

Bazele programării I

Tablouri bidimensionale

Tablouri bidimensionale. Necesitate

Temperaturile zilnice ale anului pot fi păstrate într-o structură de tip tablou unidimensional cu 365 elemente.

Type

`Zile_an = Array [1..365] of Integer`

Var

`Temp: Zile_an`



`[1] [2] [3] [4]`

`[...]`

`[364] [365]`

Tablouri bidimensionale. Necesitate

Temperaturile zilnice ale anului pot fi păstrate într-un tablou bidimensional în felul următor:

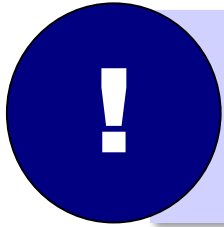
	1	2	3	4	5	6	7	8	9	10	...	26	27	28	29	30	31
1																	
2																	
3																	
4																	
5							-1										
6																	
7																	
8																	
9																	
10																	
11																	
12																	

coloana 7

rândul 5

Temperatura la data 7 mai

Tablouri bidimensionale. Noțiune



Tabloul bidimensional este un vector elementele căruia sunt de tip vector.

Type

`<Nume_tip> = Array <tip_indice1, tip_indice2>
of <tip_element>`

tip_indice1 – poate fi de orice tip ordinal și indică modalitatea de specificare a rândurilor.

tip_indice2 – poate fi de orice tip ordinal și indică modalitatea de specificare a coloanelor.

Tip element – indică tipul de date a elementelor și poate fi de orice tip de date.

Tablouri bidimensionale. Exemplu

Type

Zi = 1.. 31

Luna = 1..12

Temp = Array[Luna] of Array[Zi] Of **Integer**

Var

An_2003: **Temp**

An_2005: **Temp**

An_2012: **Temp**

An_2003[3,4]

An_2005[6,15]

An_2012[8,3]

Accesarea unui element al tabloului bidimensional

<Nume_variabila>[<Indice_rand>,<Indice coloana>]

Tablouri bidimensionale. Exemplu

Type

Luna = (Martie, Apr, Mai)

Tablou = Array[1..5] of Array[Luna] of
Integer

Var

X: Tablou

	Martie	Apr	Mai
1			
2	-5		
3			
4			
5			12

X[2, Martie] := -5

X[5, Mai] := 12

X[3, Dec]

Tablou[2, Martie]

Tablouri bidimensionale. Exemplu

Type

```
Zi_de_scoala = (Luni, Marti, Miercuri,  
                Joi, Vineri)
```

```
Orar: Array[1..4] of Array [Zi_de_scoala]  
      of String
```

Var

```
Orarul_meu: Orar
```

	Luni	Marti	Miercuri	Joi	Vineri
1					
2			Matem		
3					
4					

`Orarul_meu[2,Miercuri]`

Parcurgerea elementelor matricei

Const

N = ...

M = ...

Tabloul bidimensional **A** va
conține **N*M** elemente

Type

Tb = Array[1..N] of Array[1..M] of
Integer

Var

A: **Tb**

A_{11}	A_{12}	...	A_{1m}
A_{21}	A_{22}	...	A_{2m}
A_{31}	A_{32}	...	A_{3m}
...
A_{n1}	A_{n2}		A_{nm}

Parcurgerea elementelor matricei

Elementele unui tablou bidimensional pot fi accesate secvențial prin două metode:

- pe rânduri;
- pe coloane.

