C# and .NET Framework – Lab sheet 03

1. switch...case Statements

Type the following program into SharpDevelop, and answer the questions

```
using System ;
class Coffee {
        static void Main () {
                int add_up ;
                double cost = 35.0;
                 Console .WriteLine ("You .can .add .into .your.coffee .with .either: ");
                 Console .WriteLine (".1-Milk ,.2-Sugar,.3-M ilk + Sugar,.4-Nothing ");
                 Console .Write ("Your.choice .is: .");
                 add_up = int.Parse (Console .ReadLine ());
                 switch (add_up ) {
                         case 1 : cost += 9.00;
                         break;
                         case 2 : cost += 5.00;
                         break;
                         case 3 : cost += 12.00;
                        break;
                        default: cost += 0.0;
                         break;
                 }
                Console .W riteLine ("Your.coffee cost.is.{0}.baht.",cost);
                 Console .W riteLine ("Thank.you.");
         }
 }
```

• What are the input values of add up to make the value of cost less than 45?

1 and 2

• If the user enters 1 to the program, what will be the resulting value of cost?

44

• If the user enters 4 to the program, what will be the results displayed?

Your.coffee cost.is.35.baht.
Thank.you.

• If the user enters -1 to the program, what will be the results displayed?

Your.coffee cost.is.35.baht.
Thank.you.

Modify the switch statement and the last two display statements so that the program can handle invalid inputs from the user. If the user enters an invalid input, the program should set cost to zero and display to the user that “Your coffee is canceled.”

```
static void Main(string[] args)
{
 int add_up;
 double cost = 35.0;
 Console.WriteLine("You .can .add .into .your.coffee .with .either: ");
 Console.WriteLine(".1-Milk ,.2-Sugar,.3-Milk + Sugar,.4-Nothing ");
 Console.Write("Your.choice .is: .");
 add_up = int.Parse(Console.ReadLine());
 switch (add_up)
 {
 case 1:
 cost += 9.00;
 break;
 case 2:
 cost += 5.00;
 break;
 case 3:
 cost += 12.00;
 break;
 case 4:
 cost += 0.0;
 break;
 default:
 Console.WriteLine("Your coffee is canceled.");
 cost = -1;
 break;
 }
 if (cost>0)
 {
 Console.WriteLine("Your.coffee cost.is.{0}.baht.", cost);
 Console.WriteLine("Thank.you.");
 }
}
```

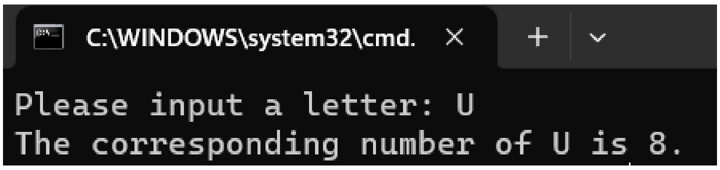
Programming Exercises

a) Task 2.1: Which number am I pressing?

Write a C# program to translate a letter to a number according to a given mapping table (from a cell-phone’s dial pad).

Letters	Number
A B C	2
D E F	3
G H I	4
J K L	5
M N O	6
P Q R S	7
T U V	8
W X Y Z	9

```
static void Main(string[] args)
 {
 Console.Write("Please input a letter: ");
 var input = Console.ReadKey();
 int num = 0;
 switch (input.KeyChar)
 {
 case 'A': case 'a':
 case 'B': case 'b':
 case 'C': case 'c':
 num = 2; break;
 case 'D': case 'd':
 case 'E': case 'e':
 case 'F': case 'f':
 num = 3; break;
 case 'G': case 'g':
 case 'H': case 'h':
 case 'I': case 'i':
 num = 4; break;
 case 'J': case 'j':
 case 'K': case 'k':
 case 'L': case 'l':
 num = 5; break;
 case 'M': case 'm':
 case 'N': case 'n':
 case 'O': case 'o':
 num = 6; break;
 case 'P': case 'p':
 case 'Q': case 'q':
 case 'R': case 'r':
 case 'S': case 's':
 num = 7; break;
 case 'T': case 't':
 case 'U': case 'u':
 case 'V': case 'v':
 num = 8; break;
 case 'W': case 'w':
 case 'X': case 'x':
 case 'Y': case 'y':
 case 'Z': case 'z':
 num = 9; break;
 default: num = -1; break;
 }
 Console.WriteLine($"\r\nThe corresponding number of {input.KeyChar} is {num}.");
        }
```



