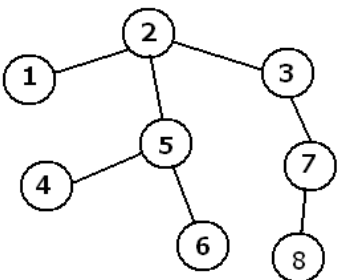


VARIANTA 10 - Rezolvare

Subiectul 1

1.	Se accesează valorile memorate în variabilele x și y . Pentru precizarea răspunsului d) se acordă 4p .
2.	Se urmăresc apelurile recursive și se fac calculele. Se compară două câte două și se va afișa valoarea maximă dintre cifre. Pentru răspunsul b) se acordă 4p .
3.	Se generează în aceeași manieră ca în exemplul din enunț. Pentru răspunsul c) se acordă 4p .
4.	Frunza = într-un arbore cu rădăcina nodul x este frunză dacă nu are nici un descendent direct. Observăm că nodurile 1,4,6,8 sunt frunze.  <pre> graph TD 2((2)) --- 1((1)) 2 --- 3((3)) 1 --- 5((5)) 3 --- 7((7)) 5 --- 4((4)) 5 --- 6((6)) 7 --- 8((8)) </pre> <p>Pentru precizarea răspunsului c) se acordă 4p.</p>
5.	Observăm că se pot elimina maxim 3 muchii. Putem elimina muchiile (1,3), (2,4), (5,3) și de aici rezulta un graf conex. Pentru precizarea răspunsului c) se acordă 4p .

Subiectul 2

1.	a)	Algoritmul calculează numărul obținut din cifrele lui pare aflate pe poziții impare, numărarea pozițiilor cifrelor începând cu cifra cea mai semnificativă. Pentru precizarea valorii 28 se acordă 6p .
	b)	Pentru variabila n poate fi citit orice număr natural de cinci cifre care are pe una din pozițiile impare, o singură cifră egală cu valoarea 4 . Pe celelalte 2 poziții pot fi oricare alte 2 cifre impare. Un exemplu de număr: 13475. Pentru precizarea valorii corecte se acordă 6p .
	c)	Înlocuirea structurii cât timp n>0 execută ... presupune utilizarea unei structuri repetitive repetă – cât timp sau, repetă - până când :

	<pre> citește n(număr natural) r<-0; daca n>0 atunci repetă r<-r*10 + n%10; n<-[n/10] cât timp n>0 repetă dacă r%2=0 atunci n<-n*10 +r%10 ■ r<-[r/100] cât timp r>0 ■ scrie n </pre>	<pre> citește n(număr natural) r<-0; daca n>0 atunci repetă r<-r*10 + n%10; n<-[n/10] până când n=0 repetă dacă r%2=0 atunci n<-n*10 +r%10 ■ r<-[r/100] până când r=0 ■ scrie n </pre>	<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 6p.</p>
	<p>d) 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 instrucțiunile repetitive se acordă 3p., iar pentru celelalte atribuiri se acordă 2p. Pentru structura corectă a programului se mai acordă 1p., în total 10p.</p>		
2.	<p>a==b && x==y</p> <p>Pentru precizarea răspunsului corect se acordă 6p.</p>		
3.	<p>În urma executării instrucțiunilor din secvența dată se vor afișa prefixele cuvântului :</p> <p>c ca cal calc calcu calcul</p> <p>Pentru precizarea răspunsului corect se acordă 6p.</p>		

