Ch;
}

short FindCharater(charCh, stringT)
{
    short Sum = 0;
    for (short i = 0; i <T.length(); i++)
    {
        if (Ch == T[i])
        {
            Sum++;
        }
    }
    return Sum;
}

short FindCharaterMatchCase(charCh, stringT)
{
    short Sum = 0;
    for (short i = 0; i <T.length(); i++)
    {
        if (Ch == (T[i]) || isupper(T[i]))
        {
            Sum++;
        }
    }
    return Sum;}
```

```

int main()
{
    string T = ReadText();
    char Ch = ReadChar();
    char Up = toupper(Ch);

    cout << "\n\tText Lenght : " << T.length() << endl;
    cout << "\n\tLetter '" << Ch << "' Count = " << FindCharater(Ch, T) << endl;
    cout << "\n\tLetter '" << Ch << "' Or '" << Up << "' Count = "
    << FindCharaterMatchCase(Ch, T) << endl;

    system("pause>0");
}

```

Problem #32: Is Vowel

Write a program to read a character the check if it is a Vowel or not (Vowels are: a, e, i, o, u).
 اكتب برنامجاً لقراءة الحرف والتحقق مما إذا كان حرفاً متحركاً أم لا (حروف العلة هي: a, e, i, o, u).

Please Enter a character : Q	Please Enter a character : A
NO Letter 'Q' is NOT Vowel.	YES Letter 'A' is Vowel.

Solving:

```

#include<iostream>
#include<string>
usingnamespace std;

char ReadLetter()
{
    char Ch;
    cout << "\n\tPlease Enter a character : ";
    cin >> Ch;
    return Ch;
}

bool CheckLetter(charCh, charM[5])
{
    Ch = tolower(Ch);

    for (short i = 0; i < 5; i++)
    {
        if (Ch == M[i])
        {
            returntrue;
        }
    }
    returnfalse;
}

void PrintResult(charCh, charM[5])
{
    if (CheckLetter(Ch, M))
        cout << "\n\n\tYES Letter \'< Ch << \' is Vowel." << endl;
    else
        cout << "\n\n\tNO Letter \'< Ch << \' is NOT Vowel." << endl;
}

```



```

int main()
{
    char M[5] = { 'a', 'e', 'i', 'o', 'u' };

    char Ch = ReadLetter();

    PrintResult(Ch, M);

    system("pause>0");
}

```

Problem #33: Count Vowel

Write a program to read a string then count all Vowels in that string (Vowels are: a, e, i, o, u).

اكتب برنامجًا لقراءة سلسلة ثم قم بعد جميع حروف العلة في تلك السلسلة (حروف العلة هي: a, e, i, o, u).

```
Please Enter Your Text : Ahmad ElSayed Mohamed AbdelRahim
```

```
Number Of Vowel is : 12
```

Solving:

```

#include<iostream>
#include<string>

```

```
using namespace std;
```

```

string ReadString()
{
    string S1;
    cout << "\n\tPlease Enter Your Text : ";
    getline(cin, S1);
    return S1;
}

```

```

bool IsVowel(char Ch)
{
    Ch = tolower(Ch);

    return ((Ch == 'a') || (Ch == 'e') || (Ch == 'i') || (Ch == 'o') || (Ch == 'u'));
}

```

```

short CountLettersVowel(string S1)
{
    short Sum = 0;

    for (short i = 0; i < S1.length(); i++)
    {
        if (IsVowel(S1[i]))
        {
            Sum++;
        }
    }
    return Sum;
}

```

```

void PrintResult(string S1)
{
    cout << "\n\n\tNumber Of Vowel is : " << CountLettersVowel(S1) << endl;
}

int main()
{
    string S1 = ReadString();

    PrintResult(S1);

    system("pause>0");
}

```

=====

Problem #34: Print All Vowels In String

Write a program to read a string then print all Vowels in that string (Vowels are: a, e, i, o, u).
 اكتب برنامجًا لقراءة سلسلة ثم أطبع جميع حروف العلة الموجودة في تلك السلسلة (حروف العلة هي: ا، e، i، o، u).

Please Enter Your Text : Ahmad ElSayed Mohamed AbdelRahim

Vowels In String Are : A a E a e o a e A e a i

Solving:

```

#include<iostream>
#include<string>
#include<iomanip>

using namespace std;

string ReadString()
{
    string S1;
    cout << "\n\tPlease Enter Your Text : ";
    getline(cin, S1);
    return S1;
}

bool IsVowel(char Ch)
{
    Ch = tolower(Ch);

    return ((Ch == 'a') || (Ch == 'e') || (Ch == 'i') || (Ch == 'o') || (Ch == 'u'));
}

void PrintVowels(string S1)
{
    for (short i = 0; i < S1.length(); i++)
    {
        if (IsVowel(S1[i]))
        {
            cout << S1[i] << setw(4);
        }
    }
    cout << endl;
}

```

```
int main()
{
    string S1 = ReadString();

    cout <<"\n\tVowels In String Are : ";
    PrintVowels(S1);

    system("pause>0");
}
```

=====

Problem #35: Print Each Word In String

Write a program to read a string then print each word in that string.

اكتب برنامجًا لقراءة سلسلة ثم أطبع كل كلمة في تلك السلسلة.

```
Please Enter Your Text:
Ahmed ElSayed Mohamed AbdelRahim

Your Words In String Are:
Ahmed
ElSayed
Mohamed
AbdelRahim
```

Solving:

```
#include<iostream>
#include<string>
usingnamespace std;

string ReadText()
{
    string S1;
    cout <<"Please Enter Your Text:"<< endl;
    getline(cin, S1);
    return S1;
}

void PrintEachWordInString(stringS1)
{
    string delimi = " ";
    short pos = 0;
    string sWord;

    while ((pos = S1.find(deli)) != std::string::npos)
    {
        sWord =S1.substr(0, pos);

        if (sWord !=" ")
        {
            cout << sWord << endl;
        }
        S1.erase(0, pos + delimi.length());
    }

    if (S1!=" ")
    {
        cout <<S1<< endl;
    }
}
```

```
int main()
{
    string S1 = ReadText();

    cout << "\nYour Words In String Are: \n";

    PrintEachWordInString(S1);

    system("pause>0");
}
```

Problem #36: Count Each Word In String

Write a program to read a string then count each word in that string.

اكتب برنامجًا لقراءة سلسلة ثم قم بعد كل كلمة في تلك السلسلة.

```
Please Enter Your Text : Mahmoud Ahmad ElSayed Mohamed
```

```
The Number Of Words In Your String Is : 4
```

Solving:

```
#include<iostream>
#include<string>
usingnamespace std;

string ReadText()
{
    string S1;
    cout << "Please Enter Your Text : ";
    getline(cin, S1);
    return S1;
}

short CountWords(stringS1)
{
    short counter = 0;
    string delimi = " "; // Delimiter فراغ - فجوة
    short pos = 0;
    string sWord;        // Define a string variable

    // Use find() function to get the position of the delimiters
    while ((pos = S1.find(deli)) != std::string::npos)
    {
        sWord =S1.substr(0, pos); // store the word

        if (sWord != "")
        {
            counter++;
        }
        // erase() until position and move to next word
        S1.erase(0, pos + delimi.length());
    }

    if (S1!=" ")
    {
        counter++; // it print last word of the string
    }
    return counter;
}
```

```
int main()
{
    string S1 = ReadText();
    cout << "\nThe Number Of Words In Your String Is : ";
    cout << CountWords(S1) << endl;

    system("pause>0");
}
```

Problem #37: Split String

Write a program to read a string then make a function to split each word in Vector.

اكتب برنامجًا لقراءة سلسلة ثم قم بإنشاء دالة لفصل (فاصل Spliter) كل كلمة في Vector.

```
Please Enter Your Text : Ahmed ElSayed Mohamed
Tokens = 1
Ahmed ElSayed Mohamed
```

Solving:

```
#include<iostream>
#include<string>
#include<vector>
usingnamespace std;

string ReadText()
{
    string Text;
    cout << "Please Enter Your Text : ";
    getline(cin, Text);
    return Text;
}

vector<string> SplitString(stringS1, stringDelimi)
{
    vector<string> vString;

    short pos = 0;
    string sWord;           // Define a string variable

    // Use find() function to get the position of the delimiters
    while ((pos = S1.find(Delimi)) != std::string::npos)
    {
        sWord = S1.substr(0, pos); // store the word

        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        S1.erase(0, pos + Delimi.length());
    }

    if (S1 != " ")
    {
        vString.push_back(S1); // it print last word of the string
    }
    return vString;
}
```

```

int main()
{
    vector<string>vString;

    string St = ReadText();

    vString = SplitString(St, ",");
    cout <<"Tokens = "<< vString.size() << endl;

    for (string& Text : vString)
    {
        cout << Text << endl;
    }
}

```

Problem #38: Trim Left, Trim Right, Trim

Write a program to read a string then Trim Left, Right, All.

اكتب برنامجًا لقراءة سلسلة ثم قم بقص اليسار واليمين والكل.

```

Please Enter Your Text :      Ahmed ElSayed

String      :      Ahmed ElSayed

Left Trim   : Ahmed ElSayed

Right Trim  :      Ahmed ElSayed

Trim        : Ahmed ElSayed

```

Solving:

```

#include<iostream>
#include<string>

using namespace std;

string ReadText()
{
    string Text;
    cout <<"Please Enter Your Text :";
    getline(cin, Text);
    return Text;
}

string TrimLeft(string S1)
{
    for (short i = 0; i < S1.length(); i++)
    {
        if (S1[i] != ' ')
        {
            return S1.substr(i, S1.length() - i);
        }
    }
    return "";
}

```

```

string TrimRight(string S1)
{
    for (short i = S1.length() - 1; i >= 0; i--)
    {
        if (S1[i] != ' ')
        {
            return S1.substr(0, i + 1);
        }
    }
    return "";
}

string Trim(string S1)
{
    return TrimLeft(TrimRight(S1));
}

int main()
{
    string S1 = ReadText();

    cout << "\nString      : " << S1 << endl;
    cout << "\nLeft Trim   : " << TrimLeft(S1) << endl;
    cout << "\nRight Trim  : " << TrimRight(S1) << endl;
    cout << "\nTrim       : " << Trim(S1) << endl;
    system("pause>0");
}

```

Problem #39: Join String

Write a program to join vector of string into a one string with separators.

اكتب برنامجًا لربط متجه السلسلة في سلسلة واحدة باستخدام الفواصل.

