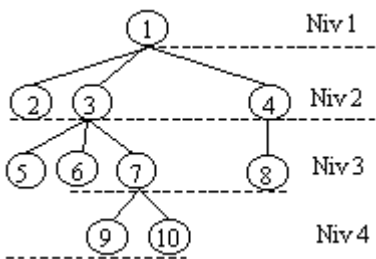


## VARIANTA 9 - Rezolvare

### Subiectul 1

1.	Se accesează valorile memorate în variabilele <b>a</b> și <b>b</b> folosind operatorii <b>mod</b>   %, -, *. Pentru precizarea răspunsului <b>d</b> ) se acordă <b>4p</b> .
2.	Se urmăresc apelurile recursive și se fac calculele: $f(5427) = f(542) * 10 + 7 - 1$ $f(542) = f(54) * 10 + 2 + 1$ $f(54) = f(5) * 10 + 4 + 1$ $f(5) = f(0) * 10 + 5 - 1$ $f(0) = 0$ $\Rightarrow f(5) = 0 * 10 + 5 - 1 = 4$ $\Rightarrow f(54) = f(5) * 10 + 4 + 1 = 45$ $\Rightarrow f(542) = f(54) * 10 + 2 + 1 = 453$ $\Rightarrow f(5427) = f(542) * 10 + 7 - 1 = 4536$ Pentru precizarea răspunsului <b>b</b> ) se acordă <b>4p</b> .
3.	Se generează în aceeași manieră ca în exemplul din enunț. Pentru răspunsul <b>b</b> ) se acordă <b>4p</b> .
4.	Din vectorul de referințe ascendente deducem că rădăcina arborelui este nodul <b>1</b> ( <b>tata[1]=0</b> ). Observăm că nodurile <b>5,6,7</b> sunt fii nodului <b>3</b> .  <p>Pentru precizarea răspunsului <b>a</b>) se acordă <b>4p</b>.</p>
5.	Putem folosi <b>10</b> noduri pentru a face un graf complet cu <b>45</b> de muchii sau cu <b>9</b> noduri pentru a face un graf cu <b>36</b> de muchii. Cum noi avem <b>44</b> de muchii, le putem folosi cu <b>10</b> noduri. Deci ne rămân <b>44</b> de noduri cu grad 0. Pentru precizarea răspunsului <b>a</b> ) se acordă <b>4p</b> .

