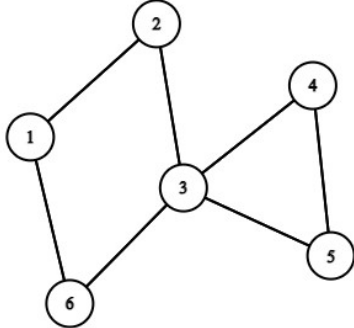


## VARIANTA 4 - Rezolvare

### Subiectul 1

1.	Un număr real $x \in [1,5] \cup [7,9]$ dacă și numai dacă $x \notin (-\infty, 1) \cup (5,7) \cup (9, +\infty)$ , deci răspunsul corect este <b>d</b> ).
2.	Deoarece variabila <b>e</b> de tip <b>Elev</b> conține data nașterii elevului respectiv în câmpul <b>data_nașterii</b> de tip <b>Data</b> , rezultă că data nașterii elevului respectiv (adică ziua, luna și anul) poate fi accesată prin expresia <b>e.data_nașterii</b> , deci anul nașterii sale poate fi accesat prin expresia <b>e.data_nașterii.an</b> . În concluzie, răspunsul corect este <b>b</b> ).
3.	Folosind metoda backtracking, primele <b>13</b> numere formate din <b>n=4</b> cifre distincte și având suma cifrelor <b>s=25</b> care se vor genera sunt 1789, 1798, 1879, 1897, 1978, 1987, 2689, 2698, 2869, 2896, 2968, 2986 și <b>3589</b> , deci răspunsul corect este <b>c</b> ).
4.	Deoarece graful neorientat $G$ este complet și are $n = 10$ vârfuri, înseamnă că el este conex și are $\frac{n(n-1)}{2} = 45$ muchii. Deoarece un graf conex minimal (adică un graf care prin eliminarea unei singure muchii nu mai este conex) este un arbore, rezultă că graful $G$ trebuie să conțină cel puțin $n - 1$ muchii (orice arbore cu $n$ vârfuri are $n - 1$ muchii!), adică se pot elimina maxim $45 - 9 = 36$ de muchii, deci răspunsul corect este <b>c</b> ).
5.	<p>În figura alăturată este reprezentat graful dat în enunț. Se observă ușor faptul că răspunsul corect este <b>a</b>).</p> 

### Subiectul 2

1.	<p><b>a)</b> Algoritmul calculează câte numere prime există între <b>a</b> și <b>b</b>, deoarece variabila <b>d</b> va conține fie cel mai mare divizor propriu al unui număr <b>i</b> cuprins între <b>a</b> și <b>b</b>, fie valoarea <b>0</b> dacă numărul <b>i</b> nu are niciun divizor propriu (ceea ce înseamnă că numărul <b>i</b> este prim!). Pentru precizarea valorii <b>3</b> (între <b>7</b> și <b>13</b> există <b>3</b> numere prime: <b>7</b>, <b>11</b> și <b>13</b>) se acordă <b>6p.</b>, iar pentru orice altă valoare se acordă <b>0p.</b></p> <p><b>b)</b> Deoarece <b>a=10</b>, înseamnă că trebuie determinată cea mai mare valoare posibilă pentru <b>b</b> astfel încât între <b>10</b> și <b>b</b> să existe <b>4</b> numere prime. Se observă ușor faptul că între <b>10</b> și <b>19</b> există <b>4</b> numere prime (<b>11</b>, <b>13</b>, <b>17</b> și <b>19</b>), numerele <b>20</b>, <b>21</b> și <b>22</b> nu sunt prime, iar numărul <b>23</b> este prim, deci cea mai mare valoare posibilă pentru <b>b</b> astfel încât între <b>10</b> și <b>b</b> să existe <b>4</b> numere prime este <b>22</b>. Pentru precizarea valorii <b>22</b> se acordă <b>6p.</b>, iar pentru orice altă valoare se acordă <b>0p.</b></p>
----	---