```
Please Enter Your String : Ahmed,ElSayed,Mohamed,AbdelRahim
```

```
=====
```

```
Token = 4
```

```
=====
```

```
Ahmed
```

```
ElSayed
```

```
Mohamed
```

```
AbdelRahim
```

```
=====
```

```
Vector after join:
```

```
Ahmed ElSayed Mohamed AbdelRahim
```

Solving:

```
#include<iostream>
```

```
#include<string>
```

```
#include<vector>
```

```
using namespace std;
```

```

string ReadText()
{
    string Text;
    cout << "Please Enter Your String : ";
    getline(cin, Text);
    return Text;
}

```

```

vector<string> SplitString(string S1, string Delimi)
{
    vector<string> Split;

    short pos = 0;
    string sWord;

    while ((pos = S1.find(Delimi)) != std::string::npos)
    {
        sWord = S1.substr(0, pos);

        if (sWord != "")
        {
            Split.push_back(sWord);
        }
        S1.erase(0, pos + Delimi.length());
    }

    if (S1 != "")
    {
        Split.push_back(S1);
    }
    return Split;
}

string JoinString(vector<string> vString, string Delimi)
{
    string S1 = "";

    for (string& T : vString)
    {
        S1 = S1 + T + Delimi;
    }
    return S1.substr(0, S1.length() - Delimi.length());
}

int main()
{
    vector<string> vString;
    string S1 = ReadText();
    cout << "\n===== \n";
    vString = SplitString(S1, ",");
    cout << "Token = " << vString.size() << endl;
    cout << "===== \n";

    for (string& Txt : vString)
    {
        cout << Txt << endl;
    }

    cout << "===== \n";
    cout << "Vector after join: \n";

    cout << JoinString(vString, " ");

    system("pause>0");
}

```


Problem #40: Join String (Overloading)

Write a program to join Array of string into a one string with separators.

اكتب برنامجًا لدمج مصفوفة من السلسلة في سلسلة واحدة باستخدام الفواصل.

```
Vector After Join :  
Ahmad ElSayed AbdelRahim
```

```
Array After Join :  
Ahmed ElSayed Mohamed
```

Solving:

```
#include<iostream>  
#include<string>  
#include<vector>
```

```
usingnamespace std;
```

```
// دالتين لهما نفس الاسم مع اختلاف البراميترز
```

```
string JoinString(vector<string>vString, stringDelimi)  
{  
    string S1 = "";  
    for (string& T : vString)  
    {  
        S1 = S1 + T +Delimi;  
    }  
    return S1.substr(0, S1.length() - Delimi.length());  
}
```

```
string JoinString(stringArr[], shortlength, stringDelimi)  
{  
    string S2 = "";  
    for (short i = 0; i <length; i++)  
    {  
        S2 = S2 +Arr[i] +Delimi;  
    }  
    return S2.substr(0, S2.length() - Delimi.length());  
}
```

```
int main()  
{  
    vector<string>vString = { "Ahmad", "ElSayed", "AbdelRahim" };  
  
    string Arr[] = { "Ahmed", "ElSayed", "Mohamed" };  
  
    cout <<"\n\tVector After Join : \n";  
  
    cout <<"\t"<< JoinString(vString, " ");  
  
    cout <<"\n\n\tArray After Join : \n";  
    cout <<"\t"<< JoinString(Arr, 3, " ");  
  
    system("pause>0");  
}
```

Problem #41: Reverse Words

Write a program to read string and reverse its words.

اكتب برنامجا لقراءة السلسلة وعكس كلماتها.

```
Please Enter Your Text :  
Ahmed ElSayed Mohamed  
  
Reverse Words In String:  
Mohamed ElSayed Ahmed
```

Solving:

```
#include<iostream>  
  
#include<string>  
  
#include<vector>  
  
usingnamespace std;  
  
string ReadText()  
{  
    string Text;  
    cout <<"Please Enter Your Text : "<< endl;  
    getline(cin, Text);  
    return Text;  
}  
  
vector<string>SplitString(stringS1, stringDelimi)  
{  
    vector<string>vSplit;  
  
    short pos = 0;  
    string sWord = "";  
  
    while ((pos = S1.find(Delimi)) != std::string::npos)  
    {  
        sWord =S1.substr(0, pos);  
  
        if (sWord != "")  
        {  
            vSplit.push_back(sWord);  
        }  
        S1.erase(0, pos + Delimi.length());  
    }  
  
    if (S1!="")  
    {  
        vSplit.push_back(S1);  
    }  
    return vSplit;  
}
```

```

string ReverseWordsInString(string S1)
{
    vector<string> vString;
    string S2 = "";

    vString = SplitString(S1, " ");

    vector<string>::iterator iString; // declare iterator
    iString = vString.end();

    while (iString != vString.begin())
    {
        --iString;

        S2 += *iString + " "; // S2 = S2 + *iString + " "
    }
    S2 = S2.substr(0, S2.length() - 1); // remove last space

    return S2;
}

int main()
{
    string S1 = ReadText();

    cout << "\nReverse Words In String: " << endl;
    cout << ReverseWordsInString(S1);

    system("pause>0");
}

```

Problem #42: Replace Words

Write a program to Replace words in string.

اكتب برنامج لاستبدال الكلمات في السلسلة.

```

String Before Replace : Ahmed ElSayed Mohamed
String After Replace  : Ahmed ElSayed AbdelRahim

```

Solving:

```

#include<iostream>
#include<string>
using namespace std;

string ReplaceWordsInString(string S1, string Word1, string Word2)
{
    short pos = S1.find(Word1);

    while (pos != std::string::npos)
    {
        S1=S1.replace(pos, Word1.length(), Word2);
        pos = S1.find(Word1); // find next word
    }
    return S1;
}

```

```

int main()
{
    string S1 = "Ahmed ElSayed Mohamed";
    cout << "String Before Replace : " << S1 << endl;

    cout << "\nString After Replace : ";
    cout << ReplaceWordsInString(S1, "Mohamed", "AbdelRahim") << endl;

    system("pause>0");
}

```

Problem #43: Replace Words (Custom)

Write a program to Replace words in string using custom function.

اكتب برنامجًا لاستبدال الكلمات في السلسلة باستخدام دالة مخصصة.

```

Original String : Ahmed ElSayed Mohamed

Replace Word With Match Case :
Ahmed ElSayed Mohamed

Replace Word Without Match Case :
Ahmed ElSayed AbdelRahim

```

Solving:

```

#include<iostream>
#include<string>
#include<vector>

using namespace std;

// دالة لفصل سلسلة كاملة لكلمات منفصلة
vector<string> SplitString(string S1, string Delimi)
{
    vector<string> vString;

    short pos = 0;
    string sWord = "";

    while ((pos = S1.find(Delimi)) != std::string::npos)
    {
        sWord = S1.substr(0, pos);

        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        S1.erase(0, pos + Delimi.length());
    }
    if (S1 != "")
    {
        vString.push_back(S1);
    }
    return vString;
}

// دالة لتجميع كلمات منفصلة في سلسلة كاملة

```

```

string JoinString(vector<string>vString, stringDelimi)
{
    string S1 = "";
    for (string& T : vString)
    {
        S1 = S1 + T +Delimi;
    }
    return S1.substr(0, S1.length() - Delimi.length());
}
// دالة لجعل الحروف الكبيرة التي في الكلمات في السلسلة لحروف صغيرة
string LowerAllString(stringS1)
{
    for (short i = 0; i <S1.length(); i++)
    {
        if (isupper(S1[i]))
        {
            S1[i] = tolower(S1[i]);
        }
    }
    returnS1;
}
// دالة لاستبدال كلمة مكان كلمة في السلسلة
string ReplaceWordInStringUsingMatchCase(stringS1, stringW1, stringW2, boolMatchCase = true)
{
    vector<string>vString;
    vString = SplitString(S1, " ");
    for (string& S : vString)
    {
        if (MatchCase)
        {
            if (S ==W1)
            {
                S =W2;
            }
        }
        else
        {
            if (LowerAllString(S) == LowerAllString(W1))
            {
                S =W2;
            }
        }
    }
    return JoinString(vString, " ");
}
int main()
{
    string S1 = "Ahmed ElSayed Mohamed";
    string W1 = "mohamed";
    string W2 = "AbdelRahim";
    cout <<"\nOriginal String : "<< S1 << endl;
    cout <<"\nReplace Word With Match Case : "<< endl;
    cout << ReplaceWordInStringUsingMatchCase(S1, W1, W2) << endl;
    cout <<"\n\nReplace Word Without Match Case : "<< endl;
    cout << ReplaceWordInStringUsingMatchCase(S1, W1, W2, false) << endl;
    system("pause>0");
}

```

Problem #44: Remove Punctuations

Write a program to remove all Punctuations from a string.

اكتب برنامجًا لإزالة جميع علامات الترقيم من السلسلة.

```
Original String :  
Welcome to Egypt, Egypt is a nice country; it's amazing.  
  
Punctuations Removed:  
Welcome to Egypt Egypt is a nice country its amazing
```

Solving:

```
#include<iostream>  
#include<string>  
  
using namespace std;  
  
// دالة حذف أي علامة خاصة من السلسلة  
string RemovePunctuationFromString(string S1)  
{  
    string S2 = "";  
  
    for (short i = 0; i < S1.length(); i++)  
    {  
        if (!ispunct(S1[i]))  
        {  
            S2 += S1[i];  
        }  
    }  
    return S2;  
}  
  
int main()  
{  
    string S1 = "Welcome to Egypt, Egypt is a nice country; it's amazing."  
  
    cout << "Original String : \n" << S1 << endl;  
  
    cout << "\nPunctuations Removed: \n" << RemovePunctuationFromString(S1) << endl;  
  
    system("pause>0");  
}
```

Problem #45: Convert Record To Line

Write a program to read bank client data record and convert it to one line..

كتابة برنامج لقراءة سجل بيانات عميل البنك وتحويله إلى سطر واحد.

```
Please Enter Client Data:  
Enter Account Number ? A155  
Enter PinCode ? 95123  
Enter Full Name ? Ali Ahmed  
Enter Phone ? 01236548596  
Enter AccountBalance ? 600  
  
Client Record for Saving is:  
A155#/#95123#/#Ali Ahmed#/#01236548596#/#600.000000
```

Solving:

```
#include<iostream>
#include<string>
using namespace std;

struct sClient {
    string Account_Number;
    string PinCode;
    string FullName;
    string Phone;
    float AccountBalance;
};

sClient ReadClientData()
{
    sClient Client;

    cout << "Enter Account Number ? ";
    getline(cin >> ws, Client.Account_Number);

    cout << "Enter PinCode ? ";
    getline(cin, Client.PinCode);

    cout << "Enter Full Name ? ";
    getline(cin, Client.FullName);

    cout << "Enter Phone ? ";
    getline(cin, Client.Phone);

    cout << "Enter AccountBalance ? ";
    cin >> Client.AccountBalance;

    return Client;
}

string ConvertRecordToLine(sClient Client, string Separator = "#//#")
{
    string S2 = "";

    S2 += Client.Account_Number + Separator;
    S2 += Client.PinCode + Separator;
    S2 += Client.FullName + Separator;
    S2 += Client.Phone + Separator;
    S2 += to_string(Client.AccountBalance);
    return S2;
}

int main()
{
    cout << "Please Enter Client Data: \n";
    sClient Client;
    Client = ReadClientData();
    cout << "\nClient Record for Saving is: \n";
    cout << ConvertRecordToLine(Client) << endl;

    system("pause>0");
}
```

Problem #46: Convert Line Date To Record

Write a program to Convert Line Data to Record and print it.

اكتب برنامجًا لتحويل سطر بيانات إلى سجل ثم قم بطباعته.