A_{11}	A_{12}	...	A_{1m}
A_{21}	A_{22}	...	A_{2m}
A_{31}	A_{32}	...	A_{3m}
...
A_{n1}	A_{n2}		A_{nm}

Prelucrare secvențială pe rânduri

```
For I:= 1 To N Step 1
```

ciclu pe rânduri

```
For J:=1 To M step 1
```

ciclu pe coloane

```
    Prelucrarea elementului A[I,J]
```

```
End
```

```
End
```

I/J	1	2	...	M
1	A_{11}	A_{12}	...	A_{1m}
2	A_{21}	A_{22}	...	A_{2m}
3	A_{31}	A_{32}	...	A_{3m}
...
N	A_{n1}	A_{n2}		A_{nm}

Prelucrare secvențială pe coloane

```
For J:= 1 To M Step 1
```

```
  For I:=1 To N step 1
```

```
    Prelucrarea elementului A[I,J]
```

```
  End
```

```
End
```

I/J	1	2	...	M
1	A_{11}	A_{12}	...	A_{1m}
2	A_{21}	A_{22}	...	A_{2m}
3	A_{31}	A_{32}	...	A_{3m}
...
N	A_{n1}	A_{n2}		A_{nm}

Prelucrare elementelor rândului K

For J:= 1 **To** M **Step** 1

<prelucrarea elementului $A[K, J]$ >

End

I/J	1	2	...	M
1	A_{11}	A_{12}	...	A_{1m}
2	A_{21}	A_{22}	...	A_{2m}
...			...	
k	A_{k1}	A_{k2}	...	A_{km}
...
N	A_{n1}	A_{n2}		A_{nm}

Prelucrare elementelor coloanei K

```
For I:= 1 To N Step 1
```

```
    <prelucrarea elementului A[I,K]>
```

```
End
```

I/J	1	2	...	k	...	M
1	A_{11}	A_{12}	\dots	A_{1k}	\dots	A_{1m}
2	A_{21}	A_{22}	\dots	A_{2k}	\dots	A_{2m}
3	A_{31}	A_{32}	\dots	A_{3k}	\dots	A_{3m}
...	\dots	\dots	\dots	\dots	\dots	\dots
N	A_{n1}	A_{n2}	\dots	A_{nk}	\dots	A_{nm}

Citirea unei matrice

```
For I:= 1 To N Step 1
```



```
  For J:=1 To M step 1
```

```
    Prelucrare A[I,J]
```

```
  End
```

```
End
```

```
ReadInt (A[I,J])
```

 
A[1,1]= 25

A[1,2]= 14

A[1,3]= 14

...

A[3,4]= 54

```
For I:= 1 To N Step 1
```

```
  For J:=1 To M step 1
```

```
    ReadInt (A[I,J])
```

```
  End
```

```
End
```

Afișarea tabloului bidimensional

- Afișarea elementului **specificat**;
- Afișarea **tuturor** elementelor tabloului;
- Afișarea elementelor tabloului care posedă o careva **proprietate**;
- Afișarea elementelor unui **rând**;
- Afișarea elementelor unui **rând** care posedă o careva **proprietate**;
- Afișarea elementelor unei **coloane**;
- Afișarea elementelor unei **coloane** care posedă o careva **proprietate**.

Afișarea matricei pe rânduri

```
For I:= 1 To N Step 1
```

```
  For J:=1 To M step 1  
    Prelucrare A[I,J]
```

```
  End
```

```
End
```

```
WriteInt (A[I,J])
```

afișarea
rândului

în același rând

```
For I:= 1 To N Step 1
```

```
  For J:=1 To M step 1
```

```
    WriteInt (A[I,J])
```

```
  End
```

trecem la rând
nou

```
  Writeln
```

```
End
```


Afișarea elementelor matricei care posedă o careva proprietate

```
For I:= 1 To N Step 1
```

```
  For J:=1 To M step 1  
    Prelucrare A[I,J]  
  End
```

```
End
```

```
  If A[I,J]=<proprietate>  
  Then WriteInt(A[I,J])  
End
```

```
For I:= 1 To N Step 1
```

```
  For J:=1 To M step 1
```

```
    If A[I,J]=<proprietate>  
    Then WriteInt(A[I,J])  
    End
```

```
  End
```

```
End
```

Afișarea elementelor rândului K

For J:= 1 **To** M **Step** 1

<prelucrarea
elementului A[K,J]>

End

WriteInt(A[K,J])

For J:= 1 **To** M **Step** 1

WriteInt(A[K,J])

