

1. Se citesc m, n și o matrice cu m linii și n coloane, elementele unei linii fiind date pe o linie (elementele unei linii date pe o linie separate cu spațiu). Să se construiască în memorie și să se afișeze matricea transpusă (folosind și comprehensiune).

```
6 lista_aux = [int(x) for x in input().split()]
7 print(lista_aux)
8 n = lista_aux[0]
9 m = lista_aux[1]
10 matrix = [[] for i in range(n)]
11 #print(n, m, matrix, sep = " ")
12 for i in range(n):
13     matrix[i] = [int(x) for x in input().split()]
14
15 matrix_trans = [[0 for j in range(m)] for i in range(n)]
16
17 # x -> a[i][j] => x -> a_t[j][i]
18
19 for i in range(n):
20     for j in range(m):
21         matrix_trans[i][j] = matrix[j][i]
22
23 for i in range(n):
24     for j in range(m):
25         print(matrix_trans[i][j], end = " ")
26     print()
```

2. Se da un număr natural  $n > 2$ . Să se afișeze primele n linii din triunghiul lui Pascal (daca c este numărul maxim de cifre ale unui număr din triunghi, toate numerele se vor afișa pe c+1 caractere). De exemplu, pentru  $n=6$  se va afișa

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

1 5 10 10 5 1

```
48 n = int(input())
49 pascalTriangle = [[0 for i in range(n)] for j in range(n)]
50
51 for i in range(n):
52     pascalTriangle[i][0] = 1
53     pascalTriangle[i][i] = 1
54     for j in range(1, i):
55         pascalTriangle[i][j] = pascalTriangle[i-1][j] + pascalTriangle[i-1][j-1]
56     maxim = 0
57 for i in range(n):
58     for j in range(i+1):
59         pascalTriangle[i][j] = str(pascalTriangle[i][j])
60         if len(pascalTriangle[i][j]) > maxim:
61             maxim = len(pascalTriangle[i][j])
62 print(maxim)
63 for i in range(n):
64     for j in range(i+1):
65         while len(pascalTriangle[i][j]) < maxim:
66             pascalTriangle[i][j] = pascalTriangle[i][j] + " "
67 #for line in pascalTriangle:
68 #    for element in line:
69 #        print(element, end=' ')
70 #    print()
71 for i in range(n):
72     for j in range(i+1):
73         print(pascalTriangle[i][j], end=" ")
74     print()
```

3. Se da un N numar natural urmat de N linii. Fiecare linie va contine urmatoarele:

-- Un nume (sir de caractere fara spatii ex. "Gigel")

-- Un numar ce reprezinta banii pe care ii are respectivul in buzunar

(Daca nume = "Gigel" si numar = "7" atunci Gigel are 7 lei in buzunar)

Dupa cele N linii citite se citeste un M (pe o noua linie) urmat de M linii ce vor contine un sir de caractere. Pentru fiecare sir citit trebui sa se afiseze banii care ii are in buzunar. In cazul in care nu avem date despre banii detinuti de un nume citit se va afisa un mesaj corespunzator (ex: "Nu avem informatii despre cati bani are *nume*")

```
35 dict = {}
36 N = int(input())
37 for i in range(N):
38     linie = [x for x in input().split()] #Gigel 7
39     dict[linie[0]] = linie[1]
40
41 M = int(input())
42 for i in range(M):
43     cheie = input()
44     if cheie in dict:
45         print(cheie, " are in buzunar ", dict[cheie], " lei!", sep = " ")
46     else:
47         print(cheie, " nu are bani in buzunar :(", sep = " ")
48
```