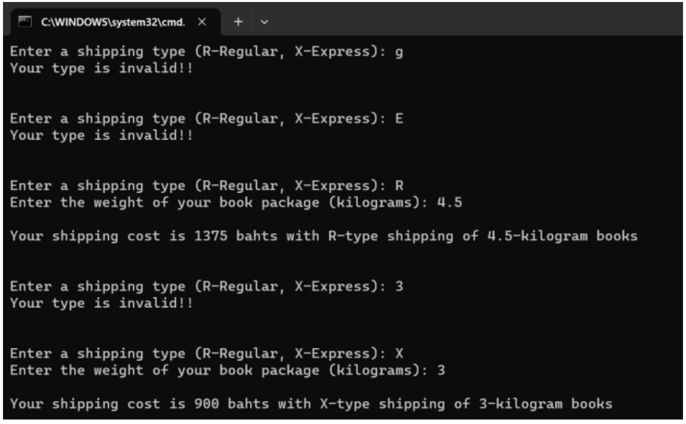
```
Please input a letter: U
The corresponding number of U is 8.
```

Book shipping

Write a C# program to compute the cost of book shipping. The shipping cost is computed according to the shipping type and the package weight. The shipping rate is show in the following table

Shipping types	Weight	Rate (bahts/gram)
Regular	First 2000	0.25
Regular	After 2000	0.35
Express	Use the same rate as regular + 50 bahts fee	

```
static void Main(string[] args)
{
 Console.Write("Enter a shipping type (R-Regular, X-Express): ");
 var type = Console.ReadKey().KeyChar;
 var cost = 0.0;
 if (type == 'R' || type == 'X')
 {
 Console.Write("\r\nEnter the weight of your book package (kilograms): ");
 var weight = Convert.ToDouble(Console.ReadLine()) * 1000;
 if (weight <= 2000)
 {
 cost = weight * 0.25;
 }
 else
 {
 const double fixedRate = 2000 * 0.25;
 cost = fixedRate + (weight - 2000) * 0.35;
 }
 if (type == 'X')
 {
 cost += 50;
 }
 Console.WriteLine($"\r\nYour shipping cost is {cost} bahts with {type}-type shipping of {weight/1000}-
kilogram books");
 }
 else
 {
 Console.WriteLine("\r\nYour type is invalid!!");
 }
}
```



```
Enter a shipping type (R-Regular, X-Express): g
Your type is invalid!!


Enter a shipping type (R-Regular, X-Express): E
Your type is invalid!!


Enter a shipping type (R-Regular, X-Express): R
Enter the weight of your book package (kilograms): 4.5

Your shipping cost is 1375 bahts with R-type shipping of 4.5-kilogram books


Enter a shipping type (R-Regular, X-Express): 3
Your type is invalid!!


Enter a shipping type (R-Regular, X-Express): X
Enter the weight of your book package (kilograms): 3

Your shipping cost is 900 bahts with X-type shipping of 3-kilogram books
```

2. while Loops

Exercise 1.1: Modify the program in Example 1.1 so that it will display the numbers from *N* to 1 instead. Determine what should be put in the blanks marked (a)–(c).

```
using System ;
class While2 {
        static void Main () {
                int N;
                int i;
                Console . Write(" Please. input.N:.");
                N = int . Parse( Console . ReadLine ());
                ___ (a) ___;
                while ( ___ (b) ___) {
                        Console . WriteLine(i);
                        ________ ( c) _______ ;
                }
        }
}
```

