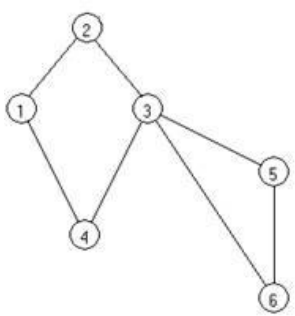
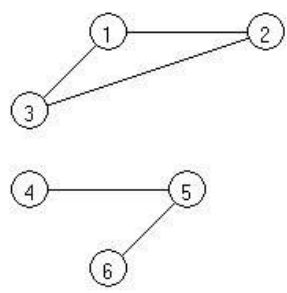
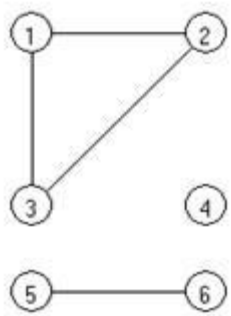
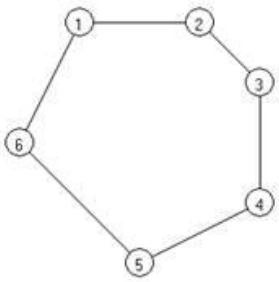


# VARIANTA 1 - Rezolvare

## Subiectul 1

1.	$25 - x * x \geq 0 \Leftrightarrow x * x - 25 \leq 0 \Leftrightarrow (x - 5) * (x + 5) \leq 0 \Leftrightarrow x \in [-5, 5]$ . Pentru precizarea răspunsului <b>b)</b> se acordă <b>4p</b> .	
2.	<p>În desenul următor este prezentat mecanismul prin care se execută apelul <b>ex(25436)</b> al subprogramului recursiv <b>ex</b>.</p> <p>Se vor afișa, în ordine: <b>6345*</b>.</p> <p>Pentru răspunsul a) se acordă <b>4p</b>.</p>	<pre> graph TD     A[ex(25436)] --&gt; B[6]     A --&gt; C[ex(2543)]     C --&gt; D[3]     C --&gt; E[ex(254)]     E --&gt; F[4]     E --&gt; G[ex(25)]     G --&gt; H[5]     G --&gt; I[ex(2)]     I --&gt; J[*]     </pre>
3.	<p>Numerele cerute sunt: <b>2125; 2135; 2145; 2225; 2235; 2245; 2325; 2335; 2345; 2425; 2435; 2445; 2525; 2535; 2545</b>. Sunt generate <b>15</b> numere cu proprietatea din enunț.</p> <p>Pentru răspunsul <b>c)</b> se acordă <b>4p</b>.</p>	
4.	<p>Fiecărui graf îi corespunde o funcție</p> $f: \{(i, j) \mid i \neq j, 1 \leq i, j \leq 5\} \rightarrow \{0, 1\}$ <p>care asociază perechii de noduri distincte <math>(i, j)</math> valoarea:</p> <ul style="list-style-type: none"> <li>• <b>0</b> dacă nu există muchie de la <b>i</b> la <b>j</b></li> <li>• <b>1</b> dacă există muchie de la <b>i</b> la <b>j</b>.</li> </ul> <p>Numărul grafurilor cu câte <b>n</b> vârfuri este egal cu numărul acestor funcții, adică</p> $C_n^2 = 2^{\frac{n(n-1)}{2}}.$ <p>Pentru răspunsul c) se acordă <b>4p</b>.</p>	
5.	<p>Pentru precizarea răspunsului <b>d)</b> se acordă <b>4p</b>.</p>	

<p>a)</p>  <p>Graf conex care conține cicluri (eulerian, nu este hamiltonian)</p>	<p>b)</p>  <p>Graf neconex</p>
<p>c)</p>  <p>Graf neconex</p>	<p>d)</p>  <p>Graf conex aciclic (eulerian și hamiltonian)</p>