End

Afișarea elementelor coloanei K

```
For I:= 1 To N Step 1
```

```
    <prelucrarea  
    elementului A[I,K]>
```

```
End
```

```
WriteInt(A[I,K])
```

```
For I:= 1 To N Step 1
```

```
    WriteInt(A[I,K])
```

```
End
```

Numărarea elementelor matricei care posedă o careva proprietate

```
Rez:=0
```

```
For I:= 1 To N Step 1
```

```
For J:=1 To M step 1
```

```
    Prelucrare A[I,J]
```

```
End
```

```
End
```

```
If A[I,J]=<proprietate>
```

```
Then  Rez:= Rez+1
```

```
End
```

inițializarea
variabilei Rez

```
Rez:=0
```

```
For I:= 1 To N Step 1
```

```
For J:=1 To M step 1
```

```
If A[I,J]=<proprietate>
```

```
Then  Rez:= Rez+1
```

```
End
```

```
End
```

```
End
```

numărarea
elementelor

Însumarea componentelor unui tablou bidimensional

Const

N = 3

M = 4

Var

A: array[1..N, 1..M] of integer

i, j, S: integer

begin

{ introducerea tabloului și afișarea lui }

S := 0

for i:=1 to N step 1

for j:=1 to M step 1

S := S + A[i,j];

end

end

WriteString('Suma elementelor matricei = ')

WriteInt(S)

end.

Determinarea elementului maxim al tabloului

Max:=A[1,1]

P_X:=1

P_Y:=1

For I:= 1 To N Step 1

For J:=1 To M step 1

Prelucrare A[I,J]

End

If A[I,J]>Max Then

Max:=A[I,J]

End

P_X:=I

P_Y:=J

End

Max:=A[1,1]

P_X:=1

P_Y:=1

For I:= 1 To N Step 1

For J:=1 To M step 1

If A[I,J]> Max Then

Max:= A[I,J]

End

P_X:=I

P_Y:=J

End

End

inițializarea
variabilei Max

actualizarea
variabilei Max

Existența în tablou a elementelor care au o anumită proprietate

```
Exist := False
```

```
I := 1
```

```
While I <= N And Not Exist Do
```

```
  J := 1
```

```
  While J <= M And Not Exist Do
```

```
    Prelucrare A[I,J]
```

```
    J := J+1
```

```
  End
```

```
  I := I+1
```

```
End
```

```
I := 1
```

```
While I <= N Do
```

```
  ...
```

```
  I := I+1
```

```
End
```

```
If A[I,J] = <prop. > Then
```

```
  Exist := True
```

```
End
```

Existența în tablou a elementelor care au o anumită proprietate

```
Exist := False
```

```
I := 1
```

```
While (I ≤ N) And Not Exist Do
```

```
    J := 1
```

```
        While (J ≤ M) And Not Exist Do
```

```
            If A[I, J] = <proprietate> Then
```

```
                Exist := True
```

```
            End
```

```
        J := J + 1
```

```
    End
```

```
    I := I + 1
```

```
End
```


Existența în rândul K a elementelor care au o anumită proprietate

J:=1

While J<=3 and A[3,J] <> 0 **Do**

J:=J+1

End

If J<=3 **Then**

WriteString('Exista')

Else

WriteString('Nu exista')

End

K = 3

J≤3	1≤3	2≤3	3≤3
A[3,J]<>0	T	T	F

	1	2	3
1	-1	3	-7
2	4	8	0
3	6	1	0
4	7	2	-2

Existența în coloană K a elementelor care au o anumită proprietate

I:=1

While I≤4 and A[I,2] <> 0 **Do**

I:=I+1

End

If I≤4 **Then**

WriteString('Exista')

Else

WriteString('Nu exista')

