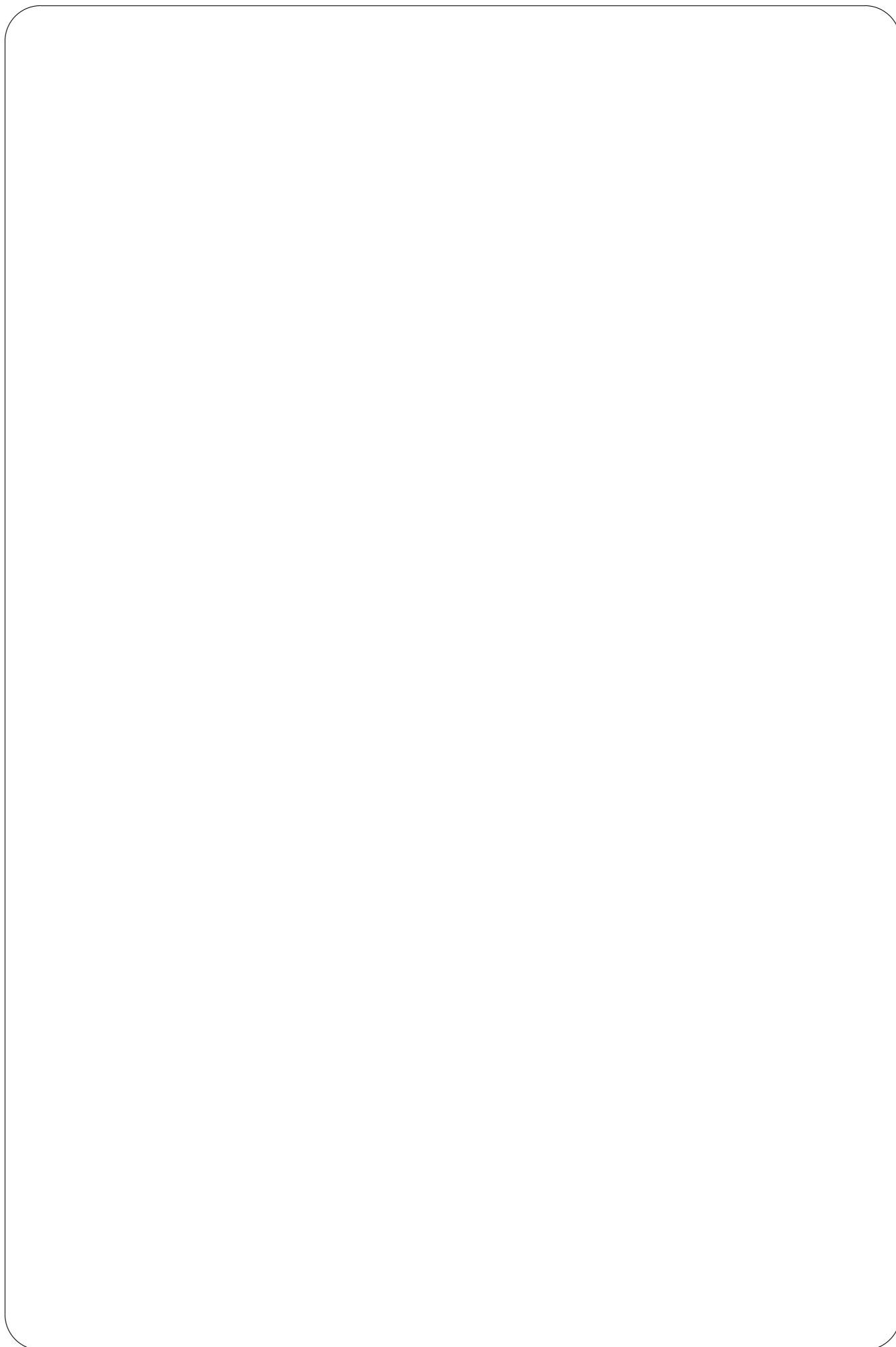


Memento Pascal

Jurcone Ramiro



Prefață

Prezenta lucrare are ca scop **Introducerea în programare** cu ajutorul limbajului Pascal.

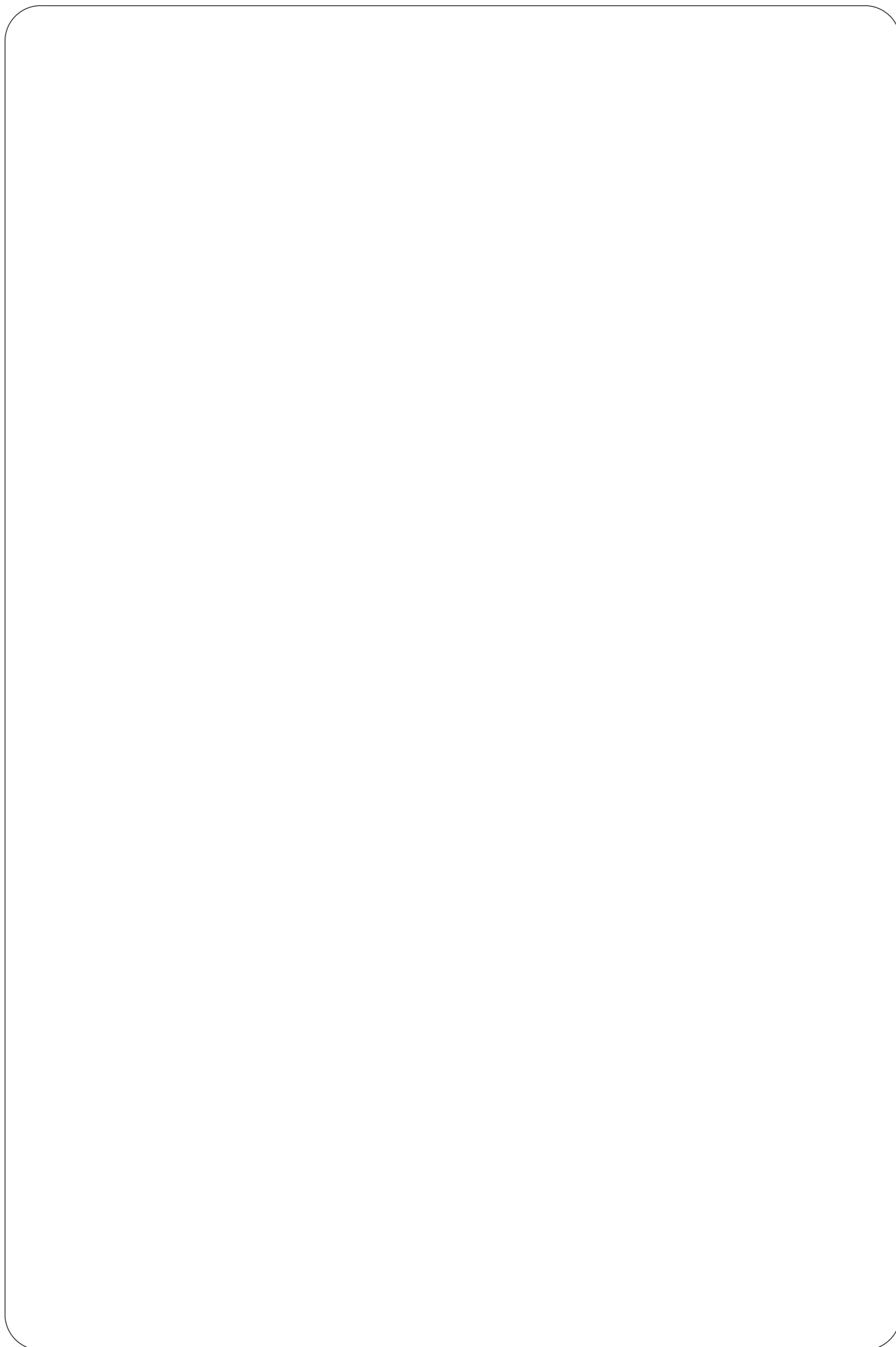
Partea I Conține modele de programe elementare în Pascal. Scopul este însușirea structurii unui program și a algoritmilor de bază. De asemenea cu ocazia rezolvării problemelor se ia contactul cu instrucțiunile de bază ale limbajului. Pentru abordarea acestei parti este utilă cunoașterea pseudocodului sau a modului de lucru cu scheme logice. Fiecare program este realizat pe sistemul un program pe o pagină.