```
Line Record is:
A150#/#1234#/#Ahmed ElSayed#/#01000000000#/#500.000000

The following is the extracted client record:
=====
Account Number : A150
Pin Code       : 1234
Full Name      : Ahmed ElSayed
Phone          : 01000000000
Account Balance: 500
```

Solving:

```
#include<iostream>
#include<string>
#include<vector>
```

```
using namespace std;
```

```
struct stClient {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    float AccountBalance;
};
```

```
// دالة فصل السلسلة الى سجلات منفصلة باستخدام فاصل #//#
vector<string> SplitString(string Line, string Delimi = "#//#")
{
    vector<string> vString;

    short pos = 0;
    string sWord = "";

    while ((pos = Line.find(Delimi)) != std::string::npos)
    {
        sWord = Line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        Line.erase(0, pos + Delimi.length());
    }
    if (Line != "")
    {
        vString.push_back(Line);
    }
    return vString;
}
```



```

stClient ConvertLineToRecord(stringLine, stringseparator = "#//")
{
    stClient eClient;
    vector<string>vString;

    vString = SplitString(Line);

    eClient.Account_Number = vString[0];
    eClient.PinCode = vString[1];
    eClient.Full_Name = vString[2];
    eClient.Phone = vString[3];
    eClient.AccountBalance = stof(vString[4]);

    return eClient;
}

```

```

void PrintClientRecord(stringLine)
{
    cout << "\nThe following is the extracted client record:\n";
    cout << "=====\n";
    stClient eClient;

    eClient = ConvertLineToRecord(Line);

    cout << "Account Number : " << eClient.Account_Number << endl;
    cout << "Pin Code       : " << eClient.PinCode << endl;
    cout << "Full Name        : " << eClient.Full_Name << endl;
    cout << "Phone           : " << eClient.Phone << endl;
    cout << "Account Balance: " << eClient.AccountBalance << endl;
}

int main()
{
    string sLine = "A150//1234//Ahmed
ElSayed//01000000000//500.000000";

    cout << "Line Record is:\n";
    cout << sLine << endl;

    PrintClientRecord(sLine);

    system("pause>0");
}

```

Problem #47: Add Clients To File

Write a program to ask you to enter clients and save them to file.

اكتب برنامجًا يطلب منك إدخال العملاء وحفظهم في ملف.

```
Adding New Client:
Enter Account Number ? A153
Enter PinCode ? 75321
Enter Ful Name ? Ali Maher
Enter Phone ? 76541239
Enter Account Balance ? 600

Client Added Successfully, do you want to add more clients? (Y/N): n
```

Solving:

```
#include<iostream>
#include<string>
#include<fstream>
```

```
using namespace std;
```

// عمل متغير ثابت لاسم الملف في البداية

```
const string ClientsFileName = "Clients.txt";
```

```
struct stClient {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    float AccountBalance;
};
```

```
stClient ReadNewClient()
{
```

```
    stClient Client;
```

```
    cout << "Enter Account Number ? ";
```

```
    // usage for std::ws will extract all the whitespace character
```

// سيؤدي استخدام std::ws إلى استخراج كل المسافة البيضاء

```
    getline(cin >> ws, Client.Account_Number);
```

```
    cout << "Enter PinCode ? ";
```

```
    getline(cin, Client.PinCode);
```

```
    cout << "Enter Ful Name ? ";
```

```
    getline(cin, Client.Full_Name);
```

```
    cout << "Enter Phone ? ";
```

```
    getline(cin, Client.Phone);
```

```
    cout << "Enter Account Balance ? ";
```

```
    cin >> Client.AccountBalance;
```

```
    return Client;
```

```
}
```

```

string ConvertRecordToLine(stClientsClient, stringDelimi = "#//#")
{
    string line = "";
    line +=sClient.Account_Number +Delimi;
    line +=sClient.PinCode +Delimi;
    line +=sClient.Full_Name +Delimi;
    line +=sClient.Phone +Delimi;
    line += to_string(sClient.AccountBalance);
    return line;
}

void AddClientDataToFile(stringfilename, stringLine)
{
    fstream to_File;

    to_File.open(filename, ios::out | ios::app);

    if (to_File.is_open())
    {
        to_File <<Line<< endl;

        to_File.close();
    }
}

void AddNewClient()
{
    stClient Client;
    Client = ReadNewClient();
    //string line = ConvertRecordToLine(Client);
    AddClientDataToFile(ClientsFileName, ConvertRecordToLine(Client));
}

void AddClients()
{
    char AddMore = 'Y';
    do
    {
        system("cls");

        cout <<"Adding New Client:\n\n";

        AddNewClient();

        cout <<"\nClient Added Successfully, do you want to add more clients? (Y/N): ";
        cin >> AddMore;

    } while (toupper(AddMore) == 'Y');
}

int main()
{
    AddClients();

    system("pause>0");
}

```

Problem #48: Show All Clients

Write a program to read clients file and show them on the screen as follows.

قم بكتابة برنامج لقراءة ملفات العملاء وإظهارهم على الشاشة كما يلي:

Client List (3) Client(s)				
Account Number	PinCode	Full Name	Phone	Account Balance
A150	123	Ahmed ElSayed	01000000000	100
A151	456	Mahmoud Ahmed	01111111111	200
A152	789	Shahd Ahmed	01222222222	300

Solving:

```
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>

using namespace std;

const string ClientsFileName = "Clients.txt";

struct stClient {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    int AccountBalance = 0;
};

vector<string> SplitString(string Line, string Delimi = "#//#")
{
    vector<string> vString;

    short pos = 0;
    string sWord = "";

    while ((pos = Line.find(Delimi)) != std::string::npos)
    {
        sWord = Line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        Line.erase(0, pos + Delimi.length());
    }
    if (Line != "")
    {
        vString.push_back(Line);
    }
    return vString;
}
```

```

stClient ConvertLineToRecord(stringLine, stringseparator = "#//#")
{
    stClient eClient;
    vector<string>vString;
    vString = SplitString(Line);
    eClient.Account_Number = vString[0];
    eClient.PinCode = vString[1];
    eClient.Full_Name = vString[2];
    eClient.Phone = vString[3];
    eClient.AccountBalance = stof(vString[4]);
    return eClient;
}

vector<stClient> LoadClientDataFromFile(stringfilename)
{
    vector<stClient>vClients;
    fstream fr_File;

    fr_File.open(filename, ios::in);
    if (fr_File.is_open())
    {
        string Line; // لتخزين السطر الذي تم أخذه من الملف
        stClient eClient; // لتخزين السجلات التي تم فصلها بواسطة دالة تحويل السطر الى سجل

        while (getline(fr_File, Line))
        {
            eClient = ConvertLineToRecord(Line);
            // نعطي السطر لدالة تحويل السطر الى سجل لتخرج لنا سجل منفصل ويخزن في المتغير من نوع استركتشر
            vClients.push_back(eClient);
        }
        fr_File.close();
    }
    return vClients;
}

void PrintClientRecord(stClientdClient)
{
    cout <<"|"<< left << setw(15) <<dClient.Account_Number;
    cout <<"|"<< left << setw(10) <<dClient.PinCode;
    cout <<"|"<< left << setw(30) <<dClient.Full_Name;
    cout <<"|"<< left << setw(12) <<dClient.Phone;
    cout <<"|"<< left << setw(12) <<dClient.AccountBalance;
}

void PrintHeaderOfTable(vector<stClient>vClient)
{
    cout <<"\n\t\t\t\t\tClient List ("<<vClient.size() <<" ) Client(s)\n";
    cout <<"\n===== ";
    cout <<"===== \n";
    cout <<"|"<< left << setw(15) <<"Account Number";
    cout <<"|"<< left << setw(10) <<"PinCode";
    cout <<"|"<< left << setw(30) <<"Full Name";
    cout <<"|"<< left << setw(12) <<"Phone";
    cout <<"|"<< left << setw(14) <<"Account Balance";
    cout <<"\n----- ";
    cout <<"----- \n";
}

```

```

void PrintFooterOfTable()
{
    cout <<"=====";
    cout <<"=====\n";
}

void PrintAllClientsData(vector<stClient>vClient, stringfilename)
{
    PrintHeaderOfTable(vClient);

    for (stClient& Client : vClient)
    {
        PrintClientRecord(Client);
        cout << endl;
    }
    PrintFooterOfTable();
}

int main()
{
    vector<stClient>vClients = LoadClientDataFromFile(ClientsFileName);

    PrintAllClientsData(vClients, ClientsFileName);

    system("pause>0");
}

```

Problem #49: Find Client By Account Number

Write a program to find clients by Account Number and print it to the screen.

كتابة برنامج للعثور على العملاء عن طريق رقم الحساب وطباعته على الشاشة.

```
Please enter Account Number of Client? A150
```

```
The following are the client details:
```

```

Account Number : A150
Pin Code       : 123
Full Name      : Ahmed ElSayed
Phone          : 01000000000
Account Balance: 100

```

Solving:

```

#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>

usingnamespace std;

conststring ClientsFileName = "Clients.txt";

```

```

struct stClient {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    int AccountBalance = 0;
};

vector<string> SplitString(string Line, string Delimi = "#//#")
{
    vector<string> vString;
    short pos = 0;
    string sWord = "";
    while ((pos = Line.find(Delimi)) != std::string::npos)
    {
        sWord = Line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        Line.erase(0, pos + Delimi.length());
    }
    if (Line != "")
    {
        vString.push_back(Line);
    }
    return vString;
}

stClient ConvertLineToRecord(string Line, string separator = "#//#")
{
    stClient eClient;
    vector<string> vString;
    vString = SplitString(Line);
    eClient.Account_Number = vString[0];
    eClient.PinCode = vString[1];
    eClient.Full_Name = vString[2];
    eClient.Phone = vString[3];
    eClient.AccountBalance = stof(vString[4]);
    return eClient;
}

vector<stClient> LoadClientDataFromFile(string filename)
{
    vector<stClient> vClients;
    fstream fr_File;
    fr_File.open(filename, ios::in);
    if (fr_File.is_open())
    {
        string Line;
        stClient eClient;
        while (getline(fr_File, Line))
        {
            eClient = ConvertLineToRecord(Line);

            vClients.push_back(eClient);
        }
        fr_File.close();
    }
    return vClients;
}

```

```

bool FindClientByAccountNumber(string Request, stClient& Client)
{
    vector<stClient> vClients = LoadClientDataFromFile(ClientsFileName);

    for (stClient C : vClients)
    {
        if (C.Account_Number == Request)
        {
            Client = C;
            return true;
        }
    }
    return false;
}

void PrintClientRecord(stClient Client)
{
    cout << "\n\nThe following are the client details:\n\n";

    cout << "Account Number : " << Client.Account_Number << endl;
    cout << "Pin Code       : " << Client.PinCode << endl;
    cout << "Full Name        : " << Client.Full_Name << endl;
    cout << "Phone           : " << Client.Phone << endl;
    cout << "Account Balance: " << Client.AccountBalance << endl;
}

void IsAccountNumberFind()
{
    string AccountNumber;
    cout << "Please enter Account Number of Client? ";
    cin >> AccountNumber;

    stClient Client;

    if (FindClientByAccountNumber(AccountNumber, Client))
    {
        PrintClientRecord(Client);
    }
    else
    {
        cout << "\nSorry! Client with Account Number (" << AccountNumber << ")
is Not Found" << endl;
    }
}

int main()
{
    IsAccountNumberFind();

    system("pause>0");
}

```


Problem #50: Delete Client By Account Number

Write a program to delete client by Account Number.

اكتب برنامج لحذف العميل عن طريق رقم الحساب.

