irePro

⚠ Please make sure you are visible in the camera

Single line of output should contain R_O.

Sample Input 1:
68 15
24 10
5

Sample Output 1:
33402

Explanation 1:
X=68, M=15.
Y=24, N=10.
O=5.
Converting X and Y to common base for multiplication. $65_{15}=98_{10}$.
$R_{10} = 98*24=2352_{10}$.
R_5= 33402.
Hence output is 33402.

Sample Input 2:
7a 13
9 11
15

Sample Output 2:
409

Explanation 2:
$7a_{13}=101_{10}$
$9_{11}=9_{10}$.
R_{10}= 909
R_{15}=409.

Previous Question

Next Question

HirePro

Please make sure you are visible in the camera

TIME LEFT 00:49:24

1 ☆

2

QUESTION: 1 GROUP: **Java Coding** SECTION: **Java**

Mark(s): 15

In decimal notation, we can represent numbers using any digit between 0-9 whereas base – 16 allows us to use 15 digits: 0-9, a, b, c, d, e, f. Similarly base – 20 allows us to use 19 digits: 0-9, a-j. You are given 2 numbers X_M and Y_N. Write a program to find the resultant R_O, where $R_O = X_M * Y_N$, M, N and O are bases of X, Y and R, respectively.

Read the input from STDIN and print the output to STDOUT. Do not write arbitrary strings anywhere in the program, as these contribute to the standard output and test cases will fail.

Constraints:

I) 2 <= M, N, O <= 36.

II) All characters which are part of input/output numbers are in lower case.

Input Format:

The first line of input contains a number, X and an integer, M, where X is an integer of base M.
The second line of input contans a number, Y and an integer, N, where Y is an integer of base N.
The third line of input contains an integer O, which is the base of R.

Output Format:

Single line of output should contain R_O.

Sample Input 1:

```
68 15
24 10
5
```

Sample Output 1:

```
33402
```

Explanation 1:

1

2 ☆

Sample Input1:

3 3
1 2 3
4 5 6
7 8 9

Sample Output1:

15
7 5 3

Explanation1:

Given matrix is 1 2 3
4 5 6
7 8 9

Reverse diagonal sums are formed by elements 1, 2+4, 3+5+7, 6+8, 9. So sums are 1, 6, 15, 14, 9. The highest is 15 which is formed by the elements 3, 5 and 7. So printing them in descending order, 7 5 3. Thus the output is 15 and 7 5 3.

Sample Input2:

3 5
10 20 30 40 50
110 120 130 140 150
100 90 80 70 60

Sample Output2:

270
140 80 50

Explanation2:

Given matrix is 10 20 30 40 50
110 120 130 140 150
100 90 80 70 60

Converting into a square matrix

HirePro

Please make sure you are visible in the camera

TIME LEFT 00:47:40

1

2

QUESTION 2 GROUP: **Python Coding** SECTION: **Python**

Mark(s): 15

Given a matrix of dimensions *MXN*, find the sum of elements for each reverse diagonal. Write a program to print the maximum sum S and elements of diagonal with the highest sum in descending order. If the given matrix is not square, converting it into a square matrix by inserting zeros.

For example:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Here,the reverse diagonal elements are {1},{5,2},{9,6,3},{13,10,7,4},{14,11,8},{15,12},{16}.

Read the input from STDIN and print the output to STDOUT. Do not write arbitrary strings while reading the input or while printing, as these contribute to the standard output.

Constraints:

2<= M, N<=20.

Input Format:

The first line of input contains two integers M and N separated by a single white space.
Next M lines of input contain N integers in each line, separated by a single white space.

Output Format:

The first line of output contains S, which is the maximum sum of reverse diagonal elements.
The second line of output contains elements of diagonal with the highest sum in descending order separated by a single white space.

Sample Input1:

3 3
1 2 3

Previous Question | Next Question | End Test

HirePro

Please make sure you are visible in the camera

TIME LEFT 00:47:04

1

2

printing them in descending order, 7 3 5. Thus the output is 15 and 7 3 5.

Sample Input2:

3 5
10 20 30 40 50
110 120 130 140 150
100 90 80 70 60

Sample Output2:

270
140 80 50

Explanation2:

Given matrix is

10	20	30	40	50
110	120	130	140	150
100	90	80	70	60

Converting into a square matrix,

10	20	30	40	50
110	120	130	140	150
100	90	80	70	60
0	0	0	0	0
0	0	0	0	0

Reverse diagonal sums are formed by the elements 10, 20+110, 30+120+100, 40+130+90, 50+140+80, 150+70, 60 and 0. So sums are 10, 130, 250, 260, 270, 220, 60. The highest is 270 which is formed by the elements 50+140+80. So printing them in descending order, 140 80 50. Thus output is

270
140 80 50.

Current Code | Previous Versions

Python 3

```
1  b = []      # b has elements of reverse diagonal which is having maximum sum in descending order.
2  bx = 0      # bx is number of elements in the reverse diagonal which is having maximum sum.
3
```

Previous Question | Next Question