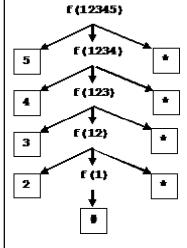
VARIANTA 3 - Rezolvare

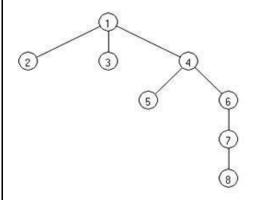
Rezolvare - Subjectul 1

Valoarea variabilei x trebuie să satisfacă condiția 100<=|x|<=999.
 Pentru precizarea răspunsului d) se acordă 4p.

2. În desenul alăturat este prezentat mecanismul prin care se execută apelul £ (12345) al subprogramului recursiv £. Se vor afișa, în ordine: 5432#****, răspuns pentru care se acordă 4p.



- 3. Răspuns corect **a.** pentru care se acordă **4p**.
- 4. Pentru răspunsul d) se acordă 4p.
- 5. Cele mai lungi lanţuri elementare din arbore cu o extremitate în nodul 8 sunt: [2,1,4,6,7,8] și [3,1,4,6,7,8].



Răspuns corect a. pentru care se acordă 4p.

Rezolvare - Subiectul 2

	_	, ·
1.	a)	Şirul de date de intrare (fără valoarea 0 utilizată pentru încheierea operației de citire) se împarte în secvențe formate din numere cu proprietatea că prima cifră a numărului curent este egală cu cifra unităților numărului precedent, iar cifra unităților numărului curent este egală cu prima cifră a numărului următor. Astfel șirul de date de intrare se împarte în trei secvențe de acest tip cu lungimile 3,1,4: 12 23 34 592 13 39 91 1 0 Pentru precizarea valorii 314 se acordă 6p.
	b)	Un răspuns posibil poate fi setul de date de intrare: 1 2 2 3 3 3 4 4 4 4 0 Pentru orice set de date de intrare corect se acordă 6p .
	c)	Înlocuirea structurii cât timp t>0 execută presupune utilizarea unei structuri repetitive repetă - cât timp sau repetă - până când: repetă t←[t/10] cât timp t>9 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. 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. Pentru instrucțiunile corecte de declarare a variabilelor, de citire a datelor, de afișare a rezultatului se acordă câte 1p, pentru instrucțiunea de decizie decizie se acordă 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 10p.
2.		Numerele cerute sunt de forma: 32xx3 cu xε{1,2,3,4}. Sunt generate 4x4=16 numere cu proprietatea din enunţ. Pentru răspunsul corect se acordă 6p.
3.		Şirului de caractere S i se atribuie valoarea 3a+b. Din acesta se șterge caracterul de pe prima poziție și noul șir va fi a+b=9. Instrucțiunea următoare șterge caracterul al doilea și al treilea. Rezultatul afișat este a=9. Pentru răspunsul corect se acordă 6p.

Rezolvare - Subjectul 3

1. Prelucrarea oferită de subprogram determină fiecare apariție a valorii x în tablou. Dublarea unui element cu valoarea x se realizează prin deplasarea cu o poziție la dreapta a tuturor elementelor situate la dreapta celui ce urmează a fi dublat, începând de la sfârșitul șirului. După realizarea deplasării, se mărește valoarea lui n cu 1.

Limbajul C/C++ Limbajul Pascal Type vector=array[1..100]of integer; void sub(int x,int &n,int v[]) procedure sub(x:integer;var n:byte;var int i=1, j;v:vector); while (i<=n) var i,j:byte; begin i := 1:if(v[i]==x) { while i<=n do begin for(j=n;j>=1;j--) if v[i]=x then begin v[j+1) = v[j];for j:=n downto 1 do n++; v[i+1] := v[i];i++; n := n+1;} i := i + 1i++; end; } i:=i+1end; end;

Pentru antet corect se acordă 3p. (Pentru respectarea structurii antetului (**procedure/void**) se acordă 1p., pentru declararea corectă a parametrilor de intrare **x**, pentru declararea corectă a parametrilor **n** si **v** încă 1p., pentru declararea variabilelor locale încă 1p).

Pentru determinarea fiecărei apariții ale numărului \mathbf{x} în tabloul 1p. Pentru dublarea primei apariții în tabloului și incrementarea valorii parametrului \mathbf{n} se acordă câte 1p. Pentru determinarea tuturor aparițiilor lui \mathbf{x} în tablou se acordă 2p. Pentru mărirea corespunzătoare a valorii parametrului \mathbf{n} se acordă 1p. Pentru corectitudinea globală a programului 1p. În total **10p.**

2. 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 m, în maniera descrisă, este:

Pentru declararea corectă a matricei se acordă 1p., pentru citirea lui n și m se mai acordă 1p., pentru completarea corectă a primei linii se acordă 1p., pentru completarea corectă a primei coloane se acordă 1p., pentru completarea corectă a celorlalte elemente 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.

b. Algoritmul de rezolvare propus presupune determinarea poziției pp a primului număr format doar din cifre pare, respectiv ui a ultimului număr format doar din cifre impare, din fişier pe măsura citirii datelor şi utilizează doar variabile simple fără folosirea unor structuri suplimentare de date.

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 utilizează doar variabile simple.

Lungimea subșirului căutat va fi: ui-pp+1 sau 0 dacă pp=0 sau ui=0.

Pentru o descriere coerentă a metodei se acordă 1p., iar pentru justificarea eficienței, încă 1p. - în total **2p.**

a.

Limbajul C/C++ Limbajul Pascal var f:text; #include <iostream> #include <fstream> nr, i,pp, ui,p, imp:longint; using namespace std; begin assign(f,'bac.in'); reset(f); ifstream f("bac.in"); while not(eof(f)) do begin int main() read(f,nr); int pp=0, ui=0, i=0, nr, p, imp; inc(i);p := 0;while(f>>nr) imp:=0;p=imp=0;repeat i++; if nr mod 2=0 then inc(p) do else inc(imp); { if(nr%2==0) p++; nr:=nr div 10 else imp++; until nr=0; nr/=10;if (imp=0) and (pp=0)}while(nr>0); then pp=i if(imp==0 && pp=0)pp=i; else if (p==0) ui:=i; else if (p==0) ui=i; end; if(pp*ui) cout<<"Nu exista";</pre> if pp*up=0 then writeln ('0') else cout<<ui-pp+1; else writeln(ui-pp+1); close(f); f.close(); end.

Se acordă 1p. pentru operaţiile cu fişiere (declarare, nume corect și deschidere pentru citire), încă 1p. pentru citirea tuturor numerelor din fişier, 1p. pentru un algoritm principial corect, încă 1p. pentru determinarea valorii cerute, 1p. pentru afișarea ei, 1p. pentru corectitudinea formală (declararea variabilelor, structura programului, sintaxa instrucţiunilor, 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.