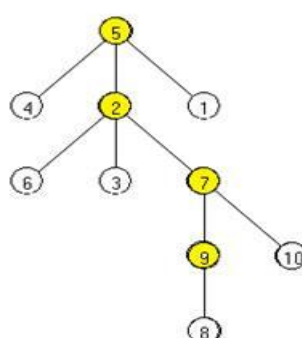


VARIANTA 2 - Rezolvare

Subiectul 1

1.	Se accesează cifrele unităților și zecilor numărului memorat în variabila a folosind operatorii div / și mod %. Pentru precizarea răspunsului d) se acordă 4p .
2.	<p>Apelul recursiv $f(x \text{ div } 10) f(x/10)$ pentru valorile x care au cifra unităților mai mică ca 5, sugerează că aceste cifre sunt ignorate.</p> <p>Cu restul cifrelor se contruiește un număr care are prima cifră 1 (prin apelul recursiv: $x \% 10 + f(x/10)$, respectiv $x \text{ mod } 10 + f(x \text{ div } 10) * 10$).</p> <p>Pentru ca rezultatul să fie 1 trebuie ca toate cifrele lui x să fie strict mai mici decât 5, iar cea mai mare valoare de 4 cifre cu toate cifrele pare este 4444.</p> <p>Pentru răspunsul c) se acordă 4p.</p>
3.	Se generează în maniera "întăinare cu revenire", cuvintele: abe, abec, abed, ace, aceb, aced , etc. Pentru răspunsul d) se acordă 4p .
4.	<p>Din vectorul de referințe ascendente deducem că rădăcina arborelui este nodul 5 ($tata[5]=0$).</p> <p>Observăm că $tata[8]=9$, $tata[9]=7$, $tata[7]=2$, $tata[2]=5$, deci ascendenții nodului 8 sunt nodurile: 5,2,7,9.</p> <p>Pentru precizarea răspunsului d) se acordă 4p.</p>
	
5.	<p>Graful, cu $n=2021$ noduri, fiind conex, nu poate avea noduri izolate, astfel gradul minim al unui nod poate fi 1. Gradul maxim poate fi $n-1$ (un nod poate fi adiacent cu toate celelalte $n-1$ noduri).</p> <p>Pentru precizarea răspunsului d) se acordă 4p.</p>

Subiectul 2

1.	<p>a) Se compară cifrele unităților numerelor a și b. Se alege cea mai mare dintre aceste cifre ale unităților, se adaugă la sfârșitul numărului nr și se elimină cifra aleasă din numărul din care provine. Se repetă acest procedeu până când unul dintre numerele a și b devine 0.</p> <p>Pentru precizarea valorii 4351 se acordă 6p.</p>
	<p>b) Pentru variabila a poate fi citit orice număr natural de patru cifre care are ultima cifră egală cu 0. Sunt 9 valori posibile pentru prima cifră, 10 valori posibile pentru a doua cifră și 10 valori posibile pentru a treia cifră. În total sunt $9 \times 10 \times 10 = 900$ numere care pot fi citite pentru variabila a.</p>

	Pentru precizarea valorii 900 se acordă 4p.
c)	<p>Înlocuirea structurii cât timp a>0 și b>0 execută... presupune utilizarea unei structuri repetitive repetă...cât timp sau repetă...până când:</p> <div style="display: flex; justify-content: space-around;"> <pre> dacă a>0 și b>0 atunci repetă dacă a %10 > b%10 atunci c←a%10 a←[a/10] altfel c←b%10 b←[b/10] nr←nr*10+c cât timp a>0 și b>0 </pre> <pre> dacă a>0 și b>0 atunci repetă dacă a %10 > b%10 atunci c←a%10 a←[a/10] altfel c←b%10 b←[b/10] nr←nr*10+c până când a=0 sau b=0 </pre> </div> <p>Fără testarea inițială a valorilor a și b (a>0 și b>0), dacă una dintre ele este nulă, algoritmul obținut prin înlocuirea mecanică a structurii cât timp cu structura repetă... va afișa răsturnatul celui alt număr, spre deosebire de algoritmul din enunț care afișează valoarea 0, deci cei doi algoritmi nu sunt echivalenți.</p> <p>Pentru alegerea unei structuri repetitive se acordă 2p., pentru testarea inițială a valorilor lui a și b 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 6p.</p>
d)	<p>Pentru instrucțiunile corecte de declarare a variabilelor, de citire a datelor, de afișare a rezultatului și de decizie se acordă câte 1p., pentru cele două instrucțiuni repetitive se acordă 3p., iar pentru cele 6 atribuiri se acordă 2p. Pentru structura corectă a programului se mai acordă 1p., în total 10p.</p>
2.	<p>Un răspuns corect: abs (C . O . y) <= C . R Pentru orice răspuns corect se acordă 6p.</p>
3.	<p>Răspuns corect: 1 3 5 7 9 11 La finalul executării secvenței date conținutul matricei A este:</p> <pre> 6 5 4 3 2 1 7 6 5 4 3 2 8 7 6 5 4 3 9 8 7 6 5 4 10 9 8 7 6 5 11 10 9 8 7 6 </pre> <p>Se acordă câte 1p. pentru fiecare element indicat corect.</p>

Subiectul 3

1.	Prelucrarea oferită de subprogram determină poziția primei apariții a valorii minime din tablou, poziția ultimei apariții a valorile maxime în șir, apoi interschimbă cele două valori situate pe pozițiile determinate. Tabloul rezultat este returnat prin intermediul parametrului v al subprogamului.
----	--