Partea II După însușirea părții I, se prezintă în continuare principalele elemente al limbajului Pascal, sub formă de **Lecții**. Lecțiile sunt organizate sub formă de fișe, pentru un subiect fiind alocată în principiu o fișă, adică două pagini față/verso. Pentru fiecare lecție, se prezintă la început noțiuni de teorie și apoi un exemplu de program.

Partea a II-a conține de asemenea o parte de **Lecții suplimentare**, în care sunt tratate capitole mai dificile, care în opinia autorului nu prezintă un interes deosebit în etapa de inițiere în programare. Din acest motiv, prezentarea lor s-a facut sub formă de *Lecturi*, adică o prezentare simplă a chestiunilor esențiale, având ca scop mai degrabă informarea cititorului privind existența și rolul acestor subiecte.

Partea a II-a conține în continuare o porțiune de **Anexe**, unde sunt prezentate anumite aspecte considerate ca utile în practică.

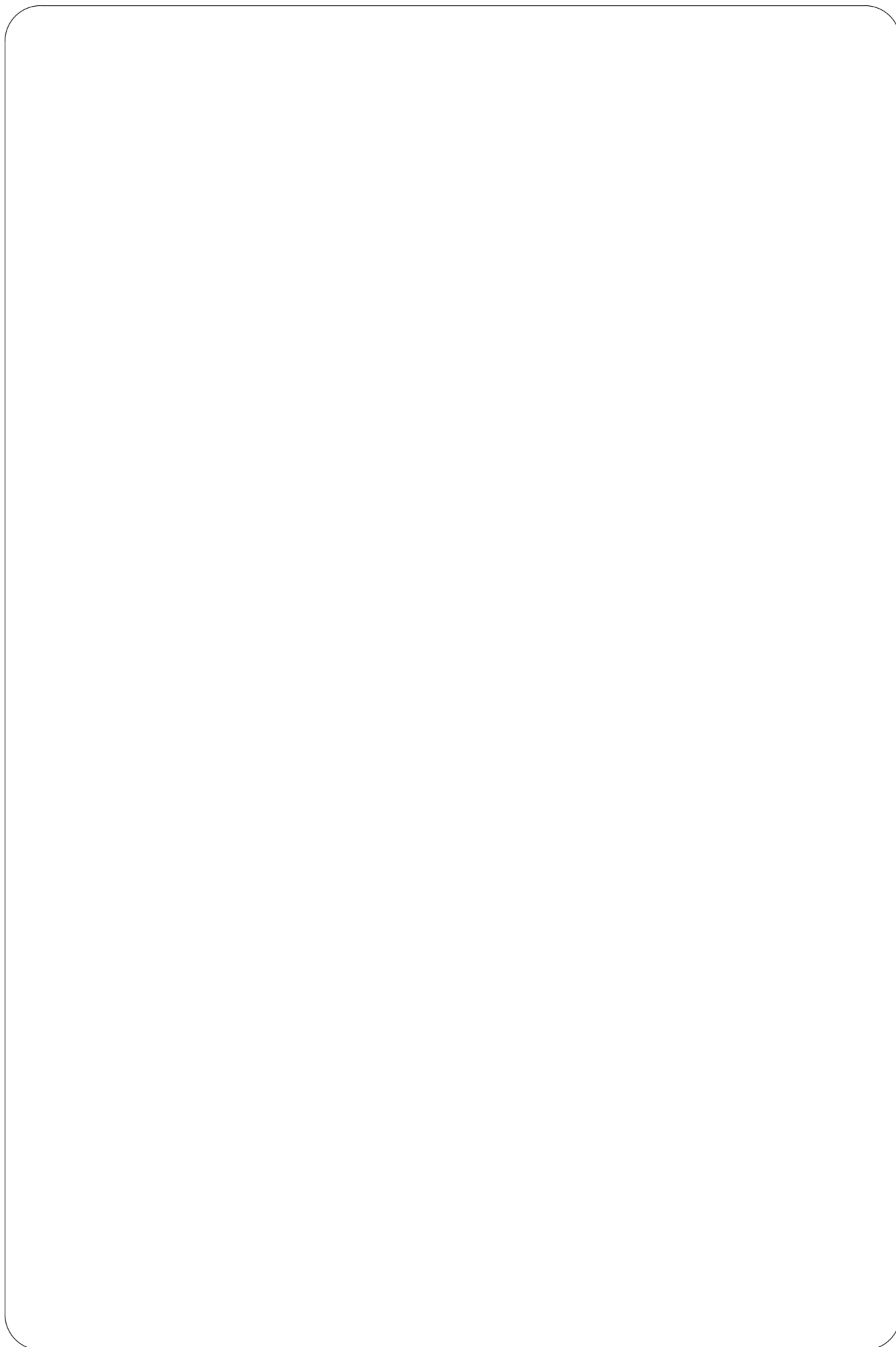
Prezenta lucrare nu are pretenția de a acoperi absolut toate aspectele limbajul Pascal, scopul fiind realizarea unei introduceri în limbaj. După stăpânirea noțiunilor prezentate, se apreciază că cititorul va dispune de noțiunile necesare pentru a aborda singur aspectele netratate în prezenta lucrare.



