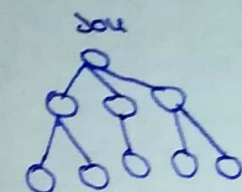
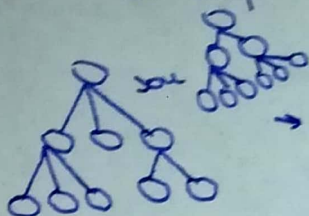


Subiectul 1

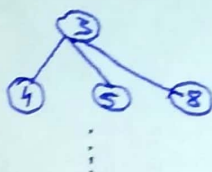
1. d) \rightarrow Expresia verifcă dacă exact una dintre cele trei valori are ultima cifră impară, adică dacă exact una dintre valori este impară

2. Un arbore de ordine cu înălțime minimă de următoarea structură



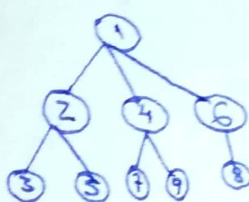
(ordinea trebuie să aibă 5 frunze)

Pentru T_1 :



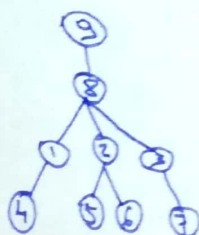
\Rightarrow vectorul de tota T_1 nu descrie un arbore, deoarece modulul 1 fiind cel mai mic (la fel și pentru modulul 2)

Pentru T_2 :



$\Rightarrow \checkmark$

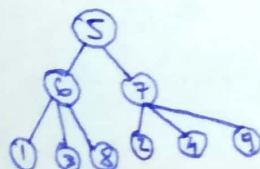
Pentru T_3 :



$\Rightarrow \times$

$\Rightarrow T_2, T_4 (c)$

Pentru T_4 :



$\Rightarrow \checkmark$

3. b)

4. G.N 20 noduri

$$d(x) > 0, \forall x = \overline{1, 20}$$

12 muchii

nr. min. comp. con = ?

\uparrow Incepem cu nr. min. comp. con = 1

\Rightarrow graful or trebui sa aiba cel putin 12 muchii
(scut doar 12)

Pentru m.m.c.c = 2

Indiferent de numarul de noduri din fiecare
comp. conex, se observa ca graful or trebui
sa aiba cel putin 18 muchii

de ex: prima componenta con. are 2 noduri tot

cea de a doua 18, atunci nr. min. muchii necesare

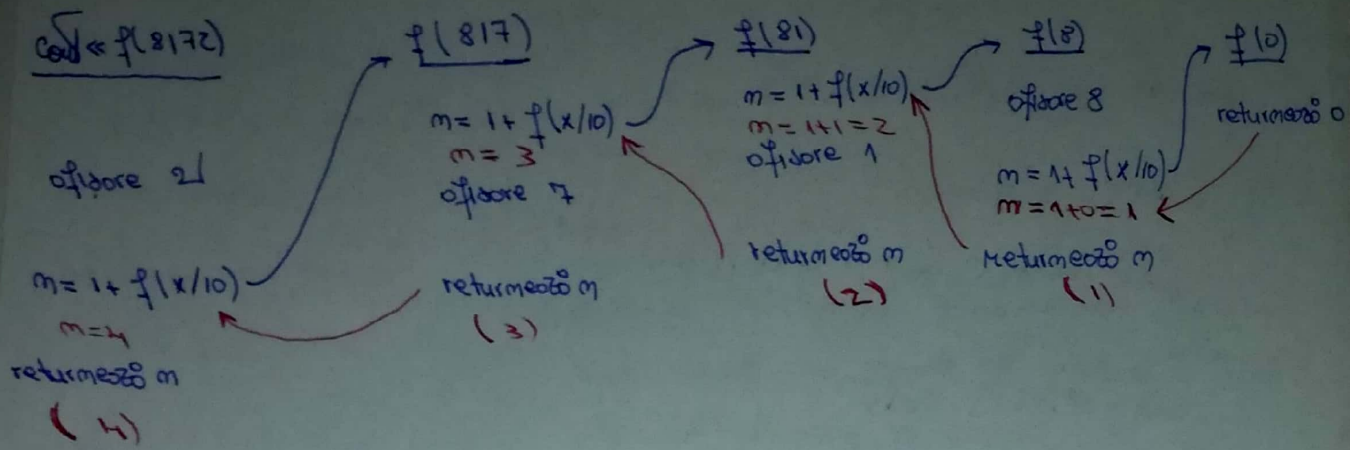
$$\text{este } 1 + 17 = 18$$

Astfel, se observa ca, numarul de muchii necesare pentru a obtine un
graf cu numarul maxim de componente conexe scade cu 1 de fiecare
data cand creste numarul minim de componente conexe.

$$\text{Deci, nr. muchii (minim)} = 20 - \text{nr. de componente conexe (minim)}$$

Cu graful are 12 muchii, se observa ca nr. minim de componente conexe
este egal cu 8.

5



Valoarea offshore este 28174 (c)

Soluție

a) 13

b) 1, 2, 3, 5, 7

c) #include <iostream>

using namespace std;

int main()

{ int m, a, b, c;

c = m;

a = 1;

b = m/2;

while(a != 0 && b > 0)

{

c = m;

while(c >= b)

c = c - b;

a = c;

b = b - 1;

}

b = b + 1;

cout << b;

return 0;

}

d)

este m (m număr natural nenul)

a ← 1

b ← ⌊m/2⌋

- cât timp a ≠ 0 și b > 0 execută

a ← m % b

a ← c

b ← b - 1

□

b ← b + 1

return b

2.

```
for (int i=0; i<M.nrProduce; i++)
```

```
{ if (M.P[i].cont > 0)
```

```
    cout << M.P[i].cont * M.P[i].pret;
```

```
    else
```

```
    cout << M.P[i].cod;
```

```
    cout << "\n";
```

```
}
```

3.

```
if (i == j)
```

```
    a[i][j] = 6;
```

```
else if (i % 2)
```

```
    a[i][j] = 5 - j;
```

```
else a[i][j] = 5 - i;
```