	Expression/Statement
(a)	i=N
(b)	i>0
(c)	i--

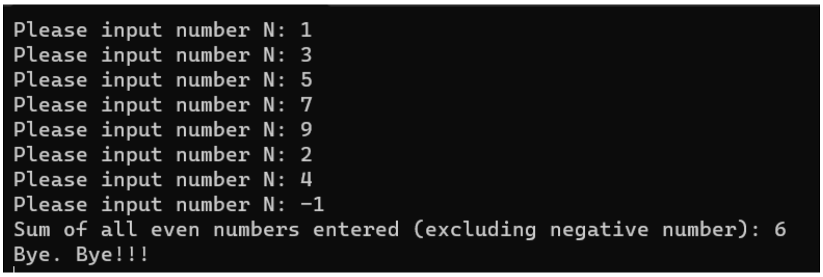
Exercise 1.2: The incomplete program below is a modification of the program in Example 1.3. In addition, this program will compute the summation of all the numbers entered, except the last one (negative number) which is used as a sen- tinel. Determine what should be put in the blanks marked (a)–(c).

```
using System ;
   class While5 {
      static void Main () {
         int N = 0 , sum ;
         ____( a) ____ ;
         while (N >= 0) {//Exit while loop when          N is negative
               Console . Write(" Please. input. N:.");
              N = int . Parse (Console . ReadLine ());
            _____________(b) __________ ;
         }
         _____________(c) __________ ;
         Console .WriteLine("Bye .Bye !!!");
      }
```

Blank	Expression/Statement
___(a)____	int input;
___(b)____	sum += input;
___(c)____	Console.WriteLine ("Sum =" + sum);

Exercise 1.3: Modify the previous program so that the summation is calculated on *even* numbers only. Again, the negative value used as a sentinel must not be included in the calculation. Write your complete code in the box provided.

```
int sum = 0;
int input = 0;
while (input >= 0)
{
 Console.Write("Please input number N: ");
 if (input % 2 == 0)
 {
 sum += input;
 }
 input = int.Parse(Console.ReadLine());
}
Console.WriteLine("Sum of all even numbers entered (excluding negative number): " + sum);
Console.WriteLine("Bye. Bye!!!");
```

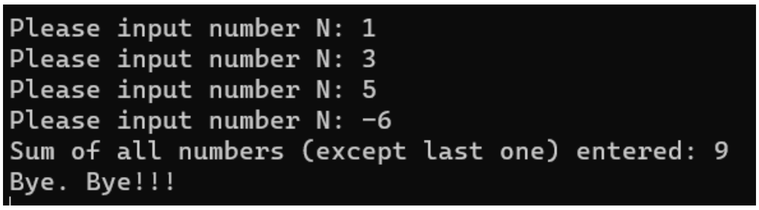


```
Please input number N: 1
Please input number N: 3
Please input number N: 5
Please input number N: 7
Please input number N: 9
Please input number N: 2
Please input number N: 4
Please input number N: -1
Sum of all even numbers entered (excluding negative number): 6
Bye. Bye!!!
```

3. do-while Loops

Exercise 2.1: Modify the program in Example 2.1 to compute the summation of all the numbers entered except the last one (similar to Exercise 1.2). However, your program must use a do-while loop. Write down your C# code in the box below