Cuprins

Partea I = Exerciții în Pascal

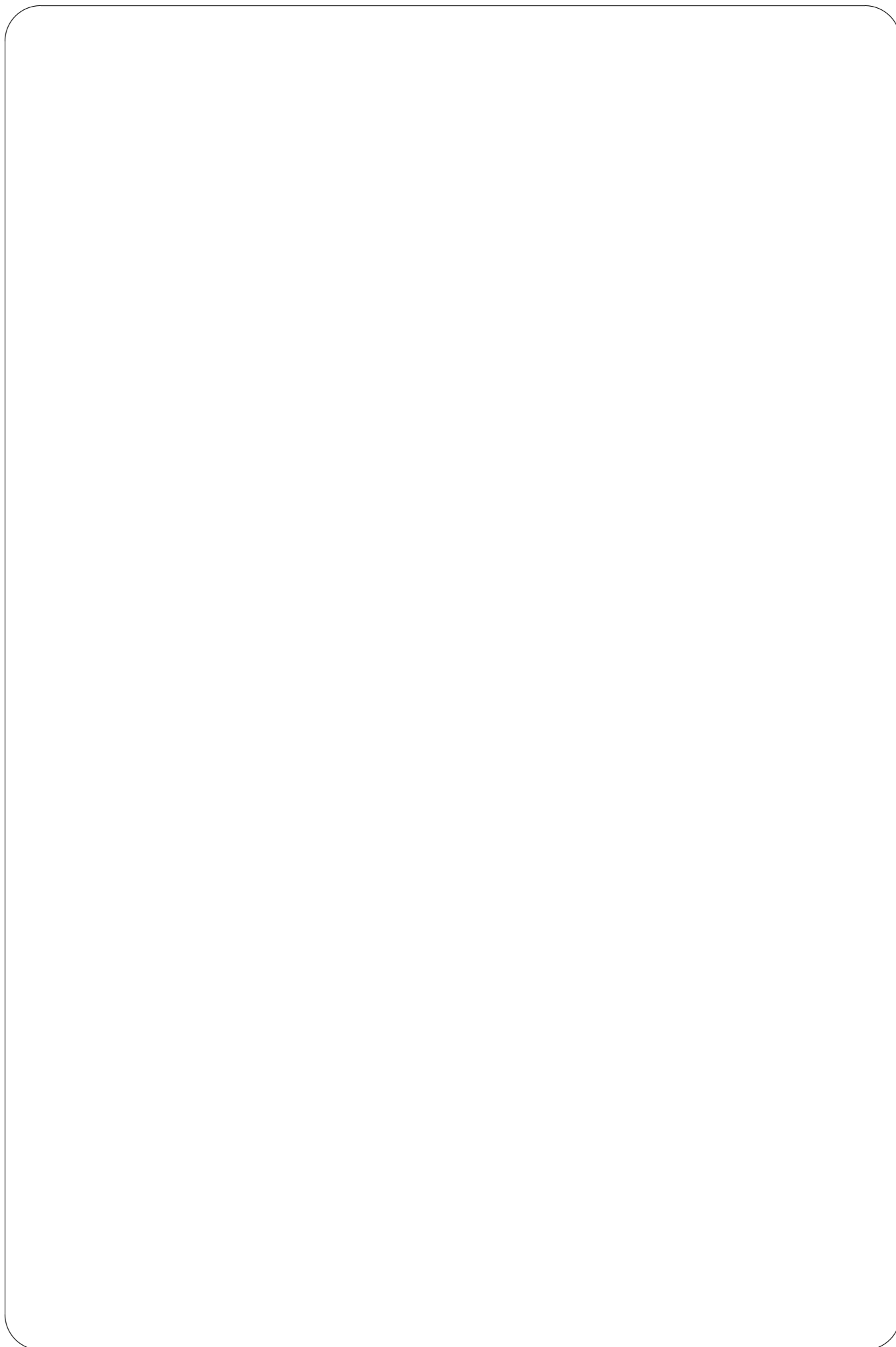
Nr.	Subiect	Pagina
1	Scheletul unui program Pascal	11
2	Calculul unei sume	12
3	Calculul unui produs	13
4	Comanda If	14
5	Tipul Real	15
6	Afișari cu tipul Real	16
7	Tipul boolean, If, număr prim	17
8	Suma cifrelor unui număr	18
9	Oglinditul unui număr	19
10	Divizorii unui număr	20
11	Maximul dintre n numere	21
12	Exercițiu	22
13	Exercițiu	23
14	Exercițiu	24
15	Operații matematice, Sqrt	25
16	Vectori=suma elementelor	26
17	Vectori=exersare	27
18	Vectori=exersare	28
19	Vectori=exersare	29
20	Vectori=maxim si minim	30
21	Vectori=generarea unui alt vector	31
22	Vectori=exersare	32
23	Vectori=exersare	33
24	Grafică	34
25	Vectori=căutare	35
26	Vectori=exercițiu căutare	36
27	Vectori=căutare rapidă	37
28	Vectori=sortare	38
29	Vectori=stergere element	39
30	Vectori=exercițiu stergere	40
31	Vectori=inserare element	41
32	Vectori=exercițiu inserare vector	42
33	Matrice=suma a două matrici	43
34	Matrice=produsul dintre matrice si scalar	43
35	Matrice=diagonala principală, secundară	45
36	Matrice=exersare	46
37	Matrice=maximul din matrice, pe linii, pe coloane	47
38	Anexă	49