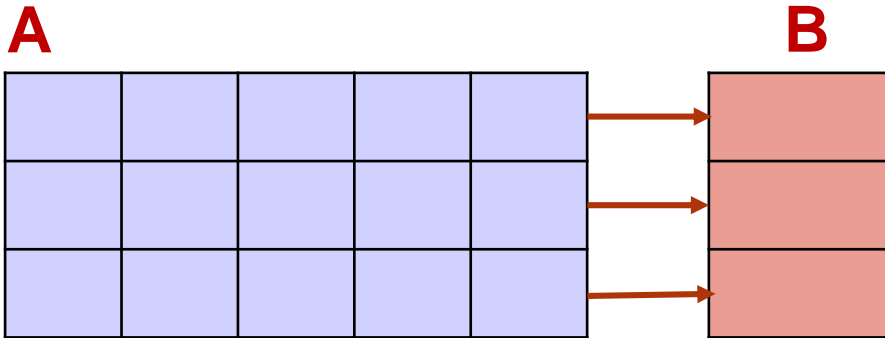
End

K = 2

I≤4	1≤4	2≤4	3≤4	4≤4	5≤4
A[I,2]<>0	T	T	T	T	F

	1	2	3
1	-1	3	-7
2	4	8	0
3	6	1	0
4	7	2	-2

Prelucrarea elementelor fiecărui rând



Type

Tb = Array[1..N] Of

Array[1..M] Of

Tu = Array[1..N] Of

Var

A: Tb

B: Tu

For I:=1 To N Step 1

S:=0

For J:=1 To M Step 1

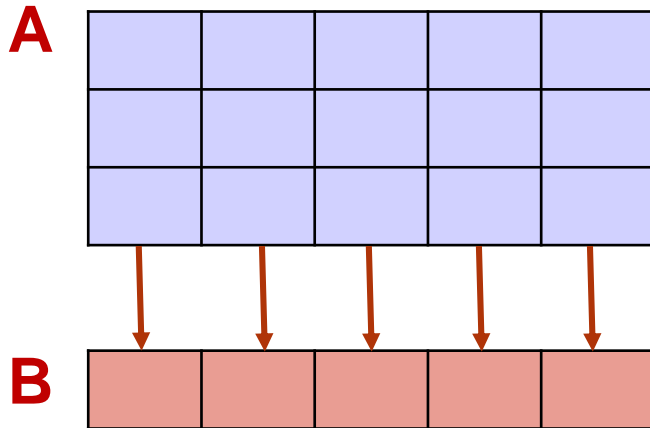
S:=S+A[I,J]

End

B[I] := S

End

Prelucrarea elementelor fiecărei coloane



```
For J:=1 To M Step 1
```

```
    S:=0
```

```
    For I:=1 To N Step 1
```

```
        S:=S+A[I,J]
```

```
    End
```

```
    B[J] := S
```

```
End
```

Type

Tb = Array[1..N] Of

Array[1..M] Of Integer

Tu = Array[1..M] Of Integer

Var

A: Tb

B: Tu

Eliminarea rândului/coloanei din tablou

```
For I:=K to N step 1
  For J:=1 to M step 1
    A[I,J]:=A[I+1,J]
  End
End
N:=N-1
```

```
For J:=K to M step 1
  For I:=1 to N step 1
    A[I,J]:=A[I,J+1]
  End
End
M:=M-1
```

Înserarea unui rând în tablou

Type

Tb = Array[1..**N+1**] Of Array[1..M] Of **Integer**

Tu = Array[1..**M**] Of **Integer**

Var

A: **Tb**

C: **Tu**

```
For I:=N to K step -1
    For J:=1 to M step 1
        A[I+1,J] := A[I,J]
    End
End
For J:=1 to M step 1
    A[K,J] := C[J]
End
N:=N+1
```

Înserarea unei coloane în tablou

Type

Tb = Array[1..N] Of Array[1..**M+1**] Of **Integer**

Tu = Array[1..**N**] Of **Integer**

Var

A: **Tb**

C: **Tu**

```
For J:=M to K step -1
  For I:=1 to N step 1
    A[I,J+1]:=A[I,J]
  End
End
For I:=1 to N step 1
  A[I,K]:=C[I]
End
M:=M+1
```