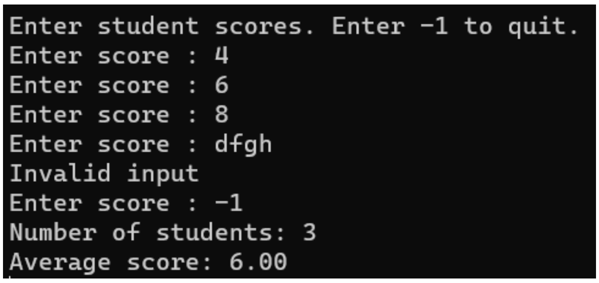
```
int sum = 0;
int input;

do
{
 Console.Write("Please input number N: ");
 input = int.Parse(Console.ReadLine());

 if (input >= 0)
 {
 sum += input;
 }
} while (input >= 0);

Console.WriteLine("Sum of all numbers (except last one) entered: " + sum);
Console.WriteLine("Bye. Bye!!!");
```



```
Please input number N: 1
Please input number N: 3
Please input number N: 5
Please input number N: -6
Sum of all numbers (except last one) entered: 9
Bye. Bye!!!
```

Task 3.1: Write a C# program to read student scores from the user until the value -1 is given. Then report the number of students and the average score with two decimal places

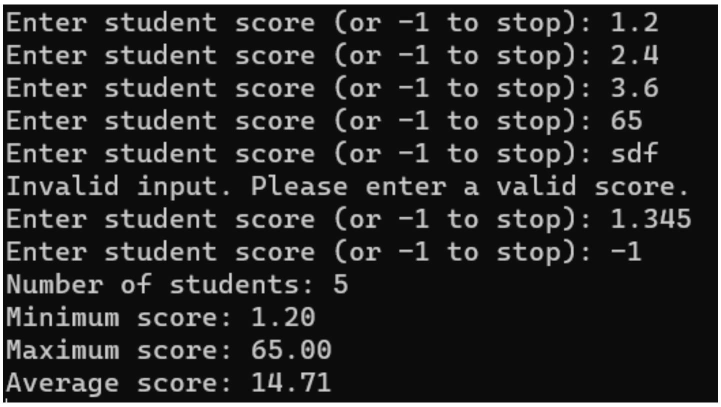
```
int count = 0;
 double totalScore = 0;

 Console.WriteLine("Enter student scores. Enter -1 to quit.");
 double input =0;
 while (input != -1)
 {
 Console.Write("Enter score : ");
 if (double.TryParse(Console.ReadLine(), out input))
 {
 if (input == -1)
 break;

 totalScore += input;
 count++;
 }
 else
 {
 Console.WriteLine("Invalid input");
 }
 }

 if (count > 0)
 {
 Console.WriteLine($"Number of students: {count}");
 Console.WriteLine($"Average score: {totalScore / count:F2}");
 }
 else
 {
 Console.WriteLine("No valid scores entered.");
 }
```



```
Enter student scores. Enter -1 to quit.
Enter score : 4
Enter score : 6
Enter score : 8
Enter score : dfgh
Invalid input
Enter score : -1
Number of students: 3
Average score: 6.00
```

Task 3.2: Modify the program you wrote in Task 3.1 so that it also reports the minimum score and the maximum score (also with two decimal places).

```
 int count = 0;
 double totalScore = 0;
 double minScore = double.MaxValue;
 double maxScore = double.MinValue;

 while (true)
 {
 Console.Write("Enter student score (or -1 to stop): ");

 if (!double.TryParse(Console.ReadLine(), out double score))
 {
 Console.WriteLine("Invalid input. Please enter a valid score.");
 continue;
 }

 if (score == -1) { break; }

 totalScore += score;
 count++;

 minScore = Math.Min(minScore, score);
 maxScore = Math.Max(maxScore, score);
 }

 if (count > 0)
 {
 Console.WriteLine($"Number of students: {count}");
 Console.WriteLine($"Minimum score is {minScore:F2}");
 Console.WriteLine($"Maximum score is {maxScore:F2}");
 Console.WriteLine($"Average score is {totalScore / count:F2}");
 }
 else
 {
 Console.WriteLine("No valid scores entered.");
 }
}
```