Cuprins

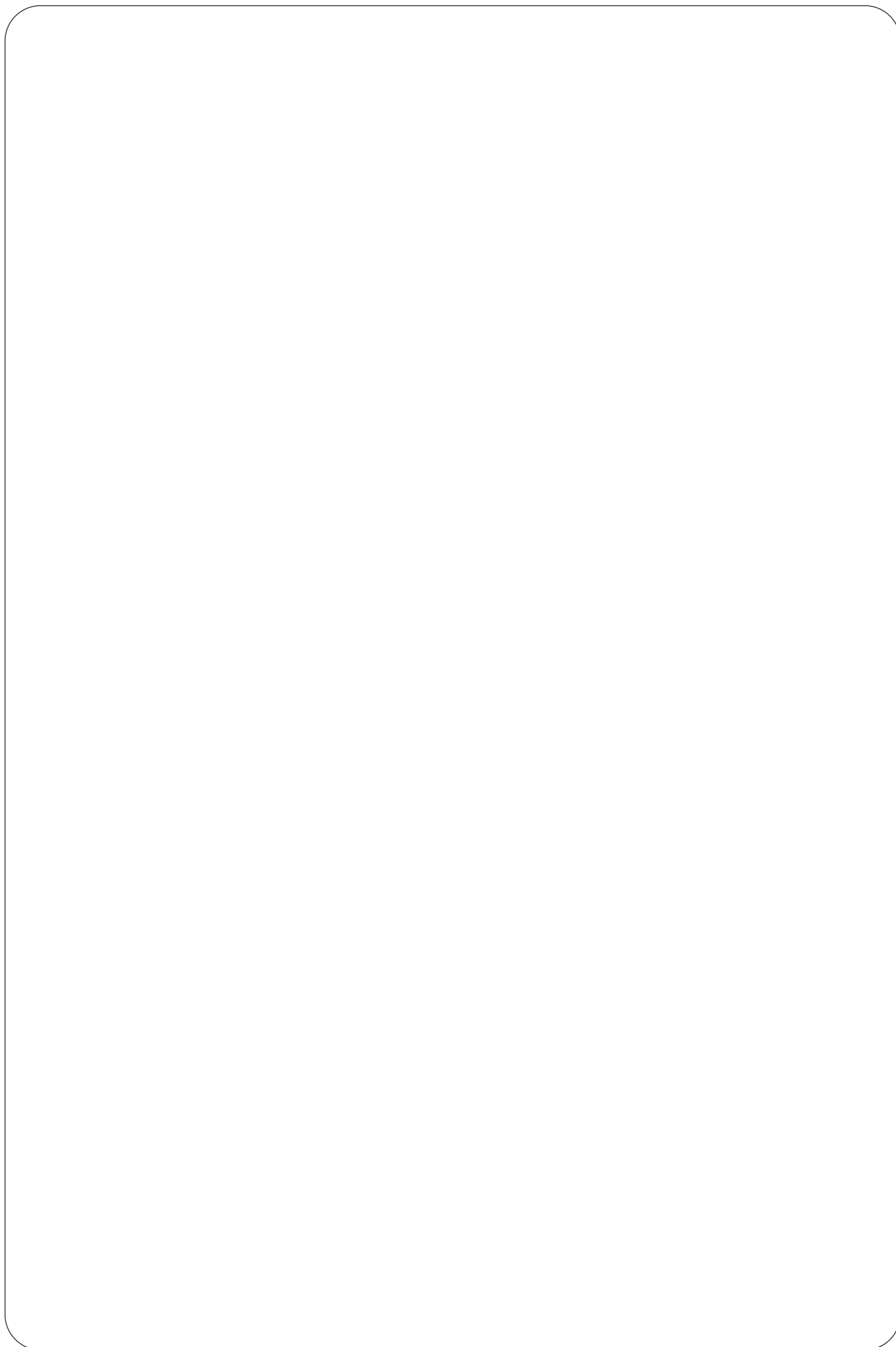
Partea a II-a = Fișe teorie și probleme

Nr	Conținut	Pagina
Lectii de Pascal		
1	Baze de numerație	53
2	Realizarea de meniuri	55
3	Tipul char. Coduri Ascii	57
4	Tipul string	59
5	Fișiere text	61
6	Tipul record	63
7	Subprograme: Function	65
8	Subprograme: Procedure	67
9	Recursivitate	69
10	Programarea orientată obiect	71
11	Programarea orientată obiect: Delphi	73
Lecturi suplimentare		
12	Metoda back-tracking	75
13	Pointeri	77
14	Liste inlantuite	79
15	Grafuri neorientate	81
16	Grafuri orientate	83
Anexe		
17	Anexa 1: Compilatoare de Pascal	85
18	Anexa 2: Pascal versus alte limbaje de programare	87
19	Anexa 3: Baze de date: Limbajul Sql	89
20	Anexa 4: Internet: Limbajul Html	91



PARTEA I

Programe elementare în Pascal



PASCAL lectia 1

Noțiuni noi

- Scheletul unui program PASCAL(program, var, begin, .. readln, end., comentariu)
- writeln, readln
- ciclul for
- calculul unei sume

Programul in PASCAL

```
{ Fisier orli1.pas.  
  Enunt: Calculati suma primelor n numere naturale}
```

```
program orlando1;  
var i,n:integer;  
    s:longint;  
begin  
    writeln('introduceti_n');  
    readln(n);  
    s:=0;  
    for i:=1 to n do  
        s:=s+i;  
    writeln(s);  
    readln;  
end.
```

PASCAL lectia 2

Noțiuni noi

- calculul unei sume mai complicate

Programul in PASCAL

```
{ Fisier orli2.pas.  
  Enunt: Calculati suma patratelor primelor n numere naturale}  
  
program orlando2;  
var i,n:integer;  
    s:longint;  
begin  
    writeln('introduceti n');  
    readln(n);  
    s:=0;  
    for i:=1 to n do  
        s:=s+i*i;  
    writeln(s);  
    readln;  
end.
```

PASCAL lectia 3

Noțiuni noi

- calculul unei produs
- longint (= întreg lung pentru numere mari)

Programul in PASCAL

```
{ Fisier orli3.pas.  
  Enunt: Calculati produsul primelor n numere naturale}  
  
program orlando3;  
var i,n:integer;  
    p:longint;  
begin  
    writeln('introduceti n');  
    readln(n);  
    p:=1;  
    for i:=1 to n do  
        p:=p*i;  
    write(p);  
    readln;  
end.
```

PASCAL lectia 4

Noțiuni noi

- IF = comanda pentru "dacă"
- MOD = modulo(restul împărțirii)

Programul in PASCAL

```
{ Fisier orli4.pas.  
  Enunt: calculati suma numerelor pare mai mici decat n}  
  
program orlando4;  
var n,i:integer;  
    s:longint;  
begin  
    writeln('introduceti n');  
    readln(n);  
    s:=0;  
    for i:=1 to n do  
        if (i mod 2=0) then  
            s:=s+i;  
    writeln('suma=',s);  
    readln;  
end.
```

PASCAL lectia 5

Noțiuni noi

- tipul REAL
- afișarea tipului REAL, de exemplu p:10:2

Programul in PASCAL

```
{ Fisier orli5.pas.  
  Enunt: calculati produsul numerelor impare mai mici  
    sau egale cu n}
```

```
program orlando5;  
var i,n:integer;  
    p:real;  
begin  
    writeln('introduceti n');  
    readln(n);  
    p:=1;  
    for i:=1 to n do  
        if (i mod 2<>0) then  
            p:=p*i;  
    writeln('produs=',p:10:2);  
    readln;  
end.
```

PASCAL lectia 6

Noțiuni noi

- Calcule si afisari cu tipul REAL

Programul in PASCAL

```
{ Fisier orli6.pas.  
  Enunt: cititi r(raza cercului) si apoi  
         calculati si afisati perimetrul si aria cercului}  
  
program orlando6;  
var r,p,a:real;  
begin  
    writeln('introduceti raza');  
    readln(r);  
    p:=2*3.14*r;  
    a:=3.14*r*r;  
    writeln('perimetrul=',p:10:2);  
    writeln('aria=',a:10:2);  
    readln;  
end.
```

PASCAL lectia 7

Noțiuni noi

- tipul BOOLEAN(= tip logic de tip true sau false)
- IF .. THEN ... ELSE (= DACA ... ATUNCI ... ALTFEL ...)
- algoritmul de verificare a unui numar daca este prim

Programul in PASCAL

```
{ Fisier orli7.pas.  
Enunt: cititi n si verificati daca e prim}
```

```
program orlando7;  
var i,n:integer;  
prim:boolean;  
begin  
    writeln('introduceti_n');  
    readln(n);  
    prim:=true;  
    for i:=2 to n-1 do  
        if (n mod i=0) then  
            prim:=false;  
    if (prim=true) then  
        writeln('n_este_prim')  
    else  
        writeln('n_nu_este_prim');  
    readln;  
end.
```

PASCAL lectia 8

Noțiuni noi

- Aflarea ultimei cifre : $UC = N \text{ MOD } 10$
- Eliminarea ultimei cifre : $N = N \text{ DIV } 10$
- algoritmul de aflare a sumei cifrelor unui numar

Programul in PASCAL

```
{ Fisier orlo8.pas=suma cifrelor unui numar }
program orlando8;
  var n, Sc, Uc: integer;
begin
  writeln( 'introduceti n' );
  readln(n);
  Sc:=0;
  while n<>0 do
    begin
      Uc:=n mod 10;
      Sc:=Sc+Uc;
      n:=n div 10;
    end;
  writeln( 'suma cifrelor=' ,Sc);
  readln;
end.
```

PASCAL lectia 9

Noțiuni noi

- Palindrom
- algoritmul de aflare a oglinditului unui numar

Programul in PASCAL

*{ Fisier orlo9.pas = cititi un numar n ,calculati
si afisati oglinditul
si verificati daca este palindrom }*

```
program orlando9;  
  var Uc,n,aux,ogl:integer;  
begin  
  writeln('introduceti n');  
  readln(n);  
  aux:=n;  
  ogl:=0;  
  while (aux<>0) do  
  begin  
    Uc:=aux mod 10;  
    ogl:=10*ogl+Uc;  
    aux:=aux div 10;  
  end;  
  writeln('oglinditul=',ogl);  
  if (n=ogl) then  
    writeln('este palindrom')  
  else  
    writeln('nu este palindrom');  
  readln;  
end.
```

PASCAL lectia 10

Noțiuni noi

- algoritmul de aflare a sumei divizorilor unui numar

Programul in PASCAL

{ Fisier orlo10.pas = calculati suma divizorilor lui n }

```
program orlando10;  
  var s,n,i:integer;  
begin  
  writeln('introduceti n');  
  readln(n);  
  s:=0;  
  for i:=1 to n do  
    if (n mod i=0) then  
      s:=s+i;  
  writeln('suma divizorilor=',s);  
  readln;  
end.
```

PASCAL lectia 11

Noțiuni noi

- MAXINT , respectiv -MAXINT
- algoritmul de aflare a maximul (cel mai mare) dintre n numere