```
Please enter Account Number of Client? A152

The following are the client details:

Account Number : A152
Pin Code       : 789
Full Name      : Shahd Ahmed
Phone          : 01222222222
Account Balance: 300

Are you sure you want to delete this client? (Y/N): y

Client Deleted Successfully.

Please enter Account Number ? A153
Client with account number (A153) not found!
```

Solving:

```
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>
using namespace std;

const string ClientsFileName = "Clients.txt";

struct Client {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    int AccountBalance = 0;
    bool MarkForDelete = false;
};

vector<string> SplitString(string Line, string Delimi = "#//#")
{
    vector<string> vString;
    short pos = 0;
    string sWord = "";
    while ((pos = Line.find(Delimi)) != std::string::npos)
    {
        sWord = Line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        Line.erase(0, pos + Delimi.length());
    }
    if (Line != "")
    {
        vString.push_back(Line);
    }
    return vString;
}
```

```

stClient ConvertLineToRecord(stringLine, stringseparator = "#//#")
{
    stClient eClient;
    vector<string>vString;
    vString = SplitString(Line);
    eClient.Account_Number = vString[0];
    eClient.PinCode = vString[1];
    eClient.Full_Name = vString[2];
    eClient.Phone = vString[3];
    eClient.AccountBalance = stof(vString[4]);
    return eClient;
}

string ConvertRecordToLine(stClientClient, stringseparator = "#//#")
{
    string Line = "";
    Line = Line +Client.Account_Number +separator;
    Line = Line +Client.PinCode +separator;
    Line = Line +Client.Full_Name +separator;
    Line = Line +Client.Phone +separator;
    Line = Line + to_string(Client.AccountBalance);
    return Line;
}

vector<stClient> LoadClientDataFromFile(stringfilename)
{
    vector<stClient>vClients;

    fstream fr_File;

    fr_File.open(filename, ios::in);

    if (fr_File.is_open())
    {
        string Line;
        stClient eClient;

        while (getline(fr_File, Line))
        {
            eClient = ConvertLineToRecord(Line);

            vClients.push_back(eClient);
        }
        fr_File.close();
    }
    return vClients;
}

void PrintClientCard(stClient&Client)
{
    cout <<"\n\nThe following are the client details:\n\n";
    cout <<"Account Number : "<<Client.Account_Number << endl;
    cout <<"Pin Code       : "<<Client.PinCode << endl;
    cout <<"Full Name        : "<<Client.Full_Name << endl;
    cout <<"Phone           : "<<Client.Phone << endl;
    cout <<"Account Balance: "<<Client.AccountBalance << endl;
}

```

```

bool FindClientByAccountNumber(stringRequest, vector<stClient>&vClients,
stClient&Client)
{
    for (stClient C : vClients)
    {
        if (C.Account_Number ==Request)
        {
            Client= C;
            returntrue;
        }
    }
    returnfalse;
}

bool MarkClientForDeleteByAccountNumber(stringAccountNumber, vector<stClient>&vClient)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            C.MarkForDelete = true;
            returntrue;
        }
    }
    returnfalse;
}

vector<stClient> SaveClientDataToFile(stringfilename, vector<stClient>&vClient)
{
    string DataLine;

    fstream fr_File;

    fr_File.open(filename, ios::out);

    if (fr_File.is_open())
    {
        for (stClient& C : vClient)
        {
            if (C.MarkForDelete == false)
            {
                DataLine = ConvertRecordToLine(C);

                fr_File << DataLine << endl;
            }
        }
        fr_File.close();
    }
    returnvClient;
}

```

```

bool DeleteClientByAccountNumber(string AccountNumber, vector<stClient>&vClient)
{
    stClient Client;
    char Answer = 'n';

    if (FindClientByAccountNumber(AccountNumber, vClient, Client))
    {
        PrintClientCard(Client);

        cout <<"\n\nAre you sure you want to delete this client? (Y/N): ";
        cin >> Answer;

        if (Answer == 'y' || Answer == 'Y')
        {
            MarkClientForDeleteByAccountNumber(AccountNumber, vClient);
            SaveClientDataToFile(ClientsFileName, vClient);

            vClient= LoadClientDataFromFile(ClientsFileName);

            cout <<"\n\nClient Deleted Successfully."<< endl;
            return true;
        }
    }
    else
    {
        cout <<"\n\nClient With Account Number (B33) Not Found!"<< endl;
    }
    return false;
}

string ReadClientAccountNumber()
{
    string AccountNumber;
    cout <<"Please enter Account Number of Client? ";
    cin >> AccountNumber;
    return AccountNumber;
}

int main()
{
    vector<stClient>vClient = LoadClientDataFromFile(ClientsFileName);

    string AccountNumber = ReadClientAccountNumber();

    DeleteClientByAccountNumber(AccountNumber, vClient);

    system("pause>0");
}

```

Problem #51: Update Client By Account Number

Write a program to Update client data by Account Number.

كتابة برنامج لتحديث بيانات العميل عن طريق رقم الحساب.

```
Please enter Account Number ? A150
The Following are the client details:
Account Number :A150
Pin Code       :123
Full Name      :Ahmed ElSayed
Phone          :01000000000
Account Balance:100
Are you sure you want update this client? (Y/N) : y
Enter Pin Code : 458
Enter Full Name: Ahmad AbdelRahim
Enter Phone    : 012365478963
Enter Account Balance: 1500
Client Updated Successfully.
Please enter Account Number ? A153
Client with account number (A153) not found!
```

Solving:

```
#include<iostream>
#include<fstream>
#include<string>
#include<vector>
```

```
usingnamespace std;
```

```
conststring ClientFileName = "Clients.txt";
```

```
structstClient {
    string Account_Number;
    string PinCode;
    string FullName;
    string Phone;
    float AccountBalance;

    bool MarkForDelete = false;
};

vector<string> SplitString(stringline, stringDelimi = "#//#")
{
    vector<string>vString;

    short pos = 0;
    string sWord = "";

    while ((pos = line.find(Delimi)) != std::string::npos)
    {
        sWord =line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        line.erase(0, pos + Delimi.length());
    }
}
```

```

    if (line!="")
    {
        vString.push_back(line);
    }
    return vString;
}

stClient ConvertLineToRecord(stringLine)
{
    vector<string>vClient;

    vClient = SplitString(Line);

    stClient Client;

    Client.Account_Number = vClient[0];
    Client.PinCode = vClient[1];
    Client.FullName = vClient[2];
    Client.Phone = vClient[3];
    Client.AccountBalance = stof(vClient[4]);

    return Client;
}

vector<stClient> LoadClientDataFromFile(stringfilename)
{
    vector<stClient>vClient;

    fstream NewFile;

    NewFile.open(filename, ios::in);

    if (NewFile.is_open())
    {
        string line;
        stClient Client;

        while (getline(NewFile, line))
        {
            Client = ConvertLineToRecord(line);
            vClient.push_back(Client);
        }
        NewFile.close();
    }
    return vClient;
}

string ReadAccountNumber()
{
    string AccountNumber;
    cout <<"Please enter Account Number ? ";
    cin >> AccountNumber;
    return AccountNumber;
}

```

```

bool FindClientByAccountNumber(string AccountNumber, vector<stClient>&vClient,
stClient&Client)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            Client= C;
            returntrue;
        }
    }
    returnfalse;
}

void PrintClientCard(stClientClient)
{
    cout <<"\n\nThe Following are the client details:\n\n";

    cout <<"Account Number : "<<Client.Account_Number << endl;
    cout <<"Pin Code      : "<<Client.PinCode << endl;
    cout <<"Full Name       : "<<Client.FullName << endl;
    cout <<"Phone          : "<<Client.Phone << endl;
    cout <<"Account Balance: "<<Client.AccountBalance << endl;
}

string ConvertRecordToLine(stClientClient, stringDelimi = "#//#")
{
    string line = "";

    line +=Client.Account_Number +Delimi;
    line +=Client.PinCode +Delimi;
    line +=Client.FullName +Delimi;
    line +=Client.Phone +Delimi;
    line += to_string(Client.AccountBalance);

    return line;
}

stClient ChangeClientRecord(stringAccountNumber)
{
    stClient Client;
    Client.Account_Number =AccountNumber;

    cout <<"Enter Pin Code : ";
    getline(cin >> ws, Client.PinCode);

    cout <<"Enter Full Name: ";
    getline(cin, Client.FullName);

    cout <<"Enter Phone      : ";
    getline(cin, Client.Phone);

    cout <<"Enter Account Balance: ";
    cin >> Client.AccountBalance;

    return Client;
}

```

```

bool MarkClientForDeleteByAccountNumber(string AccountNumber, vector<stClient>&vClient)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number == AccountNumber)
        {
            C.MarkForDelete = true;
            return true;
        }
    }
    return false;
}

```

```

vector<stClient> SaveClientDataToFile(string filename, vector<stClient>&vClient)
{
    string Line;

    fstream NewFile;

    NewFile.open(filename, ios::out);

    if (NewFile.is_open())
    {
        for (stClient& C : vClient)
        {
            if (C.MarkForDelete == false)
            {
                Line = ConvertRecordToLine(C);
                NewFile << Line << endl;
            }
        }
        NewFile.close();
    }
    return vClient;
}

```

```

bool UpdateClientDataByAccountNumber(string AccountNumber,
vector<stClient>&vClient)
{
    stClient Client;
    char Answer = 'n';

    if (FindClientByAccountNumber(AccountNumber, vClient, Client))
    {
        PrintClientCard(Client);

        cout << "Are you sure you want update this client? (Y/N) : ";
        cin >> Answer;

        if (Answer == 'Y' || Answer == 'y')
        {
            for (stClient& C : vClient)
            {
                if (C.Account_Number == AccountNumber)
                {
                    C = ChangeClientRecord(AccountNumber);
                    break;
                }
            }
        }
    }
}

```


*/ طالما انا لاقيت الكلاينت وعدلت معلوماته بعمل بريك علطول مفيش داعي اكمل - اعد على الفاضي
افرض عندي عشرتلاف كلاينت موجود ينعد في الفايل
بدي امشي على العشرتلاف
ما انا اول واحد لاقيته خلاص عدلت عليه اعمل بريك . . البريك هابتطلعني من اللوب
ليش عملنا ابريك معلمتش ريتيرن = لان بدي اكمل شغل جوه الفانكشن - الريترن تطلع بره الفانكشن
*/

```

    }
}
SaveClientDataToFile(ClientFileName, vClient);

cout <<"Client Updated Successfully."<< endl;
return true;
}

}
else
{
    cout <<"Client with account number ("<<AccountNumber<<") not
found!"<< endl;
}
return false;
}

int main()
{
    vector<stClient>vClient;

    vClient = LoadClientDataFromFile(ClientFileName);

    string AccountNumber = ReadAccountNumber();

    UpdateClientDataByAccountNumber(AccountNumber, vClient);

    system("pause>0");

}

```

Project #01:Bank 1 : Requirements:

برامج بنك: متطلبات البرنامج :

```
=====
Main Menue Screen
=====

[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.