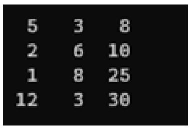


```
Enter student score (or -1 to stop): 1.2
Enter student score (or -1 to stop): 2.4
Enter student score (or -1 to stop): 3.6
Enter student score (or -1 to stop): 65
Enter student score (or -1 to stop): sdf
Invalid input. Please enter a valid score.
Enter student score (or -1 to stop): 1.345
Enter student score (or -1 to stop): -1
Number of students: 5
Minimum score: 1.20
Maximum score: 65.00
Average score: 14.71
```

4. Two-Dimensional Arrays

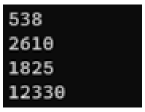
```
using System ;
class Matrix {
        public  static void Main () {
        int i, j;
        int[,]    A = new int [4 ,3] {
        { 5 , 3 , 8},
        { 2 , 6 , 10},
        { 1 , 8 , 25},
        {12 , 3 , 30}
        }

        for ( i = 0; i < 4; i++) {
                for ( j = 0; j < 3; j++) {
                        Console . Write ("{0 ,4 }", A [i,j]);
                        }
                Console . W rite Line ();
                }
        }

}
```

What is the output of the program?



```
   5   3   8
   2   6  10
   1   8  25
  12   3  30
```

What is the output of the program when line 14 is changed to Console.Write("{0}", A[i,j]); instead?



```
538
2610
1825
12330
```

5. Exercise 1.2: Write a program to read a matrix from the user, then display the entire matrix on the screen. The size of the matrix is also specified by the user.

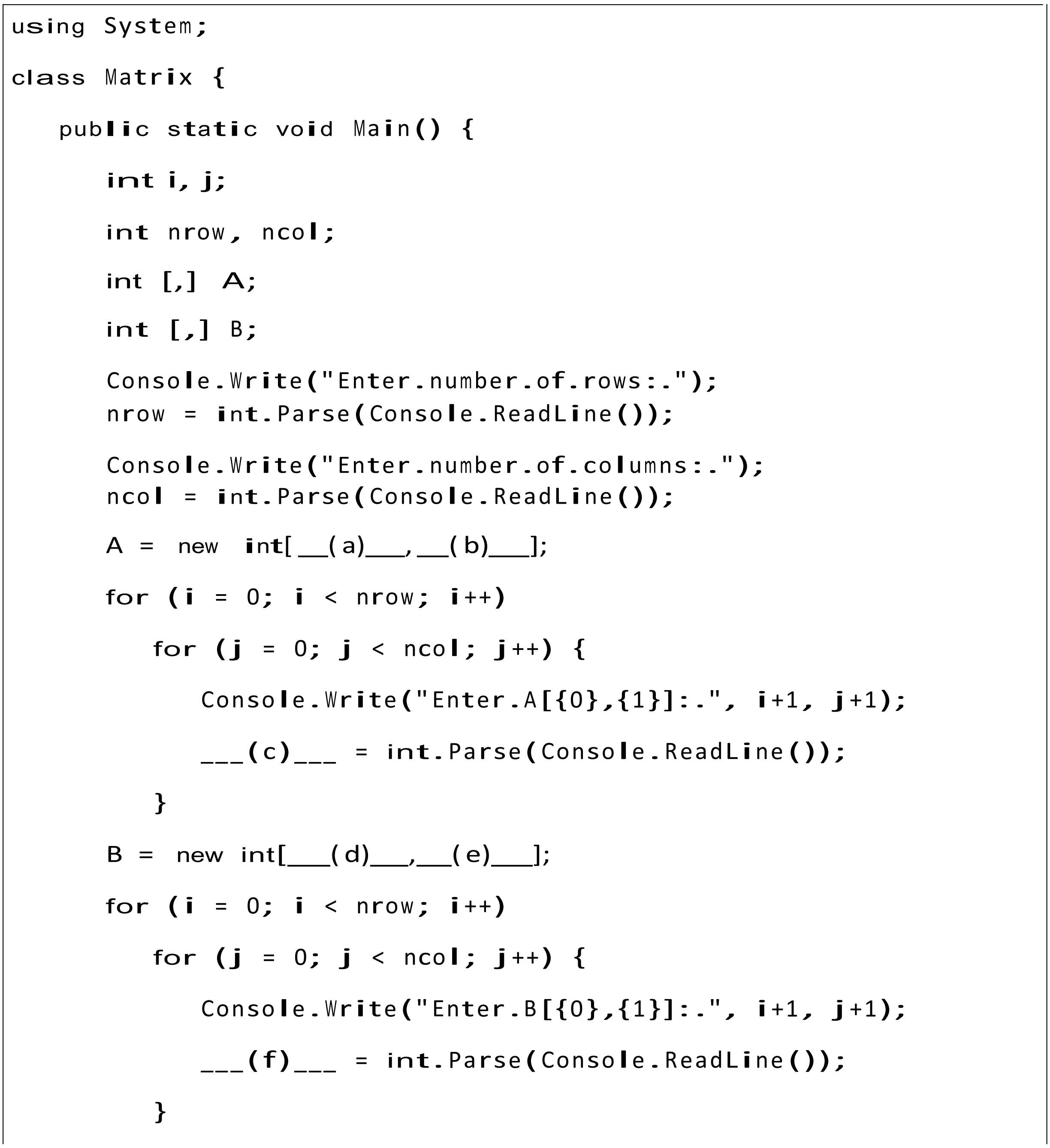
```
using System;
class Matrix {
    public static void Main() {
        int i, j j;
        int nrow, ncol;
        int [,] A;

        Console.Write("Enter.number.of.rows:."); nrow =
        int.Parse(Console.ReadLine());
        Console.Write("Enter.number.of.columns:."); ncol
        = int.Parse(Console.ReadLine());

        A =   new  int[__(a)___,__(b)___];
        for (i = 0; i < nrow; i++)
            for (j = 0; j < ncol; j++) {
                Console.Write("Enter.A[{0},{1}]:.", i+1, j+1);
                ___(c)___ = int.Parse(Console.ReadLine());
            }


        Console.WriteLine("Matrix.A.is");
        for (i = 0; i < nrow; i++) {
            for (j = 0; j < ncol; j++) {
                Console.Write("{0,4}", ___(d)___);
            }
            ______(e)______
        }
    }
}
```