Programul in PASCAL

{ Fisier orlo11.pas = cititi n ,apoi n numere si aflati si afisati maximul celor n numere }

```
program orlando11;
uses crt;
var n,i,maxim,a:integer;
begin
  clrscr;
  writeln('introduceti n');
  readln(n);
  maxim:=-MAXINT;
  for i:=1 to n do
    begin
      writeln('introduceti numarul');
      readln(a);
      if (a>maxim) then
        maxim:=a;
    end;
  writeln('maxim=',maxim);
  readln;
end.
```

PASCAL lectia 12

Noțiuni noi

- Exercițiu: fara noutati

Programul in PASCAL

```
{ Fisier Orlo12.pas
  =Cititi doua numere si verificati daca suma lor
  este mai mare decat 100, sau mai mica}

program orlando12;
uses crt;
var a,b,s:integer;
begin
    clrscr;
    write('a='); readln(a);
    write('b='); readln(b);
    s:=a+b;
    if (s>100) then
        writeln('mai_mare_ca_100')
    else
        writeln('mai_mic_ca_100');
    readln;
end.
```

PASCAL lectia 13

Noțiuni noi

- Exercițiu: fara noutati

Programul in PASCAL

```
{ Fisier Orlo13.pas
=Cititi trei numere si calculati suma ultimelor cifre
ale numerelor. Verificati daca acest numar e mai mare
sau mai mic decat 10}

program orlando13;
uses crt;
var a,b,c,x:integer;
begin
    clrscr;
    write('a='); readln(a);
    write('b='); readln(b);
    write('c='); readln(c);
    x:=(a mod 10)+(b mod 10)+(c mod 10);
    if (x>10) then
        writeln('x_mai_mare_ca_10')
    else
        writeln('x_mai_mic_ca_10');
    readln;
end.
```

PASCAL lectia 14

Noțiuni noi

- Exercițiu: fara noutati

Programul in PASCAL

*{ Fisier Orlo14.pas= cititi 3 numere a,b,c;
verificati daca a,b,c
pot fi laturile unui triunghi.
Daca da , verificati daca triunghiul
este dreptunghic.
Daca da , calculati suprafata triunghiului }*

```
program orlando14;
uses crt;
var a,b,c:integer;
    p,s:real;
begin
    clrscr;
    write('a='); readln(a);
    write('b='); readln(b);
    write('c='); readln(c);
    if (a<=b+c) and (b<=a+c) and (c<=a+b) then
        begin
            writeln('este _triunghi ');
            if (a*a=b*b+c*c) or (b*b=c*c+a*a) or (c*c=a*a+b*b) then
                writeln('este _triunghi _dreptunghic ')
            else
                writeln('nu _este _triunghi _dreptunghic ');
            p:=(a+b+c)/2;
            s:=sqrt(p*(p-a)*(p-b)*(p-c));
            writeln('suprafata= ',s:10:2);
        end
    else
        writeln('nu _este _triunghi ');
    readln;
end.
```

PASCAL lectia 15

Noțiuni noi

- sqrt(radical), operatii matematice

Programul in PASCAL

*{ Fisier Orlo15.pas= cititi 3 numere a,c,b reale;
rezolvati ecuatia de gradul 2 ax^2+bx+c }*

```
program orlando15;
uses crt;
var a,b,c,delta,x1,x2:real;
begin
  clrscr;
  write('a='); readln(a);
  write('b='); readln(b);
  write('c='); readln(c);
  delta:=b*b-4*c;
  if (delta>0)then
    begin
      x1:=(-b+sqrt(delta))/(2*a);
      x2:=(-b-sqrt(delta))/(2*a);
      writeln('x1=', x1:10:2);
      writeln('x2=', x2:10:2);
    end
  else
    if (delta=0) then
      begin
        x1:=(-b)/(2*a);
        x2:=(-b)/(2*a);
        writeln('x1=x2=', x1:10:2);
      end
    else
      Writeln('imaginar');
  readln;
end.
```

Noțiuni noi

- Vectori, citire vector
- Calculul sumei elementelor unui vector

Programul in PASCAL

```
{ Fisier: Orlo16.pas=cititi un vector de
n elemente intregi si calculati suma
elementelor vectorului}

program orlando16;
uses crt;
type vector=array[1..10] of integer;
var n,i,s:integer;
    v:vector;
begin
    clrscr;
    writeln('introduceti n');
    readln(n);
    for i:= 1 to n do
        begin
            write('introduceti v[',i,'] ');
            readln(v[ i ]);
        end;
    s:=0;
    for i:=1 to n do
        s:=s+v[ i ] ;
    writeln('suma=',s);
    readln;
end.
```

PASCAL lectia 17

Noțiuni noi

- Vectori, exersare

Programul in PASCAL

*{ Orlo17.pas=cititi un vector de n elemente intregi
si calculati suma elementelor pare din vector,
dupa care afisati suma }*

```
program orlando17;
uses crt;
type vector=array[1..20] of integer;
var n,i,s:integer;
    v:vector;
begin
    clrscr;
    write('introduceti n=');
    readln(n);
    for i:=1 to n do
        begin
            write('introduceti v[',i,'] ');
            readln(v[i]);
        end;
    s:=0;
    for i:=1 to n do
        if (v[i] mod 2=0) then
            s:=s+v[i];
    writeln('suma nr pare=',s);
    readln;
end.
```

PASCAL lectia 18

Noțiuni noi

- Vectori, exersare

Programul in PASCAL

*{Orlo18.pas=cititi un vector de n elemente intregi.
Calculati si apoi afisati produsul numerelor impare}*

```
program orlando18;
uses crt;
type vector=array[1..20] of integer;
var n,i,p:integer;
    v:vector;
begin
    clrscr;
    write('introduceti n ');
    readln(n);
    for i:= 1 to n do
        begin
            write('introduceti v[',i,'] ');
            readln(v[ i ]);
        end;
    p:=1;
    for i:=1 to n do
        if (v[ i ] mod 2<>0) then
            p:=p*v[ i ];
    writeln('produsul numerelor impare=',p);
    readln;
end.
```

PASCAL lectia 19

Noțiuni noi

- Vectori, exersare

Programul in PASCAL

*{orlo19.pas=cititi un vector de n elemente intregi.
Calculati cate elemente ale vectorului au ultima cifra
egala cu 3}*

```
program orlando19;
uses crt;
type vector=array[1..20] of integer;
var i,n,k:integer;
    v:vector;
begin
    write('introduceti n= ');
    readln(n);
    for i:=1 to n do
        begin
            write('introduceti v[ ',i, ' ] ');
            readln(v[ i ]);
        end;
    k:=0;
    for i:=1 to n do
        if (v[ i ] mod 10=3) then
            k:=k+1;
    writeln(k);
    readln;
end.
```

PASCAL lectia 20

Noțiuni noi

- Vectori. Minimul si maximul dintr-un vector.

Programul in PASCAL

*{ Orlo20.pas=cititi un vector de n elemente intregi.
Afisati minimul element al vectorului }*

```
program orlando20;
uses crt;
type vector=array[1..20] of integer;
var n,i,minim:integer;
    v:vector;
begin
    clrscr;
    write('introduceti_n_');
    readln(n);
    for i:=1 to n do
        begin
            writeln('introduceti_v[',i,']_');
            readln(v[ i ]);
        end;
    minim:=v[1];
    for i:=2 to n do
        if (v[ i ]<minim) then
            minim:=v[ i ];
    writeln('minim=_', minim);
    readln;
end.
```

Noțiuni noi

- Vectori. Generarea unui vector dintr-un vector existent.