=====
Choose what do you want to do ? [1 to 6] ?
```

```
=====
Main Menue Screen
=====

[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.

=====
Choose what do you want to do ? [1 to 6] ? 1
```

```
=====
Client List (3) Client(s)
=====
|Account Number |Pic Code |Client Name |Phone |Balance
|=====|
|A150 |458 |Ahmad AbdelRahim |012365478963 |1500
|A151 |456 |Mahmoud Ahmed |01111111111 |200
|A152 |789 |Shahd Ahmed |01222222222 |300
|=====|

Press any key to go back to Main Menue...
```

```
=====
Main Menue Screen
=====

[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.

=====
Choose what do you want to do ? [1 to 6] ? 2_
```

```
=====
Add New Clients Screen
=====

Adding New Client:
Enter Account Number ? A153
Enter PinCode ? 756
Enter Ful Name ? Ali Maher
Enter Phone ? 7856932143
Enter Account Balance ? 400

Client Added Successfully, do you want to add more clients? (Y/N): n

Press any key to go back to Main Menue...
```

```
=====
Main Menue Screen
=====

[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.

=====
Choose what do you want to do ? [1 to 6] ? 3_
```

```
=====
Delete Client Screen
=====

Please enter Account Number: A153

The Following are the client details:
Account Number : A153
Pin Code : 756
Full Name : Ali Maher
Phone : 7856932143
Account Balance: 400

Are you sure you want delete this client ? (Y / N) : y

Client Deleted Successfully.

Press any key to go back to Main Menue...
```

```

=====
Main Menu Screen
=====
[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.
=====
Choose what do you want to do ? [1 to 6] ? 4_

```

```

=====
Update Client Info Screen
=====

Please enter Account Number: A150

The Following are the client details:
Account Number : A150
Pin Code       : 458
Full Name      : Ahmad AbdelRahim
Phone          : 012365478963
Account Balance: 1500

Are you sure you want update this client ? (Y / N) : y
Enter PinCode ? 123
Enter Ful Name ? Ahmed ElSayed
Enter Phone ? 01478963258
Enter Account Balance ? 2000

Client Updated Successfully.

Press any key to go back to Main Menu...

```

```

=====
Main Menu Screen
=====
[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.
=====
Choose what do you want to do ? [1 to 6] ? 5

```

```

=====
Find Client Screen
=====

Please enter Account Number: A151

The Following are the client details:
Account Number : A151
Pin Code       : 456
Full Name      : Mahmoud Ahmed
Phone          : 01111111111
Account Balance: 200

Press any key to go back to Main Menu...

```

```

=====
Main Menu Screen
=====
[1] Show Client List.
[2] Add New Client.
[3] Delete Client.
[4] Update Client Info.
[5] Find Client.
[6] Exit.
=====
Choose what do you want to do ? [1 to 6] ? 6_

```

```

=====
Program End, THANK YOU :- )
=====

```

Solving:

```

#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>

```

```
using namespace std;
```

```
// عمل متغير ثابت اسم الملف في البداية
```

```
const string ClientsFileName = "Clients.txt";
```

```
void ShowMainMenuScreen();
```

```

struct stClient {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    float AccountBalance;
    // Flag - لا يخزن على الفايل.. لكن نستخدمه عند وضع قيمة ترو له كعلامة للحذف
    bool MarkForDelete = false;
};

stClient CovertLineToRecord(string line);

//----- Add Clients To File -----//

bool ClientExistsByAccountNumber(string AccountNumber, string filename)
    // للتحقق إذا كان الـ Account Number موجود أصلاً
{
    vector<stClient> vClient;
    fstream NewFile;

    NewFile.open(filename, ios::in);
    if (NewFile.is_open())
    {
        string line;
        stClient Client;

        while (getline(NewFile, line))
        {
            Client = CovertLineToRecord(line);
            if (Client.Account_Number == AccountNumber)
            {
                NewFile.close(); // مهم جداً ينقل الملف هنا
                return true;
            }
            vClient.push_back(Client);
        }
        NewFile.close();
    }
    return false;
}

```

```

stClient ReadClientData()
{
    stClient Client;

    cout <<"Enter Account Number ? ";
    // usage for std::ws will extract all the whitespace character
    // سيؤدي استخدام std::ws إلى استخراج كل أحرف المسافة البيضاء
    getline(cin >> ws, Client.Account_Number);

    // للتحقق إذا كان الـ Account Number موجود أصلا
    while(ClientExistsByAccountNumber(Client.Account_Number, ClientsFileName))
    {
        cout <<"Client with ["<< Client.Account_Number
            <<"] already exists, Enter Another Account Number: ";
        getline(cin >> ws, Client.Account_Number);
    }
    cout <<"Enter PinCode ? ";
    getline(cin, Client.PinCode);

    cout <<"Enter Ful Name ? ";
    getline(cin, Client.Full_Name);

    cout <<"Enter Phone ? ";
    getline(cin, Client.Phone);

    cout <<"Enter Account Balance ? ";
    cin >> Client.AccountBalance;

    return Client;
}

string ConvertRecordToLine(stClientsClient, stringDelimi = "#//#")
{
    string line = "";

    line +=sClient.Account_Number +Delimi;
    line +=sClient.PinCode +Delimi;
    line +=sClient.Full_Name +Delimi;
    line +=sClient.Phone +Delimi;
    line += to_string(sClient.AccountBalance);

    return line;
}

void AddClientDataToFile(stringfilename, stringLine)
{
    fstream to_File;

    to_File.open(filename, ios::out | ios::app);

    if (to_File.is_open())
    {
        to_File <<Line<< endl;

        to_File.close();
    }
}

```

```

void AddNewClient()
{
    stClient Client;
    Client = ReadClientData();

    AddClientDataToFile(ClientsFileName, ConvertRecordToLine(Client));
}

void AddClients()
{
    char AddMore = 'Y';
    do
    {
        cout <<"\nAdding New Client:\n\n";
        AddNewClient();
        cout <<"\nClient Added Successfully, do you want to add more clients?
(Y/N): ";
        cin >> AddMore;

    } while (toupper(AddMore) == 'Y');
}

void ShowAddNewClientsScreen()
{
    system("cls");

    cout <<"\n=====\\n";
    cout <<"\t\tAdd New Clients Screen\\n";
    cout <<"=====\\n";

    AddClients();
}

//----- Show All Clients -----//

void HeaderOfTable(vector<stClient>vClient)
{
    cout <<"\n\t\t\t\tClient List ("<<vClient.size() <<") Client(s)\n";
    cout <<"\n=====\\n";
    cout <<"=====\\n";
    cout <<"|"<< left << setw(15) <<"Account Number";
    cout <<"|"<< left << setw(12) <<"Pic Code";
    cout <<"|"<< left << setw(30) <<"Client Name";
    cout <<"|"<< left << setw(14) <<"Phone";
    cout <<"|"<< left << setw(14) <<"Balance";
    cout <<"\n=====\\n";
    cout <<"=====\\n";
}

```

```

vector<string> SplitString(stringline, stringDelimi = "#//#")
{
    vector<string>vString;
    short pos = 0;
    string sWord = "";

    while ((pos = line.find(Delimi)) != std::string::npos)
    {
        sWord =line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        line.erase(0, pos + Delimi.length());
    }
    if (line!="")
    {
        vString.push_back(line);
    }
    return vString;
}

stClient CovertLineToRecord(stringline)
{
    vector<string>vString;
    vString = SplitString(line);

    stClient Client;
    Client.Account_Number = vString[0];
    Client.PinCode = vString[1];
    Client.Full_Name = vString[2];
    Client.Phone = vString[3];
    Client.AccountBalance = stof(vString[4]);

    return Client;
}

vector<stClient> LoadClientDataFromFile(stringfilename)
{
    vector<stClient>vClient;

    fstream NewFile;
    NewFile.open(filename, ios::in);
    if (NewFile.is_open())
    {
        string line;
        stClient Client;

        while (getline(NewFile, line))
        {
            Client = CovertLineToRecord(line);
            vClient.push_back(Client);
        }
        NewFile.close();
    }
    return vClient;
}

```

```

void MidOfTable(stClientClient)
{
    cout <<"|" << left << setw(15) <<Client.Account_Number;
    cout <<"|" << left << setw(12) <<Client.PinCode;
    cout <<"|" << left << setw(30) <<Client.Full_Name;
    cout <<"|" << left << setw(14) <<Client.Phone;
    cout <<"|" << left << setw(14) <<Client.AccountBalance;
}

void PrintMidOfTable(vector<stClient>vClient)
{
    if (vClient.size() == 0)
    {
        cout <<"\t\t\tNo Clients Available In The System!";
    }
    else
    {
        for (stClient C : vClient)
        {
            MidOfTable(C);
            cout << endl;
        }
    }
}

void FooterOfTable()
{
    cout <<"===== ";
    cout <<"=====\n";
}

void ShowAllClientsScreen()
{
    vector<stClient>vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);

    HeaderOfTable(vClient);

    PrintMidOfTable(vClient);

    FooterOfTable();
}

//----- Find Client By Account Number -----//

string EnterAccountNumber()
{
    string AccountNumber;
    cout <<"Please enter Account Number: ";
    cin >> AccountNumber;
    return AccountNumber;
}

```



```

void ReadClientCard(stClient Client)
{
    cout << "\n\nThe Following are the client details:\n\n";

    cout << "Account Number : " << Client.Account_Number << endl;
    cout << "Pin Code       : " << Client.PinCode << endl;
    cout << "Full Name        : " << Client.Full_Name << endl;
    cout << "Phone           : " << Client.Phone << endl;
    cout << "Account Balance: " << Client.AccountBalance << endl;
}

bool FindClientByAccountNumber1(stClient&Client, string AccountNumber)
{
    vector<stClient> vClient;

    vClient = LoadClientDataFromFile(ClientsFileName);

    for (stClient C : vClient)
    {
        if (C.Account_Number == AccountNumber)
        {
            Client = C;
            return true;
        }
    }
    return false;
}

void ShowFindClientScreen()
{
    cout << "\n\n===== \n";
    cout << "\t\Find Client Screen\n";
    cout << "===== \n\n";

    string AccountNumber = EnterAccountNumber();
    stClient Client;

    if (FindClientByAccountNumber1(Client, AccountNumber))
    {
        ReadClientCard(Client);
    }
    else
    {
        cout << "\n\nClient with account number (" << AccountNumber << ") not found!\n";
    }
}

```

