1. Se citesc 𝑛 – 2 numere naturale distincte din mulțimea {1, 2, … , 𝑛}. Să se afișeze cele două numere lipsă.
2. nums = [int(x) for x in input('Introduceti numerele : ').split()]
3. n = len(nums) + 2
4. x\_xor\_y = 0
5. for num in nums:
6. x\_xor\_y ^= num
7. for i in range(1,n+1):
8. x\_xor\_y ^= i
9. mask = x\_xor\_y & (~(x\_xor\_y - 1))
10. x = 0
11. for num in nums:
12. if (mask & num) == 0:
13. x ^= num
14. for i in range(1,n+1):
15. if (mask & i) == 0:
16. x ^= i
17. y = x ^ x\_xor\_y
18. print(x, y)

0. Pozitiile pe care apare sirul t in sirul s

s = input("s=").strip()

t = input("t=").strip()

poz = []

lung = len (t)

nr = s.find (t)

while nr != -1:

poz += [nr]

nr = s.find (t, nr + 1)

if poz == []:

print ("sirul nu apare")

else:

print("sirul apare pe pozitiile", end=" ")

print (poz)

4. Se citește un cuvânt w, un număr natural nenul 𝑝 și un șir format din 𝑛cuvinte. Să se afișeze toate cuvintele care sunt 𝑝-rime cu 𝑤, respectiv ultimele 𝑝 caractere coincid.

w = input ("w:")  
p = int (input("p:"))  
n = int (input("n:"))  
r = ""  
for i in range (n):  
 x=input("Cuvant:")  
 if w[-p:] == x[-p:]:  
 r += x + ' '  
if r != "":  
 print(r)  
else:  
 print("Niciun cuvant nu rimeaza")

5. Se citește o propoziție/frază în care cuvintele sunt despărțite între ele prin spații și semnele de punctuație uzuale. Să se afișeze toate cuvintele distincte de lungime maximă.

prop = input("Propozitia: ")  
  
semne = ',.?!;:'  
for i in semne:  
 prop = prop.replace(i, ' ')  
  
  
prop = prop.split()  
max = 0  
rez = ""  
for cuv in prop:  
 if len(cuv) == max and cuv not in rez:  
 rez += cuv + ' '  
 else:  
 if len(cuv) > max:  
 rez = cuv + ' '  
 max = len(cuv)  
  
print(rez)

6. Se citește un șir de caractere 𝑠. Să se verifice dacă există un șir 𝑡, diferit de 𝑠, astfel încât 𝑠 să se poată obține prin concatenarea de un număr arbitrar de ori 𝑘 a șirului 𝑡. Exemplu: pentru 𝑠 = ”𝑎𝑏𝑐𝑎𝑏𝑐” => 𝑡 = ”𝑎𝑏𝑐”, 𝑘 = 2.

