

# TUTORIAT 1

1. Se dă un număr natural pozitiv  $n$ .

a) Scrieți o expresie care este adevărată dacă și numai dacă  $n$  este o putere a lui 2 (v. seminar)

```
#a)
n = int(input())
if n & (n - 1) == 0:
    print("Putere a lui 2")
else:
    print("Nu este putere a lui 2")
```

b) Să se verifice  $n$  este de forma  $2^k$ . În caz afirmativ să se afișeze valoarea  $k$  (folosind operatori pe biți).

```
#b)
n = int(input())
if n & 1 == 1:
    print("Numar impar")
else:
    print(n >> 1)
```

2. Să se genereze toate submulțimile mulțimii  $A = \{1, 2, \dots, n\}$ , unde numărul natural nenul  $n \leq 10$  se citește de la tastatură (fără backtracking). Indicație: Folosim vectori caracteristici memorați ca numere naturale de la 0 la  $2^n - 1$

```
n = 3

for nr in range(2 ** n):
    for bit in range(n):
        if nr & (1 << bit) != 0:
            print(bit + 1, end=' ')

    print('\n', end='')
```

3. Se citesc un număr natural  $n$  și un șir format din  $n$  numere întregi ( $n \geq 2$ ). Să se afișeze cele mai mari două valori distincte din șir sau mesajul "Imposibil", dacă aceste

```
n = int(input())

INF = 1000000007
max1 = -INF
max2 = -INF

for i in range(n): # [0, 1 ... n - 1]
    val = int(input())
    if val > max1:
        max2 = max1
        max1 = val
    else:
        if val > max2 and val != max1:
            max2 = val
if max1 == -INF or max2 == -INF:
    print('Imposibil')
else:
    print(max1, max2, sep = ' ')
```

Varianta cu cititul fiecărui element pe câte o linie din consolă

```

n = int(input())

INF = 1000000007
maxx1 = -INF
maxx2 = -INF
v = [int(x) for x in input().split()]
for elem in v:
    if elem > maxx1:
        maxx2 = maxx1
        maxx1 = elem
    else:
        if elem < maxx1 and elem > maxx2:
            maxx2 = elem
if maxx1 == -INF or maxx2 == -INF:
    print("Imposibil")
else:
    print(maxx1, maxx2, sep = ' ')

```

**Varianta cu cititul listei de pe aceeași linie din consolă**

4. Se dă o sumă  $S$  și avem la dispoziție bancnote cu valorile: 1, 5, 10, 25. Să se determine o modalitate de a plăti suma  $S$  folosind un număr minim de bancnote. Algoritmul propus mai funcționa și dacă aveam bancnote cu valorile 1, 10, 30, 40? Justificați. Exemplu de ieșire pentru suma 123: 3 x bancnote de 1 2 x bancnote de 10 4 x bancnote de 25 (nu contează ordinea în care se afișează bancnotele din descompunere) Declarați vector de bancnote:  
bancnote=[1,5,10,25]

```

sum = int(input())
bancnote = [1, 5, 10, 25]
for i in range(len(bancnote) - 1, -1, -1): #range descrescator [3, 2, 1, 0]
    if sum == 0:
        break
    print('Numarul de bancnote de', bancnote[i], 'lei este', sum // bancnote[i], sep = ' ')
    sum = sum - (sum // bancnote[i]) * bancnote[i]

```

5. Sir periodic. 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 ( $k > 1$ ) a șirului  $t$  (adică să se verifice dacă șirul  $s$  este periodic). Dacă există mai multe astfel de șiruri  $t$  se va determina cel mai lung. Exemplu:  $s = abbaabbaabbaabba \Rightarrow t = abbaabba$

```
ok = False
s = input()
for i in range(0, len(s) - 1):
    if len(s) % (i + 1) == 0:
        t = s[:i + 1]
        aux = t
        for j in range(1, len(s) // (i + 1)):
            t = t + aux
        if t == s:
            ok = True
            cel_mai_mare_sir = aux
if ok == False:
    print('Nu exista sir')
else:
    print(cel_mai_mare_sir)
```