## Subiectul 2

1.	<p>a) Se compară succesiv cifrele lui <b>n</b> cu valoarea variabilei <b>max</b> inițializată la început cu -1. Pentru a avea acces la toate cifrele lui <b>n</b>, se începe compararea cu cifra unităților și apoi se elimină această cifră repetând procedeul până când <b>n</b> devine <b>0</b>. Variabila <b>max</b> va memora cea mai mare cifră a numărului <b>n</b>.</p> <p>Pentru precizarea valorii <b>5</b> se acordă <b>6p</b>.</p>		
	<p>b) Pentru variabila <b>n</b> poate fi citit orice număr natural de trei cifre care are prima cifră diferită de <b>0</b>. Sunt 4 valori posibile pentru prima cifră, <b>5</b> valori posibile pentru a doua cifră și <b>5</b> valori posibile pentru a treia cifră. În total sunt</p> $1 \times 5 \times 5 + 4 \times 1 \times 5 + 4 \times 5 \times 1 = 65$ <p>numere care pot fi citite pentru variabila <b>n</b>.</p> <p>Pentru precizarea valorii <b>65</b> se acordă <b>4p</b>.</p>		
	<p>c) Înlocuirea structurii <b>cât timp n&gt;0 execută...</b> presupune utilizarea unei structuri repetitive <b>repetă - cât timp</b> sau <b>repetă - până când</b>:</p> <table border="1" data-bbox="260 913 1278 1361"> <tr> <td> <pre> citește n (număr natural) max&lt;-1 dacă n&gt;0 atunci     repetă         dacă max&lt;n%10 atunci             max&lt;-n%10         n&lt;-[n/10]     cat timp n&gt;0 scrie max </pre> </td> <td> <pre> citește n (număr natural) max&lt;-1 dacă n&gt;0 atunci     repetă         dacă max&lt;n%10 atunci             max&lt;-n%10         n&lt;-[n/10]     pana cand n=0 scrie max </pre> </td> </tr> </table> <p>Pentru alegerea unei structuri repetitive se acordă 2p., pentru testarea condiției de continuare sau oprire se acordă 2p, iar pentru scrierea integrală a restului algoritmului se mai acordă 2p. – în total <b>6p</b>.</p>	<pre> citește n (număr natural) max&lt;-1 dacă n&gt;0 atunci     repetă         dacă max&lt;n%10 atunci             max&lt;-n%10         n&lt;-[n/10]     cat timp n&gt;0 scrie max </pre>	<pre> citește n (număr natural) max&lt;-1 dacă n&gt;0 atunci     repetă         dacă max&lt;n%10 atunci             max&lt;-n%10         n&lt;-[n/10]     pana cand n=0 scrie max </pre>
<pre> citește n (număr natural) max&lt;-1 dacă n&gt;0 atunci     repetă         dacă max&lt;n%10 atunci             max&lt;-n%10         n&lt;-[n/10]     cat timp n&gt;0 scrie max </pre>	<pre> citește n (număr natural) max&lt;-1 dacă n&gt;0 atunci     repetă         dacă max&lt;n%10 atunci             max&lt;-n%10         n&lt;-[n/10]     pana cand n=0 scrie max </pre>		
	<p>d) Pentru instrucțiunile corecte de declarare a variabilelor, de citire a datelor, de afișare a rezultatului și de decizie se acordă câte 1p., pentru instrucțiunea repetitivă se acordă 3p., iar pentru celelalte atribuiri se acordă 2p. Pentru structura corectă a programului se mai acordă 1p., în total <b>10p</b>.</p>		
2.	<p><b>Un răspuns corect:</b></p> <p>Pascal: <b>med:=(x.nota1+x.nota2)/2;</b></p> <p>C/C++: <b>med =(x.nota1+x.nota2)/2.0;</b></p> <p>Pentru orice răspuns corect se acordă <b>6p</b>.</p>		

3.	<p>La finalul executării secvenței date conținutul matricei <b>A</b> este:</p> <pre> 0 4 1 7 5 1 7 1 1 8 1 10 9 1 11 2 </pre> <p>Se observă că suma elementelor de pe diagonala principală este <b>4</b> (<math>=0+1+1+2</math>)</p> <p>Pentru scrierea numărului <b>4</b> se acordă <b>6p.</b></p>
----	---

### Subiectul 3

1.	<p><b>Limbajul Pascal</b></p> <pre> function sub(n,a,b:byte;v:vector):integer; var i:byte;rezultat:integer; begin     i:=a;     while (i&lt;=b) and (v[i] mod 3 &lt;&gt;0) do         if i=b+1 then rezultat:= -1         else rezultat:= i;     end; </pre> <p><b>Limbajul C/C++</b></p> <pre> int sub(int v[100], int n, int a, int b) {int i; i=a; while(i&lt;=b &amp;&amp; a[i]%3!=0) i++; if(i==b+1) return -1; else return i; } </pre> <p>Pentru antet corect se acordă 2p. Pentru condiția divizibilității din tabloul a se acordă 2p. Pentru folosirea corectă a structurii <b>while</b> se dau 2p, iar pentru determinarea poziției se acordă 2p. Pentru corectitudinea globală a subprogramului 2p. – în total <b>10p</b></p>
2.	<p>Pentru rezolvarea acestei probleme ca soluție se propune scrierea unei funcții care primește două șiruri ca parametri și testează dacă acestea rimează. Apoi ar trebui să numărăm pentru primul cuvânt cu ce cuvinte din șir rimează (mai exact cu câte cuvinte care sunt după el în șir rimează).</p> <p>Se pot utiliza funcțiile predefinite pentru șiruri de caractere specifice limbajului.</p> <p>Pentru declararea corectă a variabilelor se acordă 2p. Pentru citirea șirului se acordă 2p. Pentru numărarea aparițiilor sufixelor de două litere identice cu primul sufix se acordă 2p. Pentru afișarea numărului de cuvinte cu proprietatea cerută se acordă 2p.</p> <p>Pentru corectitudinea globală a programului se acordă 2p – în total <b>10p.</b></p> <p><b>Limbajul Pascal</b></p> <pre> var s,t:string; l1,l2,k,n,i:byte; </pre>

