2.

a) Scrieți o funcție care returnează o matrice triunghiulară de dimensiune n, având forma următoare:

- prima coloană conține numerele 1, 2, 3, …, n.

- ultima linie conține numerele n, n-1, …, 2, 1.

- restul elementelor aflate în triunghiul de sub diagonala principală se calculează ca suma elementelor vecine de la vest, sud și sud-vest.

Exemplu: pentru n = 4 => M=[[1], [2,15], [3,10,15], [4,3,2,1]]

1

2 15

3 10 15

4 3 2 1

b) Scrieți o funcție care afișează o listă de liste sub formă de matrice, cu coloanele aliniate la dreapta.

Varianta 1:

def matrice(n):  
 m = [[i+1] + [0] \* i for i in range(n-1)]  
 m.append(list(reversed(range(1,n+1))))  
  
 for i in range(n-2,0,-1):  
 for j in range(1,i+1):  
 m[i][j] = m[i][j-1] + m[i+1][j] + m[i+1][j-1]  
 return m  
  
  
def afisare(m):  
 maxi = max([max(linie) for linie in m])  
  
 for linie in m:  
 for elem in linie:  
 print(str(elem).rjust(len(str(maxi)),' '),end=' ')  
 print()  
  
  
d = matrice(4)  
print(d) #a  
afisare(d) #b

Varianta 2:

def makeMatrix(n): # O functie care construieste matricea  
 global m  
 k = n  
 m = [[0 for x in range(k)] for k in range(n, 0, -1)] # Construim matricea goala, invers  
 for i in range (n-1,-1,-1): # Construim ultima linie si prima coloana  
 m[0][i] = n-i # a matricei cerute  
 m[i][0] = n-i # Ultima linie e de fapt prima in m  
 for i in range(1,n-1):  
 for j in range(1,n-i):  
 m[i][j] = m[i-1][j-1] + m[i-1][j] + m[i][j-1] # Construim restul elementelor  
def showMatrix(): # O functie care afiseaza matricea  
 for i in range(n-1,-1,-1):  
 print(m[i], end='\n')  
def showMaxInMatrix():  
 global maxim  
  
 for i in range(0,n):  
 maxim = max(maxim,max(m[i]))  
 print("Maximul este:", maxim)  
  
n = int(input("Introdu numarul n:" ))  
m = []  
maxim = 0  
makeMatrix(n)  
showMatrix()  
showMaxInMatrix()

3. Scrieți o funcție cu număr variabil de parametri care furnizează toate listele care conțin un x dat ca prim parametru.

Exemplu: cauta\_x(7, [5,1,7,3,7], [2,3], [-3,7,1]) => [5,1,7,3,7], [-3,7,1]

5. a) Pentru un student se cunosc următoarele informații: numele, grupa și o listă cu creditele obținute la toate examenele din anul respectiv. Considerând o listă de studenți, scrieți o funcție care să adauge la fiecare student situația sa școlară: promovat (True) sau nepromovat (False). Pentru a fi considerat promovat, un student trebuie să nu aibă nici un examen nepromovat (adică un 0 în lista cu creditele), iar suma creditelor obținute să fie mai mare sau egală decât un anumit număr minim de credite dat

b) Scrieți câte o funcție comparator care să sorteze studenții după următoarele criterii: • crescător după grupă și în fiecare grupă în ordine alfabetică

• întâi cei promovați, apoi cei nepromovați și în fiecare categorie în ordine alfabetică 1 2 15 3 10 15 4 3 2 1 2

• descrescător după suma creditelor, iar în cazul unor sume egale în ordinea crescătoare a grupei și în ordine alfabetică în cadrul grupei

• în ordinea crescătoare a grupelor, în cadrul fiecărei grupe mai întâi studenții promovați, iar apoi cei nepromovați, în fiecare categorie (promovat/nepromovat) în ordinea descrescătoare a sumei creditelor și, în cazul unor sume egale, în ordine alfabetică