	Limbajul Pascal
	<pre> Type vector=array[1..100]of integer; procedure sub(n:byte;var v:vector); var pmin,pmax,i:byte; aux:integer; begin pmin:=1; pmax:=1; for i:=1 to n do if v[i]>=v[pmax] then pmax:=i else if v[i]<v[pmin] then pmin:=i; aux:=v[pmin]; v[pmin]:=v[pmax]; v[pmax]:=aux end;</pre>
	Limbajul C/C++
	<pre> void sub(int n, int v[]) { int i, pmin=1, pmax=1,aux; for(i=2;i<=n;i++) if (v[i]>=v[pmax]) pmax=i; else if (v[i]<v[pmin]) pmin=i; aux=v[pmin]; v[pmin]=v[pmax]; v[pmax]=aux; }</pre>
	<p>Pentru antet corect se acordă 2p. Pentru determinarea numărului minim din tabloul v se acordă 1p, iar pentru determinarea poziției primei apariții a minimului se acordă 2p. Pentru determinarea numărului maxim din tablou v se acordă 1p, iar pentru determinarea poziției ultimei apariții a maximului se acordă 2p. Pentru interchimbarea celor două numere determinate se acordă 1p. Pentru corectitudinea globală a programului 1p.</p>
2.	<p>O soluție se poate obține separând cuvintele din text și verificând apariția primului cuvânt ca sufix pentru restul cuvintelor din text; în caz afirmativ, aceste cuvinte vor fi eliminate. Se poate contrui un șir nou care să conțină cuvintele ce vor rămâne în text, prin concatenarea acestora și inserarea caracterului * după fiecare cuvânt.</p> <p>O altă soluție se poate obține, după separarea și eliminarea primului cuvânt din șir, prin determinarea tuturor aparițiilor primului cuvânt în șirul rămas. În situația în care caracterul * succede o astfel de apariție, atunci vom șterge din șir cuvântul corespunzător apariției.</p> <p>Se pot utiliza funcțiile predefinite pentru șiruri de caractere specifice limbajului.</p> <p>Pentru declararea corectă a variabilelor se acordă 1p. Pentru citirea șirului se acordă 1p. Pentru separarea primului cuvânt din text se acordă 1p. Pentru determinarea unui</p>

	cuvânt din text care admite ca sufix primul cuvânt și pentru eliminarea lui se acordă 2p . Pentru eliminarea tuturor cuvintelor din șir cu proprietatea cerută se acordă 2p . Pentru modificarea și afișarea șirului în formatul cerut se acordă 2p . Pentru corectitudinea globală a programului se acordă 1p - în total 10p .			
	Limbajul Pascal	Limbajul C/C++		
	<pre>var i,n,j,k :byte; s,p :string; begin readln(s); n:=pos('*',s); p:=copy(s,1,n); delete(s,1,n); k:=pos(p,s); while k>0 do begin j:=k+n-1; i:=k; while(s[i]<>'*') and (i>0) do dec(i); inc(i); delete(s,i,j-i); k:=pos(p,s); end; s:='*'+s; writeln(s) end.</pre>	<pre>int main() { char s[256], t[256], *p, *q, m; int i,j,n,ok; cin>>s; strcpy(t,""); p=strtok(s,"*"); n=strlen(p); q=strtok(NULL,"*"); while(q) { m=strlen(q); ok=0; for(i=m-n;i>=0 && ok==0;i--) if(strcmp(p,q+i)==0) ok=1; if(ok==0) strcat(t,q); strcat(t,"*"); q=strtok(NULL,"*"); } strcpy(s,t); cout<<s; return 0; }</pre>		
3.	b)	<p>Algoritmul de rezolvare propus determină suma cifrelor primului număr sx și apoi citește un număr din fișier, comparând suma cifrelor acestuia cu sx: în caz de egalitate se "prelungeste" secvența curentă, în caz contrar, se trece la o nouă secvență cu o nouă sumă sx curentă. La finalul oricărei secvențe se analizează lungimea ei, în vederea memorării lungimii maxime.</p> <p>Eficiența algoritmului, ca timp de executare, constă în faptul că determinăm valoarea cerută în timpul citirii datelor, deci 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>Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. - în total 2p.</p>		
	a)	<table><tr><th>Limbajul Pascal</th></tr><tr><td><pre>var f:text; nr,lgmax,lg,x,sx,snr:longint; begin assign(f,'bac.in');reset(f); read(f,x);</pre></td></tr></table>	Limbajul Pascal	<pre>var f:text; nr,lgmax,lg,x,sx,snr:longint; begin assign(f,'bac.in');reset(f); read(f,x);</pre>
Limbajul Pascal				
<pre>var f:text; nr,lgmax,lg,x,sx,snr:longint; begin assign(f,'bac.in');reset(f); read(f,x);</pre>				

```

lg:=1;
sx:=0;
while x>0 do begin
    sx:=sx+x mod 10;
    x:=x div 10
end;
while not.eof(f) do begin
    read(f,nr); snr:=0;
    while nr>0 do begin
        snr:=snr+nr mod 10;
        nr:=nr div 10
    end;
    if snr=sx then inc(lg)
    else begin
        if lgmax<lg then lgmax:=lg;
        lg:=1; sx:=snr
    end;
end;
if lgmax<lg then lgmax:=lg;
close(f);
write(lgmax)
end.

```

Limbajul C/C++

```

#include <iostream>
#include <fstream>
using namespace std;
ifstream f("bac.in");
int main()
{
    int lgmax=0, lg=1, nr, sx=0, x, snr;
    f>>x;
    while(x)
    {
        sx=sx+x%10; x=x/10; }
    while(f>>nr)
    {
        snr=0;
        while(nr)
        {
            snr+=nr%10; nr=nr/10; }
        if (snr==sx) lg++;
        else { if(lgmax<lg) lgmax=lg;
            lg=1; sx=snr;
        }
    }
    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. În total, **8p.**