	<p>c) Structura repetitivă pentru <math>j &lt; -2, [i/2]</math> <b>execută</b> se poate înlocui cu una dintre structurile repetitive <b>cât timp... execută</b> sau <b>repetă... până când</b>, având grijă la inițializarea, testarea și incrementarea variabilei <math>j</math>:</p> <div style="display: flex; justify-content: space-around;"> <pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> citește a,b nr ← 0 pentru i ← a,b execută   d ← 0   j ← 2   cât timp j ≤ [i/2] execută     dacă i%j = 0 atunci       d ← j     j ← j+1   dacă d = 0 atunci     nr ← nr+1 scrie nr </pre> <pre style="background-color: #f0f0f0; padding: 10px; border: 1px solid #ccc;"> citește a,b nr ← 0 pentru i ← a,b execută   d ← 0   j ← 2   repetă     dacă i%j = 0 atunci       d ← j     j ← j+1   până când j &gt; [i/2]   dacă d = 0 atunci     nr ← nr+1 scrie nr </pre> </div> <p>Atragem atenția asupra faptului că structura repetitivă <b>cât timp...execută</b> asigură echivalența cu structura <b>pentru...execută</b> în toate situațiile, pe când structura <b>repetă...până când</b> este echivalentă doar în unele situații, printre care și cea din acest algoritm!</p> <p>Pentru alegerea unei structuri repetitive se acordă <b>2p.</b>, pentru inițializarea lui <math>j</math> se acordă <b>1p.</b>, pentru testarea condiției de continuare sau oprire se acordă <b>3p.</b>, pentru incrementarea lui <math>j</math> se acordă <b>1p.</b>, iar pentru scrierea integrală a restului algoritmului se mai acordă <b>3p.</b>, deci se obțin, în total, <b>10p.</b></p> <p>d) Se testează capacitatea de a reprezenta pe hârtie algoritmul dat folosind un limbaj de programare studiat. Pentru instrucțiunile corecte de declarare a variabilelor, de citire a datelor, de afișare a rezultatului și de decizie se acordă câte <b>0.5p.</b>, pentru cele două instrucțiuni repetitive se acordă <b>2p.</b>, iar pentru cele 4 atribuiri se acordă <b>1p.</b> Pentru structura corectă a programului se mai acordă <b>1p.</b>, deci se obțin, în total, <b>6p.</b></p>				
2.	<p>Pentru a testa dacă șirul <math>s</math> este un prefix al șirului <math>t</math> trebuie să verificăm dacă șirul <math>s</math> apare ca subșir în șirul <math>t</math> începând cu prima poziție (în limbajul Pascal) sau de la adresa primului caracter (în limbajele C/C++):</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Limbajul Pascal</th><th style="text-align: center;">Limbajul C++</th></tr> </thead> <tbody> <tr> <td> <pre> if pos(s,t) = 1 then   writeln('DA') else   writeln('NU'); </pre> </td><td> <pre> if(strstr(t, s) == s)   cout&lt;&lt;"DA"; else   cout&lt;&lt;"NU"; </pre> </td></tr> </tbody> </table> <p>Pentru utilizarea funcției corespunzătoare se acordă <b>3p.</b>, pentru afișarea șirului de caractere cerut se acordă <b>2p.</b>, iar pentru corectitudinea sintactică a secvenței se mai acordă <b>1p.</b>, deci se obțin, în total, <b>6p.</b></p>	Limbajul Pascal	Limbajul C++	<pre> if pos(s,t) = 1 then   writeln('DA') else   writeln('NU'); </pre>	<pre> if(strstr(t, s) == s)   cout&lt;&lt;"DA"; else   cout&lt;&lt;"NU"; </pre>
Limbajul Pascal	Limbajul C++				
<pre> if pos(s,t) = 1 then   writeln('DA') else   writeln('NU'); </pre>	<pre> if(strstr(t, s) == s)   cout&lt;&lt;"DA"; else   cout&lt;&lt;"NU"; </pre>				

3.	În urma apelului $f(24, 2)$ se va obține valoarea 5, deoarece $f(24, 2) = 2 + f(12, 3) = 2 + 3 + f(4, 4) = 2 + 3 + 0 = 5$ . Pentru precizarea valorii 5 se acordă 6p., iar pentru orice altă valoare se acordă 0p.
----	---