## Subiectul 2

1.	<p>a)</p> <p>Din numărul citit sunt eliminate toate cifrele impare iar fiecare cifră pară din număr este mărită cu 1.</p> <p>Pentru precizarea valorii <b>1355</b> se acordă <b>6p</b>.</p>
	<p>b)</p> <p>Valoarea 7 poate fi obținută din numere citite pentru n de forma <b>6x</b> sau <b>x6</b> cu <math>x \in \{1,3,5,7,9\}</math> cifră impară. Algoritmul va afișa valoarea 7 dacă se citește pentru n orice valoare din mulțimea {16, 36, 56, 76, 96, 61, 63, 65, 67, 69}.</p> <p>Pentru precizarea valorilor <b>16,36,56,76,96,61,63,65,67,69</b> se acordă <b>6p</b>. Pentru identificarea valorilor corecte cu adăugarea sau repetarea unora dintre valori se acordă 5p. Pentru identificarea a cel puțin 5 valori corecte se acordă 3p., iar pentru identificarea a cel puțin două valori corecte se acordă 1p. În celelalte cazuri se acordă 0p.</p>
	<p>c)</p> <p>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="331 197 1209 495"> <tr> <td> dacă <math>n &gt; 0</math> atunci    repetă      <math>c \leftarrow n \% 10</math>; <math>n \leftarrow [n/10]</math>      dacă <math>c \% 3 = 0</math> atunci        <math>z \leftarrow z + p * (9 - c)</math>        <math>p \leftarrow p * 10</math>      cât timp <math>n &gt; 0</math> </td><td> dacă <math>n &gt; 0</math> atunci    repetă      <math>c \leftarrow n \% 10</math>; <math>n \leftarrow [n/10]</math>      dacă <math>c \% 3 = 0</math> atunci        <math>z \leftarrow z + p * (9 - c)</math>        <math>p \leftarrow p * 10</math>      până când <math>n = 0</math> </td></tr> </table> <p>Fără testarea inițială a valorii lui <math>n</math> (<math>n &gt; 0</math>), algoritmul obținut prin înlocuirea mecanică a structurii <b>cât timp</b> cu structura <b>repetă...</b> va afișa pentru <math>n = 0</math> valoarea <b>1</b> spre deosebire de algoritmul din enunț care afișează valoarea 0 pentru <math>n = 0</math>, deci cei doi algoritmi nu sunt echivalenți.</p> <p>Pentru alegerea unei structuri repetitive se acordă 2p., pentru testarea inițială a valorii lui <math>n</math> se acordă 2p., pentru testarea condiției de continuare sau oprire se acordă 1p, iar pentru scrierea integrală a restului algoritmului se mai acordă 1p. - în total <b>6p</b>.</p>	dacă $n > 0$ atunci repetă $c \leftarrow n \% 10$ ; $n \leftarrow [n/10]$ dacă $c \% 3 = 0$ atunci $z \leftarrow z + p * (9 - c)$ $p \leftarrow p * 10$ cât timp $n > 0$	dacă $n > 0$ atunci repetă $c \leftarrow n \% 10$ ; $n \leftarrow [n/10]$ dacă $c \% 3 = 0$ atunci $z \leftarrow z + p * (9 - c)$ $p \leftarrow p * 10$ până când $n = 0$		
dacă $n > 0$ atunci repetă $c \leftarrow n \% 10$ ; $n \leftarrow [n/10]$ dacă $c \% 3 = 0$ atunci $z \leftarrow z + p * (9 - c)$ $p \leftarrow p * 10$ cât timp $n > 0$	dacă $n > 0$ atunci repetă $c \leftarrow n \% 10$ ; $n \leftarrow [n/10]$ dacă $c \% 3 = 0$ atunci $z \leftarrow z + p * (9 - c)$ $p \leftarrow p * 10$ până când $n = 0$				
d)	<p>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, privind structura programelor, declararea variabilelor, sintaxa instrucțiunilor programului, regulile de scriere a expresiilor, etc.</p> <p>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 cele 6 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><b>Pascal:</b> <math>A.x * A.x + A.y * A.y = B.x * B.x + B.y * B.y</math></p> <p><b>C/C++:</b> <math>A.x * A.x + A.y * A.y == B.x * B.x + B.y * B.y</math></p> <p>Pentru orice răspuns corect se acordă <b>6p</b>.</p>				
3.	<p>Un răspuns corect:</p> <table border="1" data-bbox="256 1659 1385 1921"> <thead> <tr> <th>Limbajul Pascal</th><th>Limbajul C/C++</th></tr> </thead> <tbody> <tr> <td> <math>s := 'bac2723denota37'</math>;  for <math>i := 1</math> to <math>length(s)</math> do    if <math>s[i] = '7'</math> then write(0)    else if <math>s[i] = '3'</math> then write(1)    else write(<math>s[i]</math>); </td><td> <math>strcpy(s, "bac2723denota37")</math>;  for (<math>i = 0</math>; <math>i &lt; strlen(s)</math>; <math>i++</math>)    if (<math>s[i] == '7'</math>) cout &lt;&lt; 0;    else if (<math>s[i] == '3'</math>) cout &lt;&lt; 1;    else cout &lt;&lt; <math>s[i]</math>; </td></tr> </tbody> </table>	Limbajul Pascal	Limbajul C/C++	$s := 'bac2723denota37'$ ; for $i := 1$ to $length(s)$ do if $s[i] = '7'$ then write(0) else if $s[i] = '3'$ then write(1) else write( $s[i]$ );	$strcpy(s, "bac2723denota37")$ ; for ( $i = 0$ ; $i < strlen(s)$ ; $i++$ ) if ( $s[i] == '7'$ ) cout << 0; else if ( $s[i] == '3'$ ) cout << 1; else cout << $s[i]$ ;
Limbajul Pascal	Limbajul C/C++				
$s := 'bac2723denota37'$ ; for $i := 1$ to $length(s)$ do if $s[i] = '7'$ then write(0) else if $s[i] = '3'$ then write(1) else write( $s[i]$ );	$strcpy(s, "bac2723denota37")$ ; for ( $i = 0$ ; $i < strlen(s)$ ; $i++$ ) if ( $s[i] == '7'$ ) cout << 0; else if ( $s[i] == '3'$ ) cout << 1; else cout << $s[i]$ ;				

### Subiectul 3

1. Prelucrarea oferită de subprogram verifică existența celor două cifre în scrierea fiecărui număr memorat în tablou. Ștergerea unui element care conține cifra **a** și nu conține cifra **b** se realizează prin deplasarea cu o poziție la stânga a tuturor elementelor situate la dreapta celui ce urmează a fi șters. După realizarea deplasării, se micșorează valoarea lui **n** cu 1.