Position	Expression/Statement
___(a)___	A = new int [2,3];
___(b)___	A = new int [2,3];
___(c)___	A [i, j] = int. Parse (Console.ReadLine());
___(d)___	Console.Write("{0,4}", A [i, j]);
___(e)___	Console.WriteLine();

Exercise 1.3: Enhance the program used in Exercise 1.2 so that it now also outputs the transpose of *A*. Write the *extended part* in the box provided

```
Console.WriteLine("Transpose of Matrix A is:");
for (j = 0; j < ncol; j++)
{
    for (i = 0; i < nrow; i++)
    {
        Console.Write("{0,4}", A[i, j]);
    }
    Console.WriteLine();
}
```

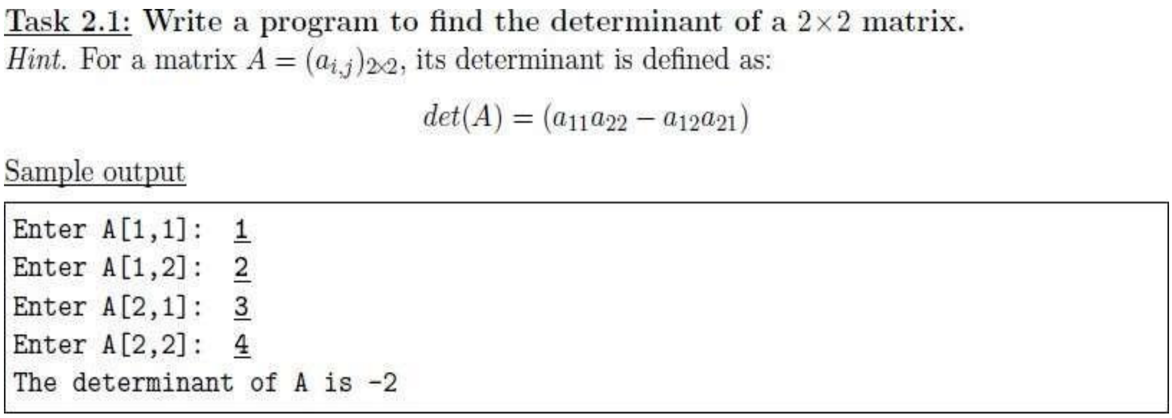
Exercise 1.4: The following (incomplete) program intends to take two matrices from the user. Then perform an addition operation on the two matrices and display the result. Complete the program by filling in an appropriate expression/statement for each of the positions marked as(a)–(g).



```
using System;

class Matrix {

   public static void Main() {

      int i, j;

      int nrow, ncol;

      int [,] A;

      int [,] B;

      Console.Write("Enter.number.of.rows:.");
      nrow = int.Parse(Console.ReadLine());

      Console.Write("Enter.number.of.columns:.");
      ncol = int.Parse(Console.ReadLine());

      A = new int[__(a)___,__(b)___];

      for (i = 0; i < nrow; i++)

         for (j = 0; j < ncol; j++) {

            Console.Write("Enter.A[{0},{1}]:.", i+1, j+1);

            ___(c)___ = int.Parse(Console.ReadLine());

         }

      B = new int[___(d)___,__(e)___];

      for (i = 0; i < nrow; i++)

         for (j = 0; j < ncol; j++) {

            Console.Write("Enter.B[{0},{1}]:.", i+1, j+1);

            ___(f)___ = int.Parse(Console.ReadLine());

         }
```

Position	Expression/Statement
___(a)____	A = new int [2,2];
___(b)____	A = new int [2,2];
___(c)____	A [i, j] = int. Parse (Console. ReadLine ());
___(d)____	B = new int [2,2];
___(e)____	B = new int [2,2];
___(f)____	B [i, j] = int. Parse (Console. ReadLine ());
___(g)____	Console. Write ("{0,4}", sum);

Programming Exercises



Task 2.1: Write a program to find the determinant of a 2×2 matrix.

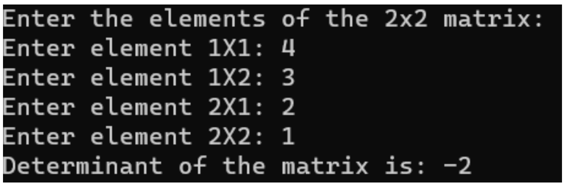
Hint. For a matrix $A = (a_{i,j})_{2\times2}$, its determinant is defined as:

$$det(A) = (a_{11}a_{22} - a_{12}a_{21})$$

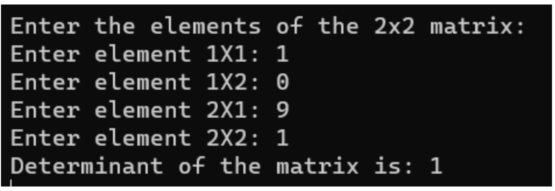
Sample output

```
Enter A[1,1]:  1
Enter A[1,2]:  2
Enter A[2,1]:  3
Enter A[2,2]:  4
The determinant of A is -2
```