Programul in PASCAL

*{ Orlo21.pas=cititi un vector v de n elemente intregi.
Generati un vector u care sa contina elementele
din v inmultite cu 10 }*

```
program orlando21;
uses crt;
type vector=array[1..20] of integer;
var n,i:integer;
    v,u:vector;
begin
    clrscr;
    write('introduceti n=');
    readln(n);
    for i:=1 to n do
        begin
            writeln('introduceti v[',i,']=');
            readln(v[i]);
        end;
    for i:=1 to n do
        u[i]:=v[i]*10;
    writeln('vectorul u=');
    for i:=1 to n do
        writeln(u[i]);
    readln;
end.
```

PASCAL lectia 22

Noțiuni noi

- Vectori, exersare

Programul in PASCAL

*{ Orlo22.pas=Fie un vector v de n elemente intregi.
Generati un vector u care sa
contina oglinditele elementelor vectorului v }*

```
program orlando22;
uses crt;
type vector=array[1..20] of integer;
var n,ogl,aux,i,uc:integer;
    v,u:vector;
begin
    clrscr;
    write('introduceti n=');
    readln(n);
    for i:=1 to n do
    begin
        writeln('introduceti v[',i,']=');
        readln(v[i]);
    end;
    for i:=1 to n do
    begin
        aux:=v[i];
        ogl:=0;
        while (aux<>0) do
            begin
                uc:=aux mod 10;
                ogl:=ogl*10+uc;
                aux:=aux div 10;
            end;
        u[i]:=ogl;
    end;
    writeln('vectorul u=');
    for i:=1 to n do
        writeln(u[i]);
    readln;
end.
```

Noțiuni noi

- Vectori, exersare

Programul in PASCAL

*{Orlo23.pas=Fie un vector v de n elemente intregi.
Calculati cate numere prime sunt in vector si
afisati acest numar}*

```
program orlando23;
uses crt;
type vector=array[1..20] of integer;
var n,i,j,k:integer;
    prim:boolean;
    u,v:vector;
begin
    clrscr;
    k:=0;
    write('introduceti n=');
    readln(n);
    for i:=1 to n do
        begin
            writeln('introduceti v[',i,']=');
            readln(v[i]);
        end;
    for i:=1 to n do
        begin
            prim:=true;
            for j:=2 to v[i]-1 do
                if (v[i] mod j=0) then
                    prim:=false;
            if (prim=true) then
                k:=k+1;
        end;
        writeln('in vector sunt k=',k,' numere prime');
        readln;
    end.
```

PASCAL lectia 24

Noțiuni noi

- Grafica in pascal, introducere. Biblioteca crt

Programul in PASCAL

{ Orlo24.pas = Introducere in grafica in Pascal }

```
program orlando24;  
uses crt;  
begin  
    clrscr;  
    gotoxy(50,3);  
    writeln('salut');  
    textbackground(red);  
    textcolor(blue);  
    gotoxy(1,10);  
    writeln('noroc');  
    sound(700);  
    delay(100);  
    sound(800);  
    delay(100);  
    nosound;  
    readln;  
end.
```

Noțiuni noi

- Cautare in vector
- Este necesara folosirea unei variabile booleene(ex: gasit) deoarece rezultatul cautarii este de tipul adevarat sau fals.
- Aceasta metoda NU este rapida, deoarece din cauza ciclului **for**, chiar dupa gasirea elementului cautat, se mai pierde timp cu parcurgerea intregului vector

Programul in PASCAL

*{ORLO25.PAS Fie un vector v de n elemente intregi.
Cautati numarul 7 in vector afisand mesajul corespunzator}*

```
program orlo25;
uses crt;
type vector=array[1..20] of integer;
var n,i:integer;
    v:vector;
    gasit:boolean;
begin
    clrscr;
    write('introduceti n=');
    readln(n);
    for i:=1 to n do
        begin
            writeln('introduceti v[',i,']');
            readln(v[i]);
        end;
    gasit:=false;
    for i:=1 to n do
        if (v[i]=7) then
            gasit:=true;
    if (gasit=true) then
        writeln('am gasit')
    else
        writeln('nu am gasit');
    readln;
end.
```

Noțiuni noi

- Cautare in vector, exercitiu

Programul in PASCAL

*{ORLO26.PAS Fie un vector v de n elemente intregi
si un numar k.
Cautati numarul k in vector,
afisand pozitiile pe care apare}*

```
program orlando26;
uses crt;
type vector=array[1..20] of integer;
var n,i,k:integer;
    v:vector;
    gasit:boolean;
begin
    clrscr;
    write('introduceti n='); readln(n);
    write('k='); readln(k);
    for i:=1 to n do
        begin
            writeln('introduceti v[',i,' ]');
            readln(v[ i ]);
        end;
    gasit:=false;
    for i:=1 to n do
        if (v[ i ]=k) then
            begin
                writeln('pozitia=',i);
                gasit:=true;
            end;
    if (gasit=false) then
        writeln('numarul',k,'nu se afla in vector');
    readln;
end.
```

Noțiuni noi

- Cautare RAPIDA in vector
- Cautarea rapida necesita utilizare ciclu **while**, **nu for**, pentru a iesi din ciclu cand este gasit elementul cautat

Programul in PASCAL

*{ORLO27.PAS cautare rapida:Fie un vector v de n
elemente intregi si un numar k.
Cautati numarul k in vector afisand
mesajul corespunzator}*

```
program orlando27;
uses crt;
type vector=array[1..20] of integer;
var i,k,n:integer;
    v:vector;
    gasit:boolean;
begin
  clrscr;
  writeln('introduceti n='); readln(n);
  writeln('k='); readln(k);
  for i:=1 to n do
    begin
      writeln('introduceti v[',i,' ]');
      readln(v[ i ]);
    end;
  gasit:=false;
  i:=1;
  while (gasit=false) and (i<=n) do
    begin
      if (v[ i ]=k) then
        gasit:=true;
        i:=i+1;
      end;
    if (gasit=true) then
      writeln('am gasit ')
    else
      writeln('nu am gasit ');
  readln;
end.
```

Noțiuni noi

- Sortare vector
- Metoda paharelor
- Ideea de doua cicluri. Primul ciclu, pentru $i=1$, **rezolva** elementul 1, pentru $i=2$, **rezolva** elementul 2, s.a.m.d.
- Indicele i avanseaza pana la $n-1$ (nu pana la n), deoarece indicele urmator j va incepe cu $i+1$
- Al doilea ciclu, incepe **de la dreapta lui i** , deci $j:=i+1$ si continua pana la ultimul element al vectorului, adica n .