### Subiectul 3

1.	<p>Subprogramul construiește valoarea cerută cifră cu cifră, adăugând la numărul <b>b</b> (inițial egal cu 0) doar cifrele numărului <b>a</b> care se găsesc pe poziții impare. Numerotarea cifrelor lui <b>a</b> se realizează de la dreapta spre stânga, folosind variabila auxiliară <b>p</b>, începând cu poziția 0 (corespunzătoare cifrei unităților). Deoarece cifrele lui <b>a</b> sunt accesibile de la ultima către prima, trebuie utilizate înmulțiri cu puteri consecutive ale lui 10.</p> <table border="1"> <thead> <tr> <th>Limbajul Pascal</th><th>Limbajul C++</th></tr> </thead> <tbody> <tr> <td> <pre> <b>procedure</b> cifre(<b>a</b>:longint;                 <b>var</b> <b>b</b>:longint); <b>var</b> <b>uc</b>,<b>p</b>:byte; <b>z</b>:longint; <b>begin</b>     <b>b</b>:=0;     <b>z</b>:=1;     <b>p</b>:=0;     <b>while</b> <b>a</b>&gt;0 <b>do</b>         <b>begin</b>             <b>uc</b>:=<b>a</b> mod 10;             <b>if</b> <b>p</b> mod 2=1 <b>then</b>                 <b>begin</b>                     <b>b</b>:=<b>b</b>+<b>uc</b>*<b>z</b>;                     <b>z</b>:=<b>z</b>*10                 <b>end</b>;             <b>a</b>:=<b>a</b> div 10;             <b>p</b>:=<b>p</b>+1         <b>end</b>     <b>end</b>; end; </pre> </td><td> <pre> <b>void</b> cifre(<b>int</b> <b>a</b>,<b>int</b> &amp;<b>b</b>) {     <b>int</b> <b>uc</b>,<b>p</b>,<b>z</b>;     <b>b</b>=0;     <b>z</b>=1;     <b>p</b>=0;     <b>while</b> (<b>a</b>&gt;0)     {         <b>uc</b>=<b>a</b>%10;         <b>if</b> (<b>p</b>%2==1)         {             <b>b</b>=<b>b</b>+<b>uc</b>*<b>z</b>;             <b>z</b>=<b>z</b>*10;         }         <b>a</b>/=10;         <b>p</b>++;     } } </pre> </td></tr> </tbody> </table> <p>Pentru respectarea structurii antetului (<b>procedure/void</b>) se acordă 1p., pentru declararea corectă a parametrului de intrare <b>a</b> încă 1p., pentru declararea corectă a parametrului de ieșire <b>b</b> încă 1p., pentru declararea variabilelor locale încă 1p., iar pentru respectarea structurii subprogramului și a sintaxei limbajului se acordă încă 1p. Corectitudinea algoritmică a prelucrării în vederea obținerii valorii cerute este notată cu 4p. Pentru transmiterea naturală a rezultatului prin parametrul <b>b</b> se acordă 1p, aceasta însemnând că valoarea lui <b>b</b> nu se afișează și nu se returnează explicit. Astfel, în total, se obțin 10p.</p>	Limbajul Pascal	Limbajul C++	<pre> <b>procedure</b> cifre(<b>a</b>:longint;                 <b>var</b> <b>b</b>:longint); <b>var</b> <b>uc</b>,<b>p</b>:byte; <b>z</b>:longint; <b>begin</b>     <b>b</b>:=0;     <b>z</b>:=1;     <b>p</b>:=0;     <b>while</b> <b>a</b>&gt;0 <b>do</b>         <b>begin</b>             <b>uc</b>:=<b>a</b> mod 10;             <b>if</b> <b>p</b> mod 2=1 <b>then</b>                 <b>begin</b>                     <b>b</b>:=<b>b</b>+<b>uc</b>*<b>z</b>;                     <b>z</b>:=<b>z</b>*10                 <b>end</b>;             <b>a</b>:=<b>a</b> div 10;             <b>p</b>:=<b>p</b>+1         <b>end</b>     <b>end</b>; end; </pre>	<pre> <b>void</b> cifre(<b>int</b> <b>a</b>,<b>int</b> &amp;<b>b</b>) {     <b>int</b> <b>uc</b>,<b>p</b>,<b>z</b>;     <b>b</b>=0;     <b>z</b>=1;     <b>p</b>=0;     <b>while</b> (<b>a</b>&gt;0)     {         <b>uc</b>=<b>a</b>%10;         <b>if</b> (<b>p</b>%2==1)         {             <b>b</b>=<b>b</b>+<b>uc</b>*<b>z</b>;             <b>z</b>=<b>z</b>*10;         }         <b>a</b>/=10;         <b>p</b>++;     } } </pre>
Limbajul Pascal	Limbajul C++				
<pre> <b>procedure</b> cifre(<b>a</b>:longint;                 <b>var</b> <b>b</b>:longint); <b>var</b> <b>uc</b>,<b>p</b>:byte; <b>z</b>:longint; <b>begin</b>     <b>b</b>:=0;     <b>z</b>:=1;     <b>p</b>:=0;     <b>while</b> <b>a</b>&gt;0 <b>do</b>         <b>begin</b>             <b>uc</b>:=<b>a</b> mod 10;             <b>if</b> <b>p</b> mod 2=1 <b>then</b>                 <b>begin</b>                     <b>b</b>:=<b>b</b>+<b>uc</b>*<b>z</b>;                     <b>z</b>:=<b>z</b>*10                 <b>end</b>;             <b>a</b>:=<b>a</b> div 10;             <b>p</b>:=<b>p</b>+1         <b>end</b>     <b>end</b>; end; </pre>	<pre> <b>void</b> cifre(<b>int</b> <b>a</b>,<b>int</b> &amp;<b>b</b>) {     <b>int</b> <b>uc</b>,<b>p</b>,<b>z</b>;     <b>b</b>=0;     <b>z</b>=1;     <b>p</b>=0;     <b>while</b> (<b>a</b>&gt;0)     {         <b>uc</b>=<b>a</b>%10;         <b>if</b> (<b>p</b>%2==1)         {             <b>b</b>=<b>b</b>+<b>uc</b>*<b>z</b>;             <b>z</b>=<b>z</b>*10;         }         <b>a</b>/=10;         <b>p</b>++;     } } </pre>				

