**Determinarea celei mai lungi secvente cu o proprietate data dintr-un vector**

1. **Cea mai lunga secventa de numere pare dintr-un vector de numere intregi (cea mai din stanga)**

**Exemplu:**

**17**

1 **68 24 6** 7 23 **2 4 6 8** 15 **44 60 100 98** 3 **64**

**Afiseaza:**

**4**

**2 4 6 8**

l=lmx=0; // l=lungimea secventei curente

for (i=1; i<=n; i++)

if (v[i]%2==0)

{l++;

if (l>lmx)

{lmx=l;

u=i; /u= indicele ultimului el. din secventa de lungime maxima

}

}

else l=0;

cout<<lmx<<endl;

for (i=**u-lmx+1**; i<=u; i++) cout<<v[i]<<” ”;

**Obs 1.** Pentru a determina cea mai din **dreapta** secventa de lungime maxima: **if (l>=lmx)**

**Obs 2. Algoritmul are complexitatea O(n), deci este algoritm liniar.**

1. **Problema platoului - Cea mai lunga secventa de numere egale dintr-un vector de numere intregi (cea mai din stanga) – se va afisa lungimea secventei maxime si numarul din care e formata.**

**Exemplu:**

**16**

**1 3 3 3 6 6 10 10 10 10 11 12 12 12 12 13**

**Afiseaza:**

**4**

**10**

**l=lmx=1**; // l=lungimea secventei curente

for (i=**2**; i<=n; i++)

if (v[i]==v[i-1])

{l++;

if (l>lmx)

{lmx=l;

r=v[i]; // r= el. din care este formata secventa de lungime maxima

}

}

else **l=1**;

cout<<lmx<<endl<<r;