MODEL PA

Subiectul I

a)
```
def divizori(*param):
    di = dict()
    for x in param:
        d = 2
        aux = x
        l = []
        while aux:
            ok = 1
            while aux % d == 0:
                aux = aux // d
                ok = 0
            if ok == 0:
                l.append(d)
            d = d + 1
        di[x] = l
    return di
```

b) `litere_10 = [chr(ord('a') + i) for i in range(0, 10)]`

c) $T(1) = 1$

$T(n) = T(n/3) + 2$

$T\left(\frac{n}{3}\right) = T\left(\frac{n}{3} \cdot \frac{1}{3}\right) + 2 = T\left(\frac{n}{3^2}\right) + 2$

$T(n) = \left(T\left(\frac{n}{3^2}\right) + 2\right) + 2$

$T(n) = \left(T\left(\frac{n}{3^3}\right) + 2\right) + 2 + 2$

$T(n) = \left(T\left(\frac{n}{3^4}\right) + 2\right) + 2 + 2 + 2$

$$T(n) = T\left(\frac{n}{3^k}\right) + 2 \cdot k \quad , k = \log_3 n$$

$$T\left(\frac{n}{3^k}\right) = T\left(\frac{n}{3^{\log_3 n}}\right) = T\left(\frac{n}{n}\right) = 1$$

$$\Rightarrow T(n) = T(1) + 2\log_3 n = 1 + 2\log_3 n$$

$$\Rightarrow O(\log_3 n)$$

Subiectul 2

```
n = int(input())
l = []
for i in range(n):
    a,b = input().split()
    a = int(a)
    b = int(b)
    ll = (a,b)
    l.append(ll)
l.sort(key = lambda e : e[1])
sol = []
sol.append(0)
u = 0
for i in range(1,n):
    if l[u][1] < l[i][0]:
        sol.append(i)
        u = i
m = len(sol)
print(m)
for i in range(m):
    print(l[i][0], l[i][1])

# Complexitatea este O(n log n)
```

Corectitudine

Presupunem spectacolele ordonate crescător după timpul de terminare.

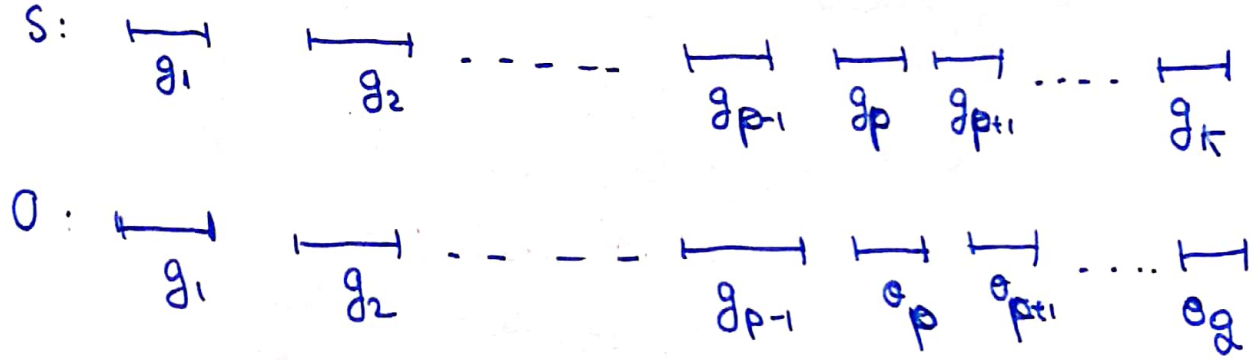
Notăm $S = \{g_1, g_2, \dots g_k\}$ - soluția greedy.

$t_{g_1} \le t_{g_2} \le \dots \le t_{g_k}$

Presupunem prin reducere la absurd că există

$O = \{o_1, o_2, \dots o_q\}$ - o soluție optimă, diferită de greedy ($q > k$), O având nr maxim de elemente în comun cu greedy.

Fie p prima poziție unde S și O diferă, $p \le k$.



Avem: $t_{g_p} \le t_{o_p}$

Consider $O' = O \setminus \{o_p\} \cup \{g_p\}$

O' - tot o soluție corectă

Avem: g_p nu se intersectează cu $g_1, g_2, \dots g_{p-1}$

g_p nu se intersectează cu $o_{p+1}, \dots o_q$

$t_{g_p} \le t_{o_p} < t_{o_{p+1}} < \dots < t_{o_q}$

$\Rightarrow |O'| = |O| \Rightarrow O'$ este soluție optimă, dar are mai multe elemente în comun cu cea greedy $\Rightarrow$ ab

Subiectul 3

```
n = int(input())
v = [int(x) for x in input().split()]
l = [0 for i in range(n+1)]
poz = [0 for j in range(n+1)]
l[n-1] = 1
poz[n-1] = -1
for i in range(n-2,-1,-1):
    ma = 0
    p = -1
    for j in range(i, n):
        if l[j] > ma and v[i] < v[j]:
            ma = l[j]
            p = j
    l[i] = ma+1
    poz[i] = p
ma = max(l)
p = l.index(ma)
print(ma)
while p != -1:
    print(v[p], end=" ")
    p = poz[p]

# Complexitatea este O(n^2)
```

Subiectul 4

```
n = int(input())
x = [0 for i in range(n+1)]
def back(k):
    if k == n+1:
        print(*x[1:k], sep=" ")
    else:
        for i in range(1,n+1):
            x[k]=i
            if x[k] not in x[:k]:
                back(k+1)

back(1)
```