Subiectul 3

1.	<div data-bbox="236 286 1283 320" style="background-color: #e0e0e0;">Limbajul Pascal</div> <pre>Type vector=array[1..100] of integer; function suma(n:byte;v:vector):integer; var ult,pult,i:byte; s:integer; begin s:=0; for i:=1 to n do begin ult:=v[i] mod 10; pult:=v[i] div 10 mod 10; if (ult mod 2=0) and (pult mod 2=0) then s:=s+v[i]; end; suma:=s; end;</pre> <div data-bbox="236 902 1238 947" style="background-color: #e0e0e0;">Limbajul C/C++</div> <pre>int suma(int n,int v[]) { int i,s=0,ult,pult; for(i=1;i<=n;i++) { ult=v[i]%10; pult=v[i]/10%10; if (pult%2==0 && ult%2==0) s+=v[i]; } return s; }</pre> <p>Pentru antet corect se acordă 2p. Pentru condiția divizibilității din tabloul v se acordă 2p. Pentru folosirea corectă a structurii for se dau 2p, iar pentru determinarea ultimelor două cifre se acordă 2p. Pentru corectitudinea globală a subprogramului 2p.</p> <p>În total 10p.</p>
2.	<p>Pentru declararea corectă a matricei se acordă 1p., pentru citirea lui n se acordă 1p., pentru completarea corectă a fiecărui element din matrice aflat pe o linie impară 2p., pentru completarea corectă a fiecărui element din matrice aflat pe o linie pară 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 2p. - în total 10p.</p> <p>Pot fi concepute mai multe metode de obținere a matricei cerute. Metoda clasică ce completează coloanele matricei, cu liniile, respectiv coloanele, numerotate de la 1 la n, respectiv de la 1 la n, în maniera descrisă, este:</p> <div data-bbox="236 1917 1283 1951" style="background-color: #e0e0e0;">Limbajul Pascal</div> <pre>var a: array[1..30,1..30] of integer; n,i,j:byte;</pre>

	<pre> begin for i:=1 to n do for j:=1 to n do if i mod 2<>0 then a[i,j]:=i*j else a[i,j]:=j*j+a[i-1,j]; for i:=1 to n do begin writeln; for j:=1 to n do write(a[i,j], ' ') end end end. </pre> <p>Limbajul C/C++</p> <pre> #include<iostream> using namespace std; int main() { int i,j,n,a[30][30]; cin>>n; for(i=1;i<=n;i++) {for(j=1;j<=n;j++) if(i%2!=0) a[i][j]=i*j; else a[i][j]=j*j + a[i-1][j]; for(i=1;i<=n;i++) {for(j=1;j<=n;j++) cout<<a[i][j]<<" "; cout<<endl; } return 0; } </pre>
3.	<p>b) Se citesc pe rând valorile din fișier, verificăm dacă se îndeplinește proprietatea cerută și în caz afirmativ, îl comparăm cu nr2 respectiv cu nr1, înlocuindu-le când e cazul.</p> <p>Eficiența algoritmului, ca timp de executare, constă în faptul că parcurgem o singură dată șirul de numere (complexitate $O(n)$). Ca spațiu de memorie, soluția propusă este eficientă, deoarece utilizează doar variabile simple, fără folosirea unor structuri suplimentare de date (complexitate $O(1)$).</p> <p>Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. – în total 2p.</p>
	<p>a)</p> <p>Limbajul Pascal</p> <pre> var x,nr1,nr2:longint; f:text; </pre>

```

begin
  assign (f,'bac.in');reset(f);
  nr1:=0;
  nr2:=0;
  while not eof(f) do
    begin read(f,x);
      if x mod 10 mod 3=0 then
        if nr2<x then
          begin
            nr1:=nr2;
            nr2:=x;
          end
        else
          if nr1<x then
            nr1:=x;
          end;
        write(nr1,' ',nr2);
      close(f);
    end.
end.

```

Limbajul C/C++

```

#include<iostream>
#include<fstream>
using namespace std;
int main()
{ int x,nr1,nr2;
  ifstream f("bac.in");
  nr1=nr2=0;
  while(f>>x)
  { if(x%10%3==0)
    { if(nr2<x)
      { nr1=nr2;
        nr2=x;
      }
    else
      if(nr1<x)
        nr1=x;
    }
  }
  cout<<nr1<<" "<<nr2;
  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 valorilor cerute, **1p.** pentru afișarea lor, **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.**