Programul in PASCAL

*{ORLO28.PAS Cititi un vector v de n elemente intregi.
Sortati vectorul v, apoi afisati vectorul v.}*

```
program orlo28;
uses crt;
type vector=array[1..20] of integer;
var n,i,j,aux:integer;
    v:vector;
begin
  clrscr;
  write('introduceti n='); readln(n);
  for i:=1 to n do
    begin
      write('introduceti v[',i,']=');
      readln(v[i]);
    end;
  for i:=1 to n-1 do
    for j:=i+1 to n do
      if (v[i]>v[j]) then
        begin
          aux:=v[i];
          v[i]:=v[j];
          v[j]:=aux;
        end;
  write('vectorul sortat=');
  for i:= 1 to n do
    write(v[i], ' ');
  readln; end.
```

Noțiuni noi

- Stergere din vector
- Se utilizeaza **while**, nu **for** deoarece NU trebuie $i:=i+1$ cum face **for**.
- In esenta, stergerea din vector implica gasirea elementului de sters, deplasarea spre stanga a elementelor urmatoare si modificarea dimensiunilor vectorului, adica $n:=n-1$

Programul in PASCAL

{ Orlo29.pas Stergeti toate numerele pare dintr-un vector v cu n elemente intregi }

```

program orlo29;
uses crt;
type vector=array[1..20] of integer;
    var n,i,p,j:integer;
        v:vector;
begin
    clrscr;
    write('n='); readln(n);
    for i:=1 to n do
        begin
            write('introduceti v[',i,',']_'); readln(v[ i ]);
        end;
    i:=1;
    while (i<=n) do
        if (v[ i ] mod 2=0) then
            begin
                p:=i;          { = deplasare la stanga }
                for j:=p+1 to n do
                    v[ j-1 ]:=v[ j ];
                n:=n-1;
            end
        { !!! AICI e capcana: NU trebuie i:=i+1, cum ar face for,
          deoarece pe pozitia i este elementul deplasat din dreapta,
          care trebuie si el analizat }
        else
            i:=i+1;
    write('vectorul=');
    for i:=1 to n do
        write(v[ i ],'_');
    readln;
end.

```

PASCAL lectia 30

Noțiuni noi

- Stergere din vector. Exercițiu. Nimic nou

Programul in PASCAL

{orlo30.pas : Stergeti toate palindroamele dintr-un vector.}

```
program orlo30;
uses crt;
var v:array [1..20] of integer;
    n,i,aux,ogl,p,j,uc:integer;
begin
    clrscr;
    write('n='); readln(n);
    for i:=1 to n do
        begin
            write('v[',i,']='); readln(v[i]);
        end;
    i:=1;
    while (i<=n) do
        begin
            aux:=v[i];
            ogl:=0;
            while (aux<>0) do
                begin
                    uc:=aux mod 10;
                    ogl:=ogl*10+uc;
                    aux:=aux div 10;
                end;
            if (ogl=v[i]) then
                begin
                    p:=i;
                    for j:=p+1 to n do
                        v[j-1]:=v[j];
                    n:=n-1;
                end
            else
                i:=i+1;
            end;
        end;
    for i:=1 to n do writeln(v[i]); readln;
end.
```

Noțiuni noi

- Inserare in vector
- Se utilizeaza **while**, **nu for** deoarece NU trebuie $i:=i+1$ cum face for.
- In esenta, inserarea in vector implica gasirea elementului dupa care se insereaza, deplasarea spre dreapta a elementelor urmatoare, depunderea elementului de inserat si modificarea dimensiunilor vectorului, adica $n:=n+1$
- Deplasarea spre dreapta se face cu **downto**, adica deplasarea incepe de la termenul n spre termenul $p+1$, altfel s-ar *distruge* valorile prin suprascriere.

Programul in PASCAL

{ Orlo31.pas : Fie un vector v de n elemente intregi.
Inserati numarul 77 dupa fiecare numar par din vector }

```

program orlo31;
uses crt;
var v:array [1..20] of integer;
    n,i,j,p:integer;
begin
    clrscr;
    write('n=');readln(n);
    for i:=1 to n do
        begin
            write('v[',i,']=');readln(v[i]);
        end;
    i:=1;
    while (i<=n) do
        if (v[i] mod 2=0) then
            begin
                p:=i;
                for j:=n downto p+1 do
                    v[j+1]:=v[j];
                n:=n+1;
                v[p+1]:=77;
                i:=i+2; { Aici e capcana. FOR ar face i=i+1 }
            end
        else
            i:=i+1;
    for i:=1 to n do
        writeln(v[i]);
    readln;
end.

```

PASCAL lectia 32

Noțiuni noi

- Inserare in vector. Exercițiu. Nimic nou

Programul in PASCAL

*{ orlo32.pas : Fie un vector v de n elemente intregi.
Inserati dupa fiecare nr. par oglinditul
acestuia. }*

```
program orlo32;
uses crt;
var v:array [1..20] of integer;
    n,i,ogl,j,p,uc,aux:integer;
begin
    clrscr;
    write('n=');readln(n);
    for i:=1 to n do begin
        write('v[',i,']=');readln(v[i]);
    end;
    i:=1;
    while (i<=n) do
        if (v[i] mod 2=0) then
            begin
                aux:=v[i];ogl:=0;
                while (aux > 0) do
                    begin
                        uc:=aux mod 10;ogl:=ogl*10+uc;
                        aux:=aux div 10;
                    end;
                p:=i;
                for j:=n downto p+1 do
                    v[j+1]:=v[j];
                n:=n+1; v[p+1]:=ogl; i:=i+2;
                { AICI este motivul pt care folosim WHILE, NU FOR }
            end
        else
            i:=i+1;
    for i:=1 to n do
        writeln(v[i]); readln;
end.
```

Noțiuni noi

- Notiunea de matrice. Citire matrice, calcule, afisare matrice

Programul in PASCAL

*{matrice1.pas: Cititi doua matrici A si B.
Calculati matricea suma C=A+B si afisati C}*

```
program m1;
uses crt;
type matrice=array[1..100,1..100] of integer;
var A,B,C:matrice;
    m,n,i,j:integer;
begin
    clrscr;
    write('m=');readln(m);
    write('n=');readln(n);
    for i:=1 to m do
        for j:=1 to n do
            begin
                write('A[',i,',',j,']=');
                readln(A[i,j]);
            end;
    for i:=1 to m do
        for j:=1 to n do
            begin
                write('B[',i,',',j,']=');
                readln(B[i,j]);
            end;
    for i:=1 to m do
        for j:=1 to n do
            C[i,j]:=A[i,j]+B[i,j];
    writeln('└─Matricea└suma└este:');
    for i:=1 to m do
        begin
            for j:=1 to n do
                write(C[i,j], '└');
            writeln;
        end;
    readln;
end.
```

Noțiuni noi

- Calcule cu matrici

Programul in PASCAL

*{matrice2.pas= cititi o matrice A si un intreg k.
a) Efectuati produsul dintr matrice si scalarul k,
B=k*A si afisati matricea B
b) Calculati suma elementelor matricii}*

```
program m2;
uses crt;
type matrice=array[1..100,1..100] of integer;
var   A,B:matrice;
      m,i,j,k,s1,s2:integer;
begin
  clrscr;
  write('m=');readln(m);
  write('k=');readln(k);
  for i:=1 to m do
    for j:=1 to m do
      begin
        write('A[',i,',',j,']=');readln(A[i,j]);
      end;
  for i:=1 to m do
    for j:=1 to m do
      B[i,j]:=k*A[i,j];
  s1:=0;
  for i:=1 to m do
    for j:=1 to m do
      s1:=s1+A[i,j];
  for i:=1 to m do
    begin
      for j:=1 to m do
        write(B[i,j],'_');
      writeln;
    end;
    writeln('suma=',s1);
  readln;
end.
```