```

//----- Delete Client By Account Number -----//

bool FindClientByAccountNumber2(string AccountNumber, vector<stClient>&vClient,
stClient&Client)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            Client= C;
            returntrue;
        }
    }
    returnfalse;
}

bool MarkClientForDeleteByAccountNumber(string AccountNumber,
vector<stClient>&vClient)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            C.MarkForDelete = true;
            returntrue;
        }
    }
    returnfalse;
}

vector<stClient> SaveClientDataToFile2(stringfilename,
vector<stClient>&vClient)
{
    fstream NewFile;

    NewFile.open(filename, ios::out); // OverWrite

    string line;

    if (NewFile.is_open())
    {
        for (stClient C : vClient)
        {
            if (C.MarkForDelete == false)
            {
                // We only write records that are not marked for delete.

                line = ConvertRecordToLine(C);

                NewFile << line << endl;
            }
        }
        NewFile.close();
    }
    returnvClient;
}

```

```

bool DeleteClientByAccountNumber2(string AccountNumber,
vector<stClient>&vClient)
{
    stClient Client;

    char Answer = 'n';

    if (FindClientByAccountNumber2(AccountNumber, vClient, Client))
    {
        ReadClientCard(Client);

        cout <<"\nAre you sure you want delete this client ? (Y / N) : ";
        cin >> Answer;

        if (toupper(Answer) == 'Y')
        {
            MarkClientForDeleteByAccountNumber(AccountNumber, vClient);

            SaveClientDataToFile2(ClientsFileName, vClient);

            // ReFresh Clients
            vClient= LoadClientDataFromFile(ClientsFileName);

            cout <<"\n\nClient Deleted Successfully.\n";
            return true;
        }
    }
    else
    {
        cout <<"\n\nClient with account number ("<<AccountNumber<<") is not
found!\n";
        return false;
    }
}

void ShowDeleteClientScreen()
{
    cout <<"\n===== \n";
    cout <<"\t\tDelete Client Screen\n";
    cout <<"===== \n\n";

    string AccountNumber = EnterAccountNumber();

    vector<stClient> vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);

    DeleteClientByAccountNumber2(AccountNumber, vClient);
}

```

```
//----- Update Client By Account Number -----//
```

```
stClient ChangeClientRecord(string AccountNumber)
{
    stClient Client;
    Client.Account_Number = AccountNumber;

    cout << "Enter PinCode ? ";
    getline(cin >> ws, Client.PinCode);

    cout << "Enter Ful Name ? ";
    getline(cin, Client.Full_Name);

    cout << "Enter Phone ? ";
    getline(cin, Client.Phone);

    cout << "Enter Account Balance ? ";
    cin >> Client.AccountBalance;
    return Client;
}
```

```
bool UpdateClientByAccountNumber2(string AccountNumber, vector<stClient>&vClient)
{
    stClient Client;
    char Answer = 'n';
    if (FindClientByAccountNumber2(AccountNumber, vClient, Client))
    {
        ReadClientCard(Client);

        cout << "\nAre you sure you want update this client ? (Y / N) : ";
        cin >> Answer;

        if (toupper(Answer) == 'Y')
        {
            for (stClient& C : vClient)
            {
                if (C.Account_Number == AccountNumber)
                {
                    C = ChangeClientRecord(AccountNumber);
                    break;
                }
            }

            SaveClientDataToFile2(ClientsFileName, vClient);

            cout << "\n\nClient Updated Successfully.\n";
            return true;
        }
    }
    else
    {
        cout << "\n\nClient with account number (" << AccountNumber << ") is not found!\n";
        return false;
    }
}
```

```

void ShowUpdateClientScreen()
{
    system("cls");
    cout <<"\n=====\\n";
    cout <<"\t\tUpdate Client Info Screen\\n";
    cout <<"=====\\n\\n";

    string AccountNumber = EnterAccountNumber();

    vector<stClient> vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);
    UpdateClientByAccountNumber2(AccountNumber, vClient);
}

//----- Project 1 -----//

void GoBackToMainMenue()
{
    cout <<"\\n\\nPress any key to go back to Main Menue...";
    system("pause>0");
    ShowMainMenueScreen();
}

void ShowEndScreen()
{
    cout <<"\\n=====\\n";
    cout <<"\t\tProgram End, THANK YOU :-)"<< endl;
    cout <<"=====\\n";
}

enumMainMenueOption {
    eListClients = 1,
    eAddNewClient = 2,
    eDeleteClient = 3,
    eUpdateClient = 4,
    eFindClient = 5,
    eExit = 6
};

void PerformMainMenueOption(enMainMenueOptionMainMenueOption)
{
    switch (MainMenueOption)
    {
    caseenMainMenueOption::eListClients:
    {
        system("cls");
        ShowAllClientsScreen();
        GoBackToMainMenue();
        break;
    }
    caseenMainMenueOption::eAddNewClient:
    {
        system("cls");
        ShowAddNewClientsScreen();
        GoBackToMainMenue();
        break;
    }
}

```

```

case enMainMenuOption::eDeleteClient:
{
    system("cls");
    ShowDeleteClientScreen();
    GoBackToMainMenu();
    break;
}
case enMainMenuOption::eUpdateClient:
{
    system("cls");
    ShowUpdateClientScreen();
    GoBackToMainMenu();
    break;
}
case enMainMenuOption::eFindClient:
{
    system("cls");
    ShowFindClientScreen();
    GoBackToMainMenu();
    break;
}
case enMainMenuOption::eExit:
{
    system("cls");
    ShowEndScreen();
    break;
}
}

short ReadMainMenuOption()
{
    short Num = 0;
    cout << "Choose what do you want to do ? [1 to 6] ? ";
    cin >> Num;
    return Num;
}

void ShowMainMenuScreen()
{
    system("cls");
    cout << "=====\n";
    cout << "\t\t Main Menu Screen\n";
    cout << "=====\n";
    cout << "\t [1] Show Client List.\n";
    cout << "\t [2] Add New Client.\n";
    cout << "\t [3] Delete Client.\n";
    cout << "\t [4] Update Client Info.\n";
    cout << "\t [5] Find Client.\n";
    cout << "\t [6] Exit.\n";
    cout << "=====\n";
    PerformMainMenuOption((enMainMenuOption)ReadMainMenuOption());
}

int main()
{
    system("color f0");
    ShowMainMenuScreen();

    system("pause>0");
}

```

Project #02: Bank Extension : Requirements:

إضافات لبرامج البنك: متطلبات البرنامج :

<pre>===== Main Menue Screen ===== [1] Show Client List. [2] Add New Client. [3] Delete Client. [4] Update Client Info. [5] Find Client. [6] Transactions. [7] Exit. ===== Choose what do you want to do ? [1 to 7] ?</pre>	<pre>===== Main Menue Screen ===== [1] Show Client List. [2] Add New Client. [3] Delete Client. [4] Update Client Info. [5] Find Client. [6] Transactions. [7] Exit. ===== Choose what do you want to do ? [1 to 7] ? 6_</pre>
<pre>===== Transactions Menu Screen ===== [1] Deposit. [2] Withdraw. [3] Total Balances. [4] Main Menu. ===== Choose what do you want to do ? [1 to 4] ? 1_</pre>	<pre>===== Deposit Screen ===== Please enter Account Number: A152 The Following are the client details: Account Number : A152 Pin Code : 789 Full Name : Shahd Ahmed Phone : 0122222222 Account Balance: 300 Please enter deposit amount: 200 Are you sure you want perfrom this transaction: (Y/N) ? y Done Successfully, New balance is: 500 Press any key to go back to Transactions Menue..._</pre>
<pre>===== Transactions Menu Screen ===== [1] Deposit. [2] Withdraw. [3] Total Balances. [4] Main Menu. ===== Choose what do you want to do ? [1 to 4] ? 2_</pre>	<pre>===== Withdraw Screen ===== Please enter Account Number: A150 The Following are the client details: Account Number : A150 Pin Code : 123 Full Name : Ahmed ElSayed Phone : 01478963258 Account Balance: 2000 Please enter withdraw amount: 300 Are you sure you want perfrom this transaction: (Y/N) ? y Done Successfully, New balance is: 1700 Press any key to go back to Transactions Menue..._</pre>
<pre>===== Transactions Menu Screen ===== [1] Deposit. [2] Withdraw. [3] Total Balances. [4] Main Menu. ===== Choose what do you want to do ? [1 to 4] ? 3_</pre>	<pre>===== Balance List (3) Client(s) ===== Account Number Client Name Balance ===== A150 Ahmed ElSayed 1700 A151 Mahmoud Ahmed 200 A152 Shahd Ahmed 500 ===== Total Balance = 2400 Press any key to go back to Transactions Menue...</pre>
<pre>===== Transactions Menu Screen ===== [1] Deposit. [2] Withdraw. [3] Total Balances. [4] Main Menu. ===== Choose what do you want to do ? [1 to 4] ? 4</pre>	<pre>===== Main Menue Screen ===== [1] Show Client List. [2] Add New Client. [3] Delete Client. [4] Update Client Info. [5] Find Client. [6] Transactions. [7] Exit. ===== Choose what do you want to do ? [1 to 7] ? _</pre>

Solving:

```
#include<iostream>
#include<string>
#include<fstream>
#include<vector>
#include<iomanip>

using namespace std;

const string ClientsFileName = "Clients.txt";

void ShowMainMenuScreen();
void ShowTransactionsMenuScreen();

struct stClient {
    string Account_Number;
    string PinCode;
    string Full_Name;
    string Phone;
    float AccountBalance;

    bool MarkForDelete = false;
};

stClient CovertLineToRecord(string line);

//----- Add Clients To File -----//

bool ClientExistsByAccountNumber(string AccountNumber, string filename)
{
    vector<stClient> vClient;

    fstream NewFile;

    NewFile.open(filename, ios::in);

    if (NewFile.is_open())
    {
        string line;
        stClient Client;

        while (getline(NewFile, line))
        {
            Client = CovertLineToRecord(line);
            if (Client.Account_Number == AccountNumber)
            {
                NewFile.close(); // مهم جداً ينقل الملف هنا
                return true;
            }
            vClient.push_back(Client);
        }
        NewFile.close();
    }
    return false;
}
```



```

stClient ReadClientData()
{
    stClient Client;

    cout <<"Enter Account Number ? ";
    // usage for std::ws will extract all the whitespace character
    // سيؤدي استخدام std::ws إلى استخراج كل أحرف المسافة البيضاء
    getline(cin >> ws, Client.Account_Number);

