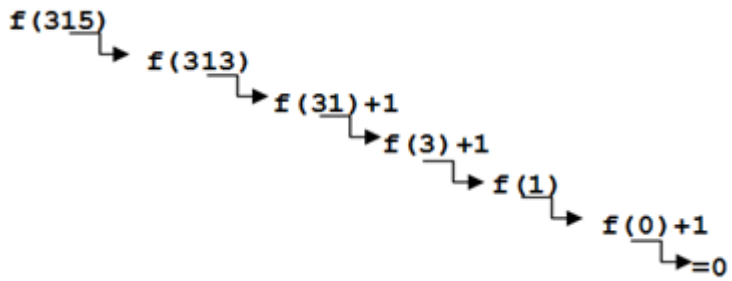
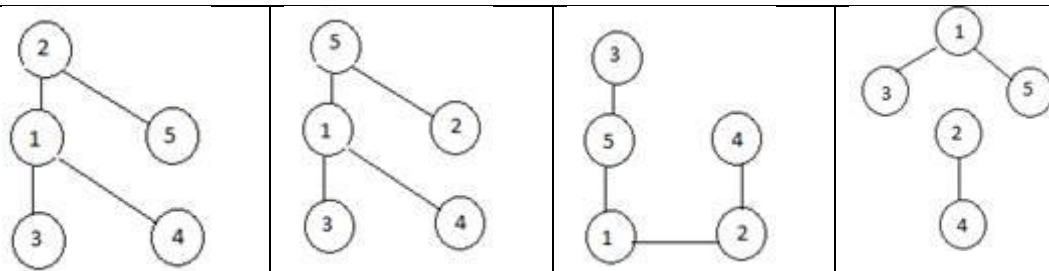
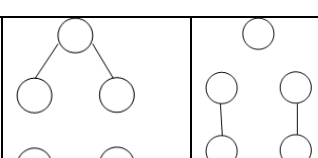


VARIANTA 7 - Rezolvare

SUBIECTUL I

1.	Rezultatul expresiei $x \bmod 5$ (Pascal) sau $x\%5$ (C/C++) este negativ în cazul unui operand x întreg negativ. Pentru precizarea răspunsului a) se acordă 4p.
2.	Se urmăresc apelurile recursive și se fac calculele:  $f(315)=5$ Pentru precizarea răspunsului a) se acordă 4p.
3.	Se generează în maniera "înaintare cu revenire", sumele: $1+2+3+4$, $1+2+7$, $1+3+6$, $1+4+5$, $1+9$, $2+3+5$, $2+8$, ..., etc. Pentru precizarea răspunsului c) se acordă 4p.
4.	Dacă reprezentăm grafic fiecare "arbore" vom obține:  Se observă că ultimul graf nu poate fi arbore întrucât are două componente conexe. Pentru precizarea răspunsului d) se acordă 4p.
5.	Graful complet cu 5 noduri are 10 muchii. Pentru a obține trei componente conexe, numărul maxim de muchii care pot fi eliminate este 8. Un asemenea graf partial ar putea fi reprezentat graphic ca în figurile următoare. Pentru precizarea răspunsului c) se acordă 4p. 

SUBIECTUL al II-lea

1.	<p>a. Se formează un număr c din ultima cifră a fiecărui prefix divizibil cu b al lui a. Pentru precizarea valorii 24 se acordă 6p.</p> <p>b. Se are în vedere că numărul căutat nu trebuie să fie format din prefixe divizibile cu 9. Pentru precizarea valorii 8999, se acordă 6p.</p> <p>c. Se testează capacitatea de a reprezenta pe hârtie algoritmul dat cu ajutorul unui limbaj de programare studiat. Deși scrierea programelor pe hârtie este o activitate improprie la informatică, prin această cerință se valorifică experiențele anterioare de implementare și testare a programelor pe calculator, experiențe care au cimentat cunoștințele privind structura programelor, declararea variabilelor, sintaxa instrucțiunilor programului, regulile de scriere a expresiilor, etc. Pentru instrucțiunile corecte de declarare a variabilelor se acordă 1p., pentru citirea datelor se acordă 1p., pentru afișarea rezultatului se acordă 1p., pentru instrucțiunea de decizie se acordă 2p., pentru instrucțiunea repetitivă se acordă 2p., iar pentru cele 5 atribuiri se acordă 2p. Pentru structura corectă a programului se mai acordă 1p., în total 10p.</p>
----	---

