

Seminar 1

● Cuvor ●:

Operatoare pe biti

$$24 = (11011)_2$$

$$\begin{array}{l} 24 : 2 = 12 \text{ r } 0 \\ 12 : 2 = 6 \text{ r } 0 \\ 6 : 2 = 3 \text{ r } 0 \\ 3 : 2 = 1 \text{ r } 1 \\ 1 : 2 = 0 \text{ r } 1 \end{array}$$

① negare $\rightarrow \sim$ (tilde)

$$\sim 24 = (00100)_2 = 4$$

$$\begin{array}{c|c|c|c} \sim & 1 & 1 & 0 & 1 & 1 \\ \hline & 0 & 0 & 1 & 0 & 0 \end{array}$$

② &/AND $\rightarrow \&$ (ampersand)

$$x = 24 = 11011$$

$$y = 5 = 00101$$

$$x \& y = 00001 = 1$$

$$\begin{array}{c|c|c|c} \& & 1 & 1 & 0 & 1 & 1 \\ \hline & 0 & 0 & 1 & 0 & 0 & 1 \end{array}$$

③ sau/or $\rightarrow |$ (pipe)

$$x | y = 11111 = 31$$

$$x = 11011, y = 00101$$

$$\begin{array}{c|c|c|c} | & 1 & 1 & 0 & 1 & 1 \\ \hline & 0 & 0 & 1 & 0 & 1 \end{array}$$

④ sau-exclusiv $\rightarrow ^$ (circumflex) XOR

$$x ^ y = 11110 = 30$$

$$\begin{array}{c|c|c|c} ^ & 1 & 1 & 0 & 1 & 1 \\ \hline & 0 & 0 & 1 & 0 & 0 \end{array}$$

⑤ deplasare la stg / left shift $\rightarrow \ll$

$$24 \ll 3 = 11011 \ll 3 = 11011000 = 196 = 24 \cdot 2^3$$

$$x \ll b = x \cdot 2^b$$

⑥ deplasare la dr / right shift $\rightarrow \gg$

$$24 \gg 3 = 11011 \gg 3 = 11 = 3 = \frac{24}{2^3}$$

Testarea parității

$$x = 27 = 11011 \rightarrow \text{dă suma} - \text{LSB} = \underbrace{1 \cdot 2^0 + 1 \cdot 2^1 + 0 \cdot 2^2 + 1 \cdot 2^3 + 1 \cdot 2^4}_{\text{impar?}} = \text{mereu par}$$

$$x = 24 = 11011$$

$$m = 1 = 00001$$

$x \oplus m = 00010 \Rightarrow \text{impar}$, dacă era par $\neq 0$

Paritatea o testăm prin &

if ($x \& 1 \neq 0$):

print ("Par")

else:

print ("impar")

Exerciții

①

XOR prope

$$\begin{cases} v \wedge v = 0 \\ v \wedge 0 = v \\ v \wedge z = z \wedge v \end{cases}$$

$$\begin{aligned} \text{fun } \text{c++} : \\ x &= x + y \\ y &= x - y \\ x &= x - y \end{aligned}$$

$$x = x \wedge y$$

$$y = x \wedge y \quad // = x \wedge y \wedge y = x \wedge 0 = x$$

$$x = x \wedge y \quad // = x \wedge y \wedge x = y$$

②

verig $m = 2^k$? și cât e dacă \nexists .

$$m = 2^k \Rightarrow m = 1000 \dots 00_{(2)} \Rightarrow \text{primul bit} = 1$$

nr. de 0 eliminate = k

```
m = int(input("m = "))
```

```
k = 0
```

```
while m & 1 == 0:
```

```
    m = m >> 1
```

```
    k = k + 1
```

```
if m == 1:
```

```
    print(k)
```

```
else:
```

```
    print("nu")
```

Complexitate: $O(\log_2 m)$

$$m \rightarrow 1 + \lceil \log_2 m \rceil \text{ biți}$$

mai eficient :

$$\begin{array}{r}
 m = 2^k = 10000 \\
 m-1 = 01111 \\
 \hline
 m \& (m-1) = 00000
 \end{array}$$

```

m = int(input("m="))
k = 0
while m & (m-1) != 0:
    print(log2(m))
    m = m & (m-1)
else:
    print("nu")

```

③

$m = 1011011000 \Rightarrow k = 5$ biți menuni

```

m = int(input("m="))
k = 0
while m:
    if m & 1 == 0:
        k = k + 1
    m = m >> 1
print(k)

```

Complexitate $O(\log_2 m)$

$$\begin{array}{r}
 m = 100010001000 \\
 m-1 = 100010000111 \quad \leftarrow 11 \\
 \hline
 m \& (m-1) = 100100000000
 \end{array}$$

act algoritma \uparrow

```

while m != 0:
    m = m & (m-1)
    k = k + 1
print(k)

```

④

verif dacă

$$x = 1001101$$

$$y = 0011011$$

sau calc. $x \wedge y = 1010110 \Rightarrow \text{roșu } k = 4$

⑤

$$2, 1, 7, 7, 1, 5, 2$$

2 sf. dacă al carei gave 1 dată

$$x = 2^1 \wedge 7^4 \wedge 7^1 \wedge 5^2 = 0101010101010101 = 5$$

```

m = int(input("m="))
for i in range(m):
    k = int(input("k="))
    x = x ^ k
print(x)

```

Programarea algoritmilor

SEMINAR NR. 1

1. Să se interschimbe valorile a două variabile de tip întreg folosind operatorul \wedge (XOR/sau exclusiv pe biți).
2. Să se verifice dacă un număr natural nenul n este de forma 2^k sau nu. În caz afirmativ să se afișeze valoarea k .
3. Să se determine în mod eficient numărul de biți nenuli din reprezentarea binară a unui număr natural.
4. Fie x și y două numere naturale. Calculați numărul biților din reprezentarea binară internă a numărului x a căror valoare trebuie comutată pentru a obține numărul y .
5. Se citește un șir format din numere naturale cu proprietatea că fiecare valoare distinctă apare de exact două ori în șir, mai puțin una care apare o singură dată. Să se afișeze valoarea care apare o singură dată în șir.
6. Să se găsească lungimea maximă a unei secvențe de biți egali cu 1 din reprezentarea binară a unui număr natural dat.
7. Să se calculeze numărul obținut prin aplicarea operatorului XOR între toate elementele tuturor submulțimilor unei mulțimi nevide $A = \{a_1, a_2, \dots, a_n\} \subset \mathbb{N}$, mai puțin mulțimea vidă. De exemplu, pentru mulțimea $A = \{2, 7, 4\}$ trebuie afișată valoarea $v = (2) \wedge (7) \wedge (4) \wedge (2 \wedge 7) \wedge (2 \wedge 4) \wedge (7 \wedge 4) \wedge (2 \wedge 7 \wedge 4) = 0$, unde am folosit parantezele pentru a evidenția submulțimile lui A .