#### Limbajul Pascal

```
type vector=array[1..100]of longint;
procedure sub(a:longint; b:byte;
              var n:byte; var v:vector);
var exa,exb:boolean;
    uc,i,j:byte;
    z:longint;
begin
    i:=1;
    while i<=n do begin
        exa:=false; exb:=false; z:=v[i];
        while z>0 do
            begin
                uc:=z mod 10;
                z:=z div 10;
                if uc=a then exa:=true
                    else if uc=b then exb:=true;
            end;
        if exa=true and exb=false then
            begin
                for j:=i+1 to n do v[j-1]:=v[j];
                n:=n-1
            end
        else inc(i);
    end
end;
```

#### Limbajul C/C++

```
void sub(int a, int b, int &n, int v[])
{
    int i=1,uc,exa,exb,z,j;
    while(i<=n)
    {
        exa=0; exb=0; z=v[i];
        while(z>0)
        {
            uc=z%10;
            z=z/10;
            if (uc==a) exa=1;
            else if(uc==b) exb=1;
        }
        if (exa && !exb)
        {
            for(j=i+1;j<=n;j++) v[j-1]=v[j];
            n--;
        }
        else i++;
    }
}
```

	<p>Pentru antet corect se acordă 3p. (Pentru respectarea structurii antetului (procedure/void) se acordă 1p., pentru declararea corectă a parametrilor de intrare a și b încă 1p., pentru declararea corectă a parametrilor <b>n</b> și <b>v</b> încă 1p.).</p> <p>Pentru declararea variabilelor locale încă 1p., iar pentru respectarea structurii subprogramului și a sintaxei limbajului se acordă încă 1p.</p> <p>Pentru determinarea numărului de apariții ale cifrei a în scrierea unui număr din tabloului 1p. Pentru determinarea numărului de apariții ale cifrei <b>b</b> în scrierea unui număr din tabloului 1p. Pentru determinarea elementelor tabloului cu proprietatea cerută 1p. Pentru ștergerea din tablou a tuturor elementelor ce memorează numerele cu proprietatea cerută 1p. Pentru corectitudinea globală a programului 1p.</p> <p>În total <b>10p.</b></p>
2.	<p>Pot fi concepute mai multe metode de obținere a matricei cerute. Metoda clasică ce completează coloanele matricei (numerotate de la <b>1</b> la <b>m</b>), respectiv liniile (numerotate de la <b>1</b> la <b>n</b>), în maniera descrisă, este:</p> <pre> <b>pentru j</b>←1,m execută     <b>k</b>← j     <b>pentru i</b>←1,n execută         A[i,j]←k         <b>k</b>← k+1     └─ └─ </pre> <p>Pentru declararea corectă a matricei se acordă 1p., pentru citirea lui <b>n</b> și <b>m</b> se mai acordă 1p., pentru parcurgerea unei coloane sau linii se acordă 1p., pentru completarea corectă a unei coloane se acordă 2p., pentru completarea corectă a tuturor coloanelor se acordă 2p., 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 <b>10p.</b></p>
3.	<p><b>b)</b> <b>Algoritmul de rezolvare propus</b> citește, unul câte unul, numerele din fișier, analizând paritatea fiecăruia: întâlnirea unui număr par va duce la "prelungirea" secvenței curente, iar întâlnirea unui număr impar va conduce la trecerea la o nouă secvență. La finele oricărei secvențe se analizează lungimea ei, în vederea memorării lungimii maxime.</p> <p>Eficiența algoritmului, ca timp de executare, constă în faptul că parcurgem o singură dată șirul. 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 utilizează doar variabile simple.</p> <p>Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. - în total <b>2p.</b></p>
	<p><b>a)</b></p> <p style="text-align: center;"><b>Limbajul Pascal</b></p> <pre> <b>Var f:text;</b>         <b>nr,lgmax,lg:longint;</b> </pre>

```

begin
    assign(f,'bac.in');reset(f);
    while not eof(f) do begin
        read(f,nr);
        if nr mod 2=0 then
            inc(lg)
        else begin
            if lgmax<lg then
                lgmax:=lg;
            lg:=0
        end;
    end;
    if lgmax<lg then
        lgmax:=lg;
    close(f);
    write(lgmax)
end.

```

### Limbajul C/C++

```

#include
#include
using namespace std;
ifstream f("bac.in");
int main()
{
    int lgmax,lg,nr;
    lg=lgmax=0;
    while(f>>nr)
        if(nr%2==0) lg++;
        else {
            if(lgmax<lg) lgmax=lg;
            lg=0;
        }
    if(lgmax<lg) lgmax=lg;
    cout<<lgmax;
    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 ( $O(n)$ ) și 1p. pentru utilizarea eficientă a memoriei ( $O(1)$ ). În total **8p.**