	<p>d. Deoarece valoarea lui a este nenulă, se poate înlocui pur și simplu structura cât timp $a \neq 0$ execută cu o structură repetă...până când. Evident, la înlocuire, se va modifica testul final: Pentru alegerea unei structuri repetitive se acordă 2p., pentru testarea condiției de oprire se acordă 2p., iar pentru scrierea integrală a algoritmului se mai acordă 2p. - în total 6p.</p>	<pre> citește a,b (numere naturale nenule) p←1 c←0 repetă dacă a≠b=0 atunci c←c+a*10*p p←p*10 a←[a/10] până când a=0 scrie c </pre>
2.	<p>Pentru ca datele să coincidă trebuie să aibă aceleași valori pentru zi, lună și an. Dacă e este variabila de tip elev atunci data nașterii elevului este e.d_nt.z,e.d_nt.l, e.d_nt.a, iar data corespunzătoare variabilei dn este dn.z, dn.l, dn.a.</p> <p>Limbaajul Pascal (e.d_nt.z=dn.z) and (e.d_nt.l=dn.l) and (e.d_nt.a=dn.a)</p> <p>Limbaajul C/C++ e.d_nt.z==dn.z&&e.d_nt.l==dn.l&&e.d_nt.a==dn.a</p> <p>Pentru acces la câmpurile variabilei dn se acordă 1p., pentru acces la câmpurile variabilei e se acordă 2p., iar pentru expresie corectă se mai acordă 3 p. - în total 6p.</p>	
3.	<p>În șir, pe poziția 4 se află caracterul '2'. Acesta se înlocuiește cu caracterul '1'. Se șterg mai întâi ultimele două caractere și apoi primele trei caractere.</p> <p>Pentru răspunsul 10 se acordă 6p. (2p. pentru caracterul '1', 1p. pentru caracterul '0' și încă 3p pentru rezultat integral corect).</p>	

SUBIECTUL al III-lea

1.	<p>Prelucrarea oferită de subprogram parcurge tabloul de la primul element la ultimul și verifică dacă elementele sunt pare și în intervalul $[a, b]$.</p> <table><tr><th>Limbaajul Pascal</th><th>Limbaajul C/C++</th></tr><tr><td><pre>type vector=array[1..100] of integer; function pare(n,a,b:integer;v:vector):integer; var i,k:integer; begin k:=0; for i:=1 to n do if (v[i] mod 2=0) and (v[i]>=a) and (v[i]<=b) then k:=k+1; pare:=k; end;</pre></td><td><pre>int pare(int n, int v[], int a, int b) { int i,k=0; for(i=0;i<n;i++) if(v[i]%2==0 && v[i]>=a && v[i]<=b) k++; return k; }</pre></td></tr></table> <p>Pentru respectarea structurii antetului (function/int) se acordă 1p., pentru declararea corectă a parametrilor de intrare n, a și b încă 1p., pentru declararea corectă a parametrului v încă 1p., pentru declararea și inițializarea variabilelor locale încă 1p., iar pentru respectarea structurii subprogramului și a sintaxei limbajului se acordă încă 1p., pentru rezultat încă 2p. Corectitudinea algoritmică a prelucrării în vederea obținerii valorii cerute este notată cu 3p. (paritate, apartenență în interval, incrementare variabilă locală). În total 10p.</p> <p>Facem câteva observații: subprogramul trebuie să se numească pare; În Pascal, parametrul v poate fi transmis prin adresă (var v:vector), deoarece valorile din tablou nu se modifică și este mai eficient ca timp de executare.</p>	Limbaajul Pascal	Limbaajul C/C++	<pre>type vector=array[1..100] of integer; function pare(n,a,b:integer;v:vector):integer; var i,k:integer; begin k:=0; for i:=1 to n do if (v[i] mod 2=0) and (v[i]>=a) and (v[i]<=b) then k:=k+1; pare:=k; end;</pre>	<pre>int pare(int n, int v[], int a, int b) { int i,k=0; for(i=0;i<n;i++) if(v[i]%2==0 && v[i]>=a && v[i]<=b) k++; return k; }</pre>	
Limbaajul Pascal	Limbaajul C/C++					
<pre>type vector=array[1..100] of integer; function pare(n,a,b:integer;v:vector):integer; var i,k:integer; begin k:=0; for i:=1 to n do if (v[i] mod 2=0) and (v[i]>=a) and (v[i]<=b) then k:=k+1; pare:=k; end;</pre>	<pre>int pare(int n, int v[], int a, int b) { int i,k=0; for(i=0;i<n;i++) if(v[i]%2==0 && v[i]>=a && v[i]<=b) k++; return k; }</pre>					
2.	<p>Valorile cu care se completează tabloul bidimensional sunt cifrele impare (1, 3, 5, 7, 9). Trebuie ținut cont de faptul că odată ce valoarea unui element depășește 9, aceasta se resetează la 1.</p> <p>Linile și coloanele matricei se numerează de la 1 (Pascal) sau de la 0 (C/C++).</p> <p>Soluții posibile:</p>					

```

val ← 1
pentru i ← 1, m execută
|
|   pentru j ← 1, n execută
|   |
|   |   dacă val > 9 atunci
|   |   |
|   |   |   val ← 1
|   |   |
|   |   |   ai,j ← val
|   |   |   val ← val + 2
|   |
|   |
|
|