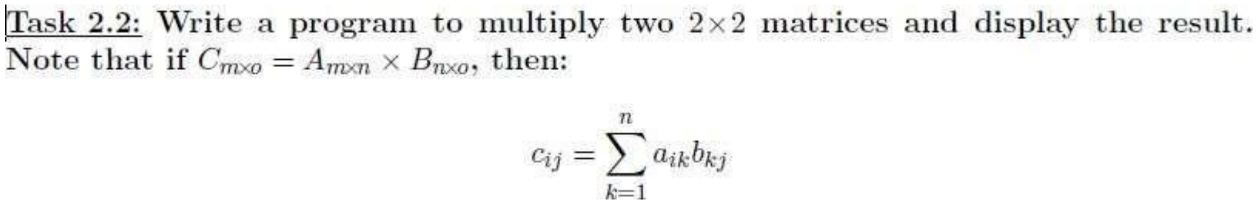
```
Console.WriteLine("Enter the elements of the 2x2 matrix:");
Console.Write("Enter element 1X1: ");
double a = double.Parse(Console.ReadLine());
Console.Write("Enter element 1X2: ");
double b = double.Parse(Console.ReadLine());
Console.Write("Enter element 2X1: ");
double c = double.Parse(Console.ReadLine());
Console.Write("Enter element 2X2: ");
double d = double.Parse(Console.ReadLine());

double determinant = (a * d) - (b * c);

Console.WriteLine("Determinant of the matrix is: " + determinant);
```



```
Enter the elements of the 2x2 matrix:
Enter element 1X1: 4
Enter element 1X2: 3
Enter element 2X1: 2
Enter element 2X2: 1
Determinant of the matrix is: -2
```



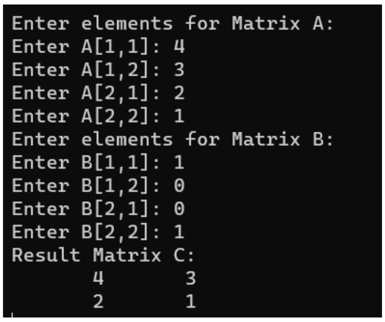
```
Enter the elements of the 2x2 matrix:
Enter element 1X1: 1
Enter element 1X2: 0
Enter element 2X1: 9
Enter element 2X2: 1
Determinant of the matrix is: 1
```



Task 2.2: Write a program to multiply two 2×2 matrices and display the result. Note that if $C_{mxo} = A_{mxn} \times B_{nxo}$, then:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj}$$

```
Console.WriteLine("Enter elements for Matrix A:");
 double[,] matrixA = new double[2, 2];
 for (int i = 0; i < 2; i++)
 {
     for (int j = 0; j < 2; j++)
     {
         Console.Write("Enter A[{0},{1}]: ", i + 1, j + 1);
         matrixA[i, j] = double.Parse(Console.ReadLine());
     }
 }

 Console.WriteLine("Enter elements for Matrix B:");
 double[,] matrixB = new double[2, 2];
 for (int i = 0; i < 2; i++)
 {
     for (int j = 0; j < 2; j++)
     {
         Console.Write("Enter B[{0},{1}]: ", i + 1, j + 1);
         matrixB[i, j] = double.Parse(Console.ReadLine());
     }
 }

 double[,] matrixC = new double[2, 2];
 for (int i = 0; i < 2; i++)
 {
     for (int j = 0; j < 2; j++)
     {
         matrixC[i, j] = 0;
         for (int k = 0; k < 2; k++)
         {
             matrixC[i, j] += matrixA[i, k] * matrixB[k, j];
         }
     }
 }

 Console.WriteLine("Result Matrix C:");
 for (int i = 0; i < 2; i++)
 {
     for (int j = 0; j < 2; j++)
     {
         Console.Write("{0,8}", matrixC[i, j]);
     }
     Console.WriteLine();
 }
```



```
Enter elements for Matrix A:
Enter A[1,1]: 4
Enter A[1,2]: 3
Enter A[2,1]: 2
Enter A[2,2]: 1
Enter elements for Matrix B:
Enter B[1,1]: 1
Enter B[1,2]: 0
Enter B[2,1]: 0
Enter B[2,2]: 1
Result Matrix C:
       4       3
       2       1
```