    // للتحقق إذا كان الـ Account Number موجود أصلا
    while (ClientExistsByAccountNumber(Client.Account_Number,
ClientsFileName))
    {
        cout <<"Client with ["<< Client.Account_Number <<"] already exists,
Enter Another Account Number: ";
        getline(cin >> ws, Client.Account_Number);
    }

    cout <<"Enter PinCode ? ";
    getline(cin, Client.PinCode);

    cout <<"Enter Ful Name ? ";
    getline(cin, Client.Full_Name);

    cout <<"Enter Phone ? ";
    getline(cin, Client.Phone);

    cout <<"Enter Account Balance ? ";
    cin >> Client.AccountBalance;

    return Client;
}

string ConvertRecordToLine(stClientsClient, stringDelimi = "#//#")
{
    string line = "";

    line +=sClient.Account_Number +Delimi;
    line +=sClient.PinCode +Delimi;
    line +=sClient.Full_Name +Delimi;
    line +=sClient.Phone +Delimi;
    line += to_string(sClient.AccountBalance);

    return line;
}

void AddClientDataToFile(stringfilename, stringLine)
{
    fstream to_File;
    to_File.open(filename, ios::out | ios::app);
    if (to_File.is_open())
    {
        to_File <<Line<< endl;

        to_File.close();
    }
}

```

```

void AddNewClient()
{
    stClient Client;
    Client = ReadClientData();

    AddClientDataToFile(ClientsFileName, ConvertRecordToLine(Client));
}

void AddClients()
{
    char AddMore = 'Y';
    do
    {
        cout << "\nAdding New Client:\n\n";

        AddNewClient();

        cout << "\nClient Added Successfully, do you want to add more clients?
(Y/N): ";
        cin >> AddMore;

    } while (toupper(AddMore) == 'Y');
}

void ShowAddNewClientsScreen()
{
    system("cls");

    cout << "\n===== \n";
    cout << "\t\tAdd New Clients Screen\n";
    cout << "===== \n";

    AddClients();
}

//----- Show All Clients -----//

void HeaderOfTable(vector<stClient>vClient)
{
    cout << "\n\t\t\t\tClient List (" << vClient.size() << ") Client(s)\n";
    cout << "\n===== ";
    cout << "===== \n";
    cout << " | " << left << setw(15) << "Account Number";
    cout << " | " << left << setw(12) << "Pic Code";
    cout << " | " << left << setw(30) << "Client Name";
    cout << " | " << left << setw(14) << "Phone";
    cout << " | " << left << setw(14) << "Balance";
    cout << "\n===== ";
    cout << "===== \n";
}

```

```

vector<string> SplitString(stringline, stringDelimi = "#//#")
{
    vector<string>vString;
    short pos = 0;
    string sWord = "";

    while ((pos = line.find(Delimi)) != std::string::npos)
    {
        sWord =line.substr(0, pos);
        if (sWord != "")
        {
            vString.push_back(sWord);
        }
        line.erase(0, pos + Delimi.length());
    }
    if (line!="")
    {
        vString.push_back(line);
    }
    return vString;
}

stClient CovertLineToRecord(stringline)
{
    vector<string>vString;
    vString = SplitString(line);
    stClient Client;
    Client.Account_Number = vString[0];
    Client.PinCode = vString[1];
    Client.Full_Name = vString[2];
    Client.Phone = vString[3];
    Client.AccountBalance = stof(vString[4]);
    return Client;
}

vector<stClient> LoadClientDataFromFile(stringfilename)
{
    vector<stClient>vClient;

    fstream NewFile;

    NewFile.open(filename, ios::in);

    if (NewFile.is_open())
    {
        string line;
        stClient Client;

        while (getline(NewFile, line))
        {
            Client = CovertLineToRecord(line);
            vClient.push_back(Client);
        }
        NewFile.close();
    }
    return vClient;
}

```

```

void MidOfTable(stClientClient)
{
    cout <<"|"<< left << setw(15) <<Client.Account_Number;
    cout <<"|"<< left << setw(12) <<Client.PinCode;
    cout <<"|"<< left << setw(30) <<Client.Full_Name;
    cout <<"|"<< left << setw(14) <<Client.Phone;
    cout <<"|"<< left << setw(14) <<Client.AccountBalance;
}

void PrintMidOfTable(vector<stClient>vClient)
{
    if (vClient.size() == 0)
    {
        cout <<"\t\t\tNo Clients Available In The System!";
    }
    else
    {
        for (stClient C : vClient)
        {
            MidOfTable(C);
            cout << endl;
        }
    }
}

void FooterOfTable()
{
    cout <<"===== ";
    cout <<"=====\n";
}

void ShowAllClientsScreen()
{
    vector<stClient>vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);
    HeaderOfTable(vClient);
    PrintMidOfTable(vClient);
    FooterOfTable();
}
//----- Find Client By Account Number -----//
string EnterAccountNumber()
{
    string AccountNumber;
    cout <<"Please enter Account Number: ";
    cin >> AccountNumber;
    return AccountNumber;
}

void ReadClientCard(stClientClient)
{
    cout <<"\nThe Following are the client details:\n\n";
    cout <<"Account Number : "<<Client.Account_Number << endl;
    cout <<"Pin Code       : "<<Client.PinCode << endl;
    cout <<"Full Name        : "<<Client.Full_Name << endl;
    cout <<"Phone           : "<<Client.Phone << endl;
    cout <<"Account Balance: "<<Client.AccountBalance << endl;
}

```

```

bool FindClientByAccountNumber1(stClient&Client, stringAccountNumber)
{
    vector<stClient>vClient;

    vClient = LoadClientDataFromFile(ClientsFileName);

    for (stClient C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            Client= C;
            returntrue;
        }
    }
    returnfalse;
}

void ShowFindClientScreen()
{
    cout <<"\n===== \n";
    cout <<"\t\Find Client Screen\n";
    cout <<"===== \n\n";

    string AccountNumber = EnterAccountNumber();
    stClient Client;

    if (FindClientByAccountNumber1(Client, AccountNumber))
    {
        ReadClientCard(Client);
    }
    else
    {
        cout <<"\n\nClient with account number ("<< AccountNumber <<") not
found!\n";
    }
}

//----- Delete Client By Account Number -----
-----//

bool FindClientByAccountNumber2(stringAccountNumber, vector<stClient>&vClient,
stClient&Client)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            Client= C;
            returntrue;
        }
    }
    returnfalse;
}

```

```

bool MarkClientForDeleteByAccountNumber(string AccountNumber,
vector<stClient>&vClient)
{
    for (stClient& C : vClient)
    {
        if (C.Account_Number ==AccountNumber)
        {
            C.MarkForDelete = true;
            returntrue;
        }
    }
    returnfalse;
}

vector<stClient> SaveClientDataToFile2(stringfilename,
vector<stClient>&vClient)
{
    fstream NewFile;

    NewFile.open(filename, ios::out); // OverWrite

    string line;

    if (NewFile.is_open())
    {
        for (stClient C : vClient)
        {
            if (C.MarkForDelete == false)
            {
                // We only write records that are not marked for delete.

                line = ConvertRecordToLine(C);

                NewFile << line << endl;
            }
        }
        NewFile.close();
    }
    returnvClient;
}

bool DeleteClientByAccountNumber2(stringAccountNumber,
vector<stClient>&vClient)
{
    stClient Client;

    char Answer = 'n';

    if (FindClientByAccountNumber2(AccountNumber, vClient, Client))
    {
        ReadClientCard(Client);

        cout <<"\nAre you sure you want delete this client ? (Y / N) : ";
        cin >> Answer;
    }
}

```

```

        if (toupper(Answer) == 'Y')
        {
            MarkClientForDeleteByAccountNumber(AccountNumber, vClient);

            SaveClientDataToFile2(ClientsFileName, vClient);

            // ReFresh Clients
            vClient= LoadClientDataFromFile(ClientsFileName);

            cout <<"\n\nClient Deleted Successfully.\n";
            return true;
        }
    }
    else
    {
        cout <<"\n\nClient with account number ("<<AccountNumber<<") is not
found!\n";
        return false;
    }
}

void ShowDeleteClientScreen()
{
    cout <<"\n===== \n";
    cout <<"\t\tDelete Client Screen\n";
    cout <<"===== \n\n";

    string AccountNumber = EnterAccountNumber();

    vector<stClient> vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);

    DeleteClientByAccountNumber2(AccountNumber, vClient);
}

//----- Update Client By Account Number -----//

stClient ChangeClientRecord(string AccountNumber)
{
    stClient Client;

    Client.Account_Number = AccountNumber;

    cout <<"Enter PinCode ? ";
    getline(cin >> ws, Client.PinCode);

    cout <<"Enter Ful Name ? ";
    getline(cin, Client.Full_Name);

    cout <<"Enter Phone ? ";
    getline(cin, Client.Phone);

    cout <<"Enter Account Balance ? ";
    cin >> Client.AccountBalance;

    return Client;
}

```

```

bool UpdateClientByAccountNumber2(string AccountNumber,
vector<stClient>&vClient)
{
    stClient Client;

    char Answer = 'n';

    if (FindClientByAccountNumber2(AccountNumber, vClient, Client))
    {
        ReadClientCard(Client);

        cout <<"\nAre you sure you want update this client ? (Y / N) : ";
        cin >> Answer;

        if (toupper(Answer) == 'Y')
        {
            for (stClient& C : vClient)
            {
                if (C.Account_Number ==AccountNumber)
                {
                    C = ChangeClientRecord(AccountNumber);
                    break;
                }
            }

            SaveClientDataToFile2(ClientsFileName, vClient);

            cout <<"\n\nClient Updated Successfully.\n";
            return true;
        }
    }
    else
    {
        cout <<"\n\nClient with account number ("<<AccountNumber<<") is not
found!\n";
        return false;
    }
}

void ShowUpdateClientScreen()
{
    system("cls");

    cout <<"\n===== \n";
    cout <<"\t\tUpdate Client Info Screen\n";
    cout <<"===== \n\n";

    string AccountNumber = EnterAccountNumber();

    vector<stClient> vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);

    UpdateClientByAccountNumber2(AccountNumber, vClient);
}

```



```
//----- Project 1 -----//
```

```
bool BalanceByAccountNumber(string AccountNumber, float Amount,
vector<stClient> vClient)
{
    char Answer = 'Y';

    cout << "\nAre you sure you want perfrom this transaction: (Y/N) ? ";
    cin >> Answer;

    if (toupper(Answer) == 'Y')
    {
        for (stClient& C : vClient)
        {
            if (C.Account_Number == AccountNumber)
            {
                C.AccountBalance += Amount;

                SaveClientDataToFile2(ClientsFileName, vClient);

                cout << "\n\nDone Successfully, New balance is: "
                     << C.AccountBalance << endl;

                return true;
            }
        }
        return false;
    }
}

void DepositSrceen()
{
    float DepositAmount = 0; // DepositAmount = قيمة الإيداع
    char Answer = 'Y';

    vector<stClient> vClient;

    string AccountNumber = EnterAccountNumber();

    vClient = LoadClientDataFromFile(ClientsFileName);

    stClient Client;

    while (!FindClientByAccountNumber2(AccountNumber, vClient, Client))
    {
        cout << "\nClient with [" << AccountNumber << "] does not exist!\n\n";
        AccountNumber = EnterAccountNumber();
    }