```

Sau se completează tabloul cu ultima cifră a valorii memorate în variabila **val**:

```

val ← 1
pentru i ← 1, m execută
|
|   pentru j ← 1, n execută
|   |
|   |   ai,j ← val % 10
|   |   val ← val + 2
|   |
|   |
|
|

```

Pentru declararea corectă a tabloului se acordă 1p., pentru citirea lui **m** și **n** se mai acordă 1p., pentru accesul la un element din tablou se acordă 1p., pentru parcurgerea tabloului pe linii și coloane se acordă 1p., pentru completarea corectă a tuturor elementelor tabloului se acordă 3p., iar pentru afișarea matricei conform cerinței se acordă 2p. În plus, pentru declararea variabilelor simple, structura și corectitudinea sintactică a programului se mai acordă 1p. - în total **10p.**

3.	a.	Limbajul Pascal	Limbajul C/C++
		<pre> Var f:text; x, k, val,nr, nr_imp:longint; gasit1,gasit2:boolean; begin assign(f,'bac.in'); reset(f); readln(f,x); while not eof(f) do begin read(f,val); if val mod 2 =1 then inc(k); if val=x then if gasit1=false then begin gasit1:=true; k:=0 end else begin nr_imp:=nr_imp+k; k:=0; gasit2:=true end; end; close(f); if gasit2=false then write('nu exista') else write(nr_imp) end. </pre>	<pre> #include <iostream> #include <fstream> using namespace std; ifstream f("bac.in"); int x,gasit1,gasit2,k,val,nr_imp; int main() { f>>x; while(f>>val) { if(val%2)k++; if(val==x) { if(!gasit1) { gasit1=1; k=0; } else { nr_imp+=k; k=0; gasit2=1; } } } f.close(); if(!gasit2) cout<<"nu exista"; else cout<<nr_imp; return 0; } </pre>
		<p>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 tuturor valorilor cerute, 1p. pentru</p>	

		<p>afișarea corectă a rezultatului (mesaj / valoare numerică) și 1p. pentru corectitudinea formală (declararea variabilelor, structura programului, sintaxa instrucțiunilor, etc.). Se acordă 1p. pentru alegerea unui algoritm eficient ca timp de executare ($O(n)$) și 1p. pentru utilizarea eficientă a memoriei. În total 8p.</p> <p>Menționăm că prin algoritm principal corect se înțelege un algoritm care, în intenție, urmărește să rezolve corect problema, deși poate să prezinte scăpări logice sau confuzii.</p>
	b.	<p>O soluție posibilă:</p> <p>Parcurgem șirul și numărăm (într-o variabilă k) valorile impare dintr-o secvență delimitată de două valori x consecutive. La sfârșitul fiecărei astfel de secvențe (cu excepția primeia), vom adăuga numărul k la valoarea cerută în problemă. Variabilele <code>gasit1</code> și <code>gasit2</code> ne ajută să semnalizăm apariția primului x din șir și respectiv a altui x din șir afară de primul.</p> <p>Eficiența algoritmului, ca timp de executare, constă în faptul că parcurgem o singură dată șirul cu un număr impresionant de numere. Deoarece numărul de valori din fișier (1000000) este dimensiunea care contează în problemă, spunem că am obținut un algoritm liniar. Ca spațiu de memorie, soluția propusă este eficientă, deoarece va utiliza doar variabile simple de lucru.</p> <p>Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. - în total 2p.</p> <p>Menționăm că descrierea coerentă a problemei nu presupune să povestim programul, ci ideea de prelucrare.</p>