Noțiuni noi

- Matrici: diagonala principala, diagonala secundara, E, V, S ,N

Programul in PASCAL

```
{matrice3.pas= diagonala principala , diagonala
secundara in matrice
a) cititi o matrice A
b) Calculati si afisati suma elementelor aflate
pe diagonala principala
c) Similar pentru elementele aflate sub diag.
principala , pe diagonala secundara , sub diagonala
principala
d) Similar pentru zona Est(=deasupra diag.princ.
si totodata sub diag.secundara)}
```

```
program m3;
uses crt;
type matrice=array[1..100,1..100] of integer;
var A:matrice;
    m,i,j,s1,s2,s3,s4,s5:integer;
begin
    clrscr; write('m=');readln(m);
    for i:=1 to m do
        for j:=1 to m do
            begin
                write('A[',i,',',j,']=');readln(A[i,j]);
            end;
    for i:=1 to m do
        for j:=1 to m do
            if (i=j) then s1:=s1+A[i,j];
    for i:=1 to m do
        for j:=1 to m do
            if (i>j) then s2:=s2+A[i,j];
    for i:=1 to m do
        for j:=1 to m do
            if (i+j=m+1) then s3:=s3+A[i,j];
    for i:=1 to m do
        for j:=1 to m do
            if (i+j>m+1) then s4:=s4+A[i,j];
    for i:=1 to m do
        for j:=1 to m do
            if (i<j) and (i+j>m+1) then s5:=s5+A[i,j];
    writeln('suma_pe_diagonala_principala=',s1);
    writeln('suma_sub_D.P=',s2);writeln('suma_pe_D.S=',s3);
    writeln('suma_sub_DS=',s4);writeln('suma_zona_E=',s5);
    readln;end.
```

Noțiuni noi

- Matrice: exercitiu = nimic nou

Programul in PASCAL

*{matrice4.pas=Calculati si afisati
numarul de palindroame dintr-o matrice}*

```
program m4;
uses crt;
type matrice=array[1..100,1..100] of integer;
var A:matrice;
    m,n,i,j,pal,aux,ogl,uc:integer;
begin
  clrscr;
  write('m=');readln(m);
  write('n=');readln(n);
  for i:=1 to m do
    for j:=1 to n do
      begin
        write('A[',i,',',j,']=');
        readln(A[i,j]);
      end;
  pal:=0;
  for i:=1 to m do
    for j:=1 to n do
      begin
        aux:=A[i,j];
        ogl:=0;
        while (aux<>0) do
          begin
            uc:=aux mod 10;
            ogl:=ogl*10+uc;
            aux:=aux div 10;
          end;
        if (ogl=A[i,j]) then pal:=pal+1;end;
      write('sunt ',pal,' palindroame');
      readln;
    end.
```

Noțiuni noi

- Matrice: maximul din matrice, maxim pe linii, maxim pe coloane

Programul in PASCAL

```
{matrice5.pas= calculati si afisati urmatoarele:  
a) Maximul dintr-o matrice  
b) Maximul de pe fiecare linie a matricii  
c) Maximul de pe fiecare coloana a matricii}  
type matrice=array [1..100,1..100] of integer;  
var i,j,m,n,max1,max2,max3:integer;  
a:matrice;  
begin  
  write( 'n=' ); readln(n); write( 'm=' ); readln(m);  
  for i:=1 to m do  
    for j:=1 to n do  
      begin  
        write( 'a[ ',i, ', ',j, ']= ' );   readln(a[i,j]);  
      end;  
    for i:=1 to m do  
      for j:=1 to n do  
        if (a[i,j]>max1) then  
          max1:=a[i,j];  
        writeln( 'maximul_din_matrice_este_',max1);  
        for i:=1 to m do  
          begin  
            max2:=a[i,1];  
            for j:=2 to n do  
              if (a[i,j]>max2) then  
                max2:=a[i,j];  
              writeln( 'maximul_pe_linia_',i, '_este_',max2);  
            end;  
          for j:=1 to n do  
            begin  
              max3:=a[1,j];  
              for i:=2 to m do  
                if (a[i,j]>max3) then  
                  max3:=a[i,j];  
                writeln( 'maximul_pe_coloana_',j, '_este_',max3);  
            end;  
          for i:=1 to m do  
            begin      for j:=1 to n do  
              write(a[i,j], ' ');   writeln; end;  
          readln; end.
```

Anexă

Capitole neconținute în lucrare:

Scopul lucrării a fost **Introducerea în programare**. Conține principalele noțiuni de bază ale limbajului Pascal. Programele prezentate anterior acoperă în principiu materia de Pascal care se studiază timp de un an școlar la un liceu de profil matematică-informatică, în regim de două ore pe săptămână. Deoarece în clasa a IX-a se studiază pseudocod și în clasa a X-a se începe Pascal, practic lucrarea de față reprezintă sinteza materiei studiate în clasa a X-a .

Nu au fost incluse capitole care se studiaza în al doilea an de studiu a Pascal - adică în clasa a XI- din doua motive:

1) prezentarea acestora depășește timpul alocat unui curs introductiv, scopul lucrării fiind introducerea în programare, nu perfecționarea în programare.

2) noțiunile netrate în lucrare sunt în principiu elemente compuse, eventual avansate, bazate în mare măsură pe noțiunile de bază prezentate. Fără stăpanirea noțiunilor de bază, este imposibil de stăpanit noțiunile compuse.

Pentru realizare unei instruiți aprofundate este necesară și studierea acestor capitole. De exemplu pentru un absolvent de liceu de profil matematică-informatică ce intenționează să susțină proba de bacalaureat din informatică, sau pentru un absolvent de liceu care intenționează să urmeze o facultate de profil informatic. În schimb studiul capitolelor menționate anterior nu este neapărat necesară pentru o persoană care intenționează să utilizeze informatica dar nu intenționează să devină informatician. Ca analogie, în urma cu 30 de ani la Politehnică se studia de către toți viitorii ingineri, în anul I timp de un an, limbajul Fortran, în scopul introducerii în informatică. Studenții de la facultăți de profil informatic studiau și în anii următori alte limbaje de programare, pe când studenții de specialități diferite de informatică, se limitau la studiul unui singur limbaj introductiv - în acele vremuri limbajul Fortran- timp de un an de zile. Cam acesta este și scopul capitolelor prezentate în lucrare și anume ca limbaj introductiv pentru neinformaticieni și ca bază de pornire pentru informaticieni.

Capitole netratate în lucrare sunt următoarele:

(Cele 12 capitole marcate cu • sunt considerate mai utile practic)

-
1. Sortare=bubble sort, numărare, inserare
 - 2. Fundamente=baze de numeratie
 - 3. Meniuri=exercițiu repeat și case
 - 4. Tipul char. Coduri Ascii
 - 5. Tipul string
 - 6. Tipul record
 7. Tipul enumerare
 8. Tipul multime
 - 9. Fișiere text
 10. Fișiere cu tip

 - 11. Subprograme=function
 - 12. Subprograme=procedure
 13. Subprograme recursive
 14. Backtracking recursiv/iterativ=permutări
 15. Backtracking recursiv/iterativ=aranjamente
 16. Backtracking recursiv/iterativ=combinări
 17. Backtracking recursiv/iterativ=produs cartezian
 18. Backtracking recursiv/iterativ=submulțimi
 19. Backtracking recursiv/iterativ=exerciții
 20. Divide et impera
 21. Căutarea binara
 22. Sortare=quicksort

 23. Grafuri=neorientate
 24. Grafuri=orientate

 25. Pointeri
 26. Liste=simplu inlanțuite
 27. Liste=dublu inlanțuite
 28. Liste=circulare
 29. Arbori=binari
 30. Arbori=parcurgerea arborilor
 31. Arbori=arbori de căutare

 - 32. Programarea obiect=teoretic
 - 33. Programarea obiect=practic:Limbajul Delphi
 34. Programarea obiect=proiect in limbajul Delphi

 - 35. Baze de date=limbajul SQL teoretic
 - 36. Baze de date=limbajul SQL practic: Oracle sau Access
 37. Baze de date=proiect: SQL accesat cu Pascal/Delphi
-

PARTEA II

FIȘE teorie și programe în PASCAL