    ReadClientCard(Client);

    cout << "\nPlease enter deposit amount: ";
    cin >> DepositAmount;

    BalanceByAccountNumber(AccountNumber, DepositAmount, vClient);
}
```

```

void WithdrawScreen()
{
    float WithdrawAmount = 0;           // WithdrawAmount = قيمة السحب
    char Answer = 'Y';

    vector<stClient>vClient;

    string AccountNumber = EnterAccountNumber();

    vClient = LoadClientDataFromFile(ClientsFileName);

    stClient Client;

    while (!FindClientByAccountNumber2(AccountNumber, vClient, Client))
    {
        cout <<"\nClient with ["<< AccountNumber
                <<"] does not exist!\n\n";
        AccountNumber = EnterAccountNumber();
    }

    ReadClientCard(Client);

    cout <<"\nPlease enter withdraw amount: ";
    cin >> WithdrawAmount;

    while (WithdrawAmount > Client.AccountBalance)
    {
        cout <<"Amount exceeds the balance, you can withdraw up to : "
                << Client.AccountBalance << endl;
        cout <<"\nPlease enter withdraw amount: ";
        cin >> WithdrawAmount;
    }

    BalanceByAccountNumber(AccountNumber, WithdrawAmount * -1, vClient);
    // الضرب في سالب واحد يعطينا النتيجة سالب
}

void HeaderOfBalanceTable(vector<stClient>vClient)
{
    cout <<"\n\t\t\t\t\tBalance List ("<<vClient.size() <<") Client(s)\n";
    cout <<"\n===== ";
    cout <<"===== \n";
    cout <<"|"<< left << setw(20) <<"Account Number";
    cout <<"|"<< left << setw(30) <<"Client Name";
    cout <<"|"<< left << setw(20) <<"Balance";
    cout <<"\n===== ";
    cout <<"===== \n";
}

void MidOfBalanceTable(stClientClient)
{
    cout <<"|"<< left << setw(20) <<Client.Account_Number;
    cout <<"|"<< left << setw(30) <<Client.Full_Name;
    cout <<"|"<< left << setw(20) <<Client.AccountBalance;
}

```

```

void PrintMidOfBalanceTable(vector<stClient>vClient)
{
    if (vClient.size() == 0)
    {
        cout <<"\t\t\tNo Clients Available In The System!";
    }
    else
    {
        for (stClient C : vClient)
        {
            MidOfBalanceTable(C);
            cout << endl;
        }
    }
}

void FooterOfBalanceTable()
{
    cout <<"\n===== ";
    cout <<"=====\n";
}

float AllBalanceCounter(vector<stClient>vClient)
{
    stClient Client;
    float BalanceCounter = 0;
    for (stClient C : vClient)
    {
        BalanceCounter += C.AccountBalance;
    }
    return BalanceCounter;
}

void ShowBalanceScreen()
{
    vector<stClient>vClient;
    vClient = LoadClientDataFromFile(ClientsFileName);

    float BalanceCounter = AllBalanceCounter(vClient);

    HeaderOfBalanceTable(vClient);
    PrintMidOfBalanceTable(vClient);
    FooterOfBalanceTable();
    cout <<"\n\t\t\t\t\tTotal Balance = "<< BalanceCounter << endl;
}

void ShowDepositScreen()
{
    system("cls");

    cout <<"\n===== \n";
    cout <<"\t\tDeposit Screen\n";
    cout <<"=====\n\n";

    DepositSrceen();
}

```

```

void ShowWithdrawScreen()
{
    system("cls");

    cout <<"\n=====\\n";
    cout <<"\t\tWithdraw Screen\\n";
    cout <<"=====\\n\\n";

    WithdrawScreen();
}

void ShowTotalBalancesScreen()
{
    system("cls");

    ShowBalanceScreen();
}

void GoBackToMainMenu()
{
    cout <<"\\n\\nPress any key to go back to Main Menu...";
    system("pause>0");
    ShowMainMenuScreen();
}

void GoBackToTransactionsMenuScreen()
{
    cout <<"\\n\\nPress any key to go back to Transactions Menu...";
    system("pause>0");
    ShowTransactionsMenuScreen();
}

void ShowEndScreen()
{
    cout <<"\\n=====\\n";
    cout <<"\t\tProgram End, THANK YOU :-)"<< endl;
    cout <<"=====\\n";
}

enum MainMenuOption {
    eListClients = 1,
    eAddNewClient = 2,
    eDeleteClient = 3,
    eUpdateClient = 4,
    eFindClient = 5,
    eTransactions = 6,
    eExit = 7
};

enum TransactionsMenuOption {

    eDeposit = 1,
    eWithdraw = 2,
    eTotalBalances = 3,
    eMainMenu = 4,
};

```

```

void PerformTransactionsMenuOption(enTransactionsMenuOptionTransactionsMenuOption)
{
    switch (TransactionsMenuOption)
    {
    case enTransactionsMenuOption::eDeposit:
    {
        system("cls");
        ShowDepositScreen();
        GoBackToTransactionsMenuScreen();
        break;
    }
    case enTransactionsMenuOption::eWithdraw:
    {
        system("cls");
        ShowWithdrawScreen();
        GoBackToTransactionsMenuScreen();
        break;
    }
    case enTransactionsMenuOption::eTotalBalances:
    {
        system("cls");
        ShowTotalBalancesScreen();
        GoBackToTransactionsMenuScreen();
        break;
    }
    case enTransactionsMenuOption::eMainMenu:
    {
        ShowMainMenueScreen();
    }
    }
}

```

```

void PerformMainMenueOption(enMainMenueOptionMainMenueOption)
{
    switch (MainMenueOption)
    {
    case enMainMenueOption::eListClients:
    {
        system("cls");
        ShowAllClientsScreen();
        GoBackToMainMenue();
        break;
    }
    case enMainMenueOption::eAddNewClient:
    {
        system("cls");
        ShowAddNewClientsScreen();
        GoBackToMainMenue();
        break;
    }
    case enMainMenueOption::eDeleteClient:
    {
        system("cls");
        ShowDeleteClientScreen();
        GoBackToMainMenue();
        break;
    }
    }
}

```

```

        case enMainMenueOption::eUpdateClient:
        {
            system("cls");
            ShowUpdateClientScreen();
            GoBackToMainMenue();
            break;
        }
        case enMainMenueOption::eFindClient:
        {
            system("cls");
            ShowFindClientScreen();
            GoBackToMainMenue();
            break;
        }
        case enMainMenueOption::eTransactions:
        {
            system("cls");
            ShowTransactionsMenuScreen();
            break;
        }
        case enMainMenueOption::eExit:
        {
            system("cls");
            ShowEndScreen();
            break;
        }
    }
}

short ReadMainMenueOption()
{
    short Num = 0;
    cout << "Choose what do you want to do ? [1 to 7] ? ";
    cin >> Num;
    return Num;
}

short ReadTransactionsMenuOption()
{
    short Num = 0;
    cout << "Choose what do you want to do ? [1 to 4] ? ";
    cin >> Num;
    return Num;
}

void ShowTransactionsMenuScreen()
{
    system("cls");
    cout << "=====\n";
    cout << "\t\tTransactions Menu Screen\n";
    cout << "=====\n";
    cout << "\t [1] Deposit.\n";
    cout << "\t [2] Withdraw.\n";
    cout << "\t [3] Total Balances.\n";
    cout << "\t [4] Main Menu.\n";

    cout << "=====\n";

    PerformTransactionsMenuOption((enTransactionsMenuOption)ReadTransactionsMenuOption());
}

```

```

void ShowMainMenueScreen()
{
    system("cls");

    cout <<"=====\\n";
    cout <<"\\t\\t Main Menue Screen\\n";
    cout <<"=====\\n";
    cout <<"\\t [1] Show Client List.\\n";
    cout <<"\\t [2] Add New Client.\\n";
    cout <<"\\t [3] Delete Client.\\n";
    cout <<"\\t [4] Update Client Info.\\n";
    cout <<"\\t [5] Find Client.\\n";
    cout <<"\\t [6] Transactions.\\n";
    cout <<"\\t [7] Exit.\\n";
    cout <<"=====\\n";

    PerformMainMenueOption((enMainMenueOption)ReadMainMenueOption());
}

int main()
{
    system("color f0");
    ShowMainMenueScreen();

    system("pause>0");
}

```

Problem #01 - 3x3 Random Matrix	1
Problem #02 - Sum Each Row in Matrix	2
Problem #03 - Sum Each Row in Matrix in Array	3
Problem #04 - Sum Each Col in Matrix	5
Problem #05 - Sum Each Col in Matrix in another Array	7
Problem #06 - 3x3 Ordered Matrix	9
Problem #07 - Transpose Matrix	10
Problem #08 - Multiply Two Matrices	12
Problem #09 - Print Middle Row and Col of Matrix	14
Problem #10 - Sum of Matrix	16
Problem #11 - Check Matrices Equality	18
Problem #12 - Check Typical Matrices	20
Problem #13 - Check Identity Matrix	22
Problem #14 - Check Scalar Matrix	23
Problem #15 - Count Number in Matrix	25
Problem #16 - Check Sparse Matrix	27
Problem #17 - Number Exists in Matrix	28
Problem #18 - Intersected Number in Matrices	31
Problem #19 - Min-Max Number in Matrix	33
Problem #20 - Palindrome Matrix	34
Problem #21 - Fibonacci Series	36
Problem #22 - Fibonacci Series with Recursion	36
Problem #23 - Print First Letter Of Each Word	38
Problem #24 - Upper First Letter Of Each Word	39
Problem #25 - Lower First Letter Of Each Word	40
Problem #26 - Upper-Lower All Letter Of a String	41
Problem #27 - Invert Character Case	42
Problem #28 - Invert All Letters Case	43
Problem #29 - Count Small-Capital Letters	43
Problem #30 - Count Letters	45
Problem #31 - Count Letters (Match Case)	46
Problem #32 - Is Vowel	47
Problem #33 - Count Vowel	48
Problem #34 - Print All Vowels In String	49
Problem #35 - Print Each Word In String	50
Problem #36 - Count Each Word In String	51
Problem #37 - Split String	52
Problem #38 - Trim Left, Trim Right, Trim	53
Problem #39 - Join String	54
Problem #40 - Join String (Overloading)	56
Problem #41 - Reverse Words	57
Problem #42 - Replace Words	58
Problem #43 - Replace Words (Custom)	59
Problem #44 - Remove Punctuations	61
Problem #45 - Convert Record To Line	61
Problem #46 - Convert Line Date To Record	63
Problem #47 - Add Clients To File	65
Problem #48 - Show All Clients	67
Problem #49 - Find Client By Account Number	69
Problem #50 - Delete Client By Account Number	72
Problem #51 - Update Client By Account Number	76
Project #01 - Bank 1 - Requirements	81
Project #01 - Bank 1 - Solution	82
Project #02 - Bank Extension	94
Project #02 - Bank Extension - Solution	95