Varianta 7

Subiectul I

1. !(x<2001) and (x<=2002) or (x>=2019) and !(x>2020)

⟺

(x>=2001) and (x<=2002) or (x>=2019) and (x<=2020)

d)

2.

f(12345)
afisare '+'
f(123)
afisare 12345
afisare '+'

f(123)
afisare '+'
f(1)
afisare 123
afisare '+'

f(1)
afisare '+'
f(0)
afisare 1
afisare '+'

f(0)
afisare '+'
afisare '+'

=> f(0) → afisare ++

=> f(1) → afisare +++1+

=> f(123) → afisare ++++1+123+

=> f(12345) → afisare +++++1+123+12345+

a)

3. Se generează mulțimile de 6 luate câte 4. Condiția ca o construire validă este ca elementele mulțimii să fie în ordine strict crescătoare.

Se observă că singura variantă ce îndeplinește acest aspect este a doua

b)

4. Pentru a determina gradul intern al unui nod calculăm suma elementelor de pe coloana corespunzătoare.

c)

5. Indiferent de numărul de noduri pe care le atribuim unei componente conexe, răspunsul va fi 15.

În general, pentru a obține x componente conexe având la dispoziție n noduri, numărul minim necesar de muchii (componentele conexe obținute vor fi arbori) va fi egal cu n-x

c)

Subiectul II

1. Algoritmul descris în pseudocod formează un număr din cifrele impare ale lui n, în ordine descrescătoare

a) 7533

b) 20 88 (cel mai mic/mare număr care nu conține cifre impare)

c)

```
# include <iostream>
 using namespace std;
 int x, n, p, m, cn, c;
 int main()
 {  cin>>n;
   x=1; m=0; p=1;
  while( x<10)
  {
    cn=n;
    while(cn!=0)
     { c=cn%10; cn/=10;}
    if(c==x)
     { m=c*p+m; p*=10;}
    }
   x+=2;
  }
 cout<<m;
 return 0;

}
```

d)

```
citeste n
m←0; p←1
pentru x←1,10,2
   cn←n
   cât timp cn≠0 execută
      c←cn%10; cn←[cn/10]
      dacă c=x atunci
         m←c*p+m; p←p*10
      □
   □
□
scrie m
```

```
2. struct biblio {
    unsigned short numar;
    struct {
      char titlu[20];
      char autor[20];
     } carte[100];
  }c;
```

3.
 au
 rosu