2.	Există mai multe metode de obținere a matricei cerute, una dintre ele fiind următoarea (am considerat faptul că liniile și coloane matricei sunt numerotate de la 0):
----	---

```

k ← 1
pentru i ← 0,n-1 execută
    dacă i%2 = 0
        atunci
            pentru j ← 0,n-1 execută
                a[i][j] ← k
                k ← k+1
            ■
        altfel
            pentru j ← n-1,0,-1 execută
                a[i][j] ← k
                k ← k+1
            ■
    ■
■

```

Pentru declararea corectă a matricei se acordă **1p.**, pentru citirea lui **n** se mai acordă **1p.**, pentru parcurgerea unei linii se acordă **1p.**, pentru completarea corectă a unei linii (pară+impară) se acordă **1p.+1p.**, pentru completarea corectă a tuturor liniilor (pare+impare) se acordă **1p.+1p.**, 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.**, deci se obțin, în total, **10p.**

3. b) Vom utiliza un vector de frecvențe **fc** cu **10** elemente pentru a contoriza aparițiile fiecărei cifre. Vom parcurge fișierul **bac.in** caracter cu caracter, folosind variabila **c** de tip **char**, și, în cazul în care variabila **c** conține o cifră, vom crește frecvența sa cu **1**. Cel mai mare număr care se poate forma folosind cifrele tuturor numerelor din fișier se obține scriind în fișierul **bac.out** fiecare cifră **c**, de la **9** la **0**, de un număr de ori egal cu frecvența sa.
- Eficiența algoritmului din punct de vedere al timpului de executare constă în faptul că parcurgem o singură dată fișierul de intrare **bac.in**, obținând astfel un algoritm liniar în raport cu numărul de cifre din fișier. Eficiența algoritmului din punct de vedere al memoriei utilizate constă în faptul că se va folosi doar un vector de frecvențe cu **10** componente, evitând astfel memorarea tuturor celor maximum **9000000** de cifre din fișier într-un vector de caractere.
- Pentru o descriere coerentă a metodei se acordă **1p.**, iar pentru justificarea eficienței, încă **1p.**, deci se obțin, în total **2p.**

a) **Limbajul Pascal**

```

var fin,fout:text;
    c:char;
    i,j:longint;
    fc:array[0..9]of byte;
begin
    for i:=0 to 9 do
        fc[i]:=0;

    assign(fin,'bac.in');
    reset(fin);

    while(not eof(fin)) do
        begin

```

```

        read(fin,c);
        if((c>='0')and(c<='9')) then
            inc(fc[ord(c)-ord('0')]);
        end;

        close(fin);

        assign(fout,'bac.out');
        rewrite(fout);

        for i:=9 downto 0 do
            for j:=1 to fc[i] do
                write(fout,i);

            close(fout);
        end.

```

### Limbajul C++

```

#include <fstream>

using namespace std;

ifstream fin("bac.in");
ofstream fout("bac.out");

int main()
{
    int fc[10],i,j;
    char c;

    for(i=0;i<10;i++)
        fc[i]=0;

    while(fin>>c)
        if((c>='0')&&(c<='9'))
            fc[c-'0']++;

    fin.close();

    for(i=9;i>=0;i--)
        for(j=0;j<fc[i];j++)
            fout<<i;

    fout.close();

    return 0;
}

```

Se acordă **1p.** pentru operațiile cu fișiere (declarare, nume corect și deschidere pentru citire), încă **1p.** pentru citirea tuturor cifrelor din fișier, **1p.** pentru un algoritm corect, încă **2p.** pentru determinarea numărului cerut și **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, deci se obțin, în total, **8p.**