		<pre> begin   readln(n);   readln(s);   l1:=length(s);    for i:=2 to n do begin     readln(t);     l2:=length(t);     if (s[l1-1]=t[l2-1]) and (s[l1]=t[l2]) then inc(k);   end;   writeln(k); end. </pre> <p><b>Limbajul C/C++</b></p> <pre> #include &lt;iostream&gt; #include &lt;cstring&gt; using namespace std; int main() {int n;   cin &gt;&gt; n;   char cuv1[21], cuv2[21];   cin &gt;&gt; cuv1;   int s = strlen(cuv1);   int nr = 0;   for(int i = 1; i &lt; n; i++)   { cin &gt;&gt; cuv2;     int s2 = strlen(cuv2);     if(cuv2[s2-2]==cuv1[s-2] &amp;&amp; cuv2[s2-1]==cuv1[s-1])       nr++;   }   cout &lt;&lt; nr;   return 0; } </pre>
3.	b)	<p><b>Algoritmul de rezolvare propus</b> parcurge toate numerele de la radical din <math>x</math> până la radical din <math>y</math>, apoi construiește răsturnatul fiecărui pătrat perfect și verifică dacă și răsturnatul este pătrat perfect.</p> <p>Eficiența algoritmului, ca timp de executare, constă în faptul că parcurgem o singură dată șirul de numere. Ca spațiu de memorie, soluția propusă este eficientă, deoarece utilizează doar variabile simple, fără folosirea unor structuri suplimentare de date.</p> <p><b>Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. – în total 2p.</b></p>
	a)	<p><b>Limbajul Pascal</b></p> <pre> var f:text;     a,b,x,y,r,i,k:longint; begin   assign(f,'bac.in');reset(f);   read(f,x,y);   k:=0; </pre>

```

a:=x;
x:=trunc(sqrt(a));
if x*x<a then inc(x);
y:=trunc(sqrt(y));
for i:=x to y do begin
    a:=i*i; r:=0;
    while a>0 do begin
        r:=r*10+a mod 10;
        a:=a div 10
    end;
    b:=trunc(sqrt(r));
    if (x<=b) and (b<=y) and (b*b=r) then k:=k+1;
end;
writeln(k);
close(f);
end.

```

#### Limbajul C/C++

```

#include <iostream>
#include<fstream>
#include <cmath>
using namespace std;
ifstream f("bac.in");
int main()
{
    int a,b, x,y,i,r,k=0;
    f>>x>>y;
    a=x;
    x=(int)sqrt(a);
    if(x*x<a)x++;
    y=(int)sqrt(y);
    for(int i=x;i<=y;i++)
    {
        a=i*i; r=0;
        while(a>0)
        {
            r=r*10+a%10;
            a=a/10;
        }
        b=(int)sqrt(r);
        if(x<=b && b<=y && b*b==r) k++;
    }
    cout<<k;
    f.close();
    return 0;
}

```

Se acordă 1p. pentru operațiile cu fișiere (declarare, nume corect și deschidere pentru citire), încă 1p. pentru citirea tuturor numerelor din fișier, 1p. pentru un algoritm principal corect, încă 1p. pentru determinarea valorii cerute, 1p. pentru afișarea ei, 1p. pentru corectitudinea formală (declararea variabilelor, structura programului, sintaxa instrucțiunilor, etc.).

**Se acordă 1p. pentru alegerea unui algoritm eficient ca timp de executare și 1p. pentru utilizarea eficientă a memoriei. – în total 8p.**