Prelucrarea matricei pătrate

Const

N=5

Type

TB = Array[1..N] Of Array[1.. N] Of Integer

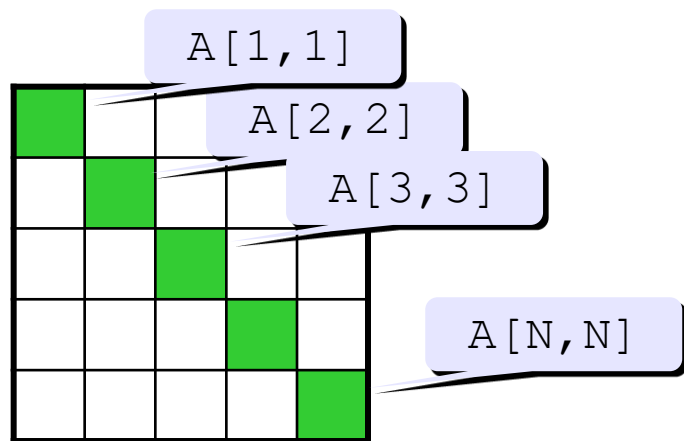
Var

A: TB

A_{11}	A_{12}	A_{13}	A_{14}	A_{15}
A_{21}	A_{22}	A_{23}	A_{24}	A_{25}
A_{31}	A_{32}	A_{33}	A_{34}	A_{35}
A_{41}	A_{42}	A_{43}	A_{44}	A_{45}
A_{51}	A_{52}	A_{53}	A_{54}	A_{55}

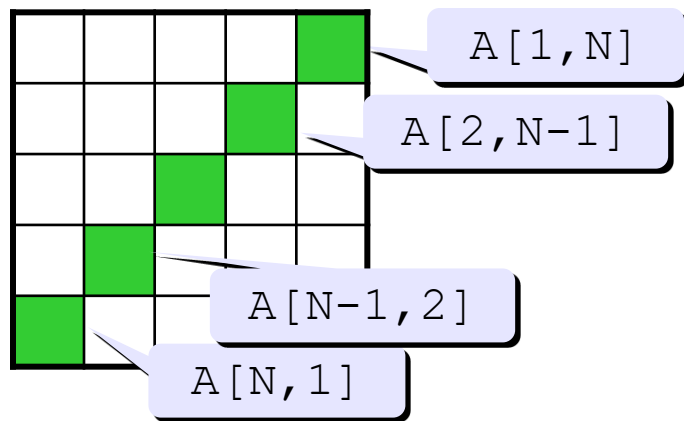
Prelucrarea matricei pătrate

Problema 1. Să se afișeze pe ecran diagonală principală a matricei pătrate din N rânduri și N coloane.



```
For I:=1 to N step 1  
  WriteInt (A[I,I])  
End
```

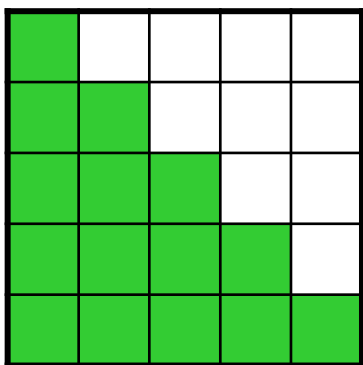
Problema 2. Să se afișeze pe ecran diagonală secundară.



```
For i:=1 to N step 1  
  WriteInt(A[I, N+1-I])  
End
```

Prelucrarea matricei pătrate

Problema 3. Să se determine suma elementelor aflate pe diagonală principală și mai jos de ea.



rândul 1: $A[1,1]$

rândul 2: $A[2,1] + A[2,2]$

...

rândul N: $A[N,1] + A[N,2] + \dots + A[N,N]$

```
S := 0
```

```
For I := 1 to N step 1
```

```
  For J := 1 to I step 1
```

```
    S := S + A[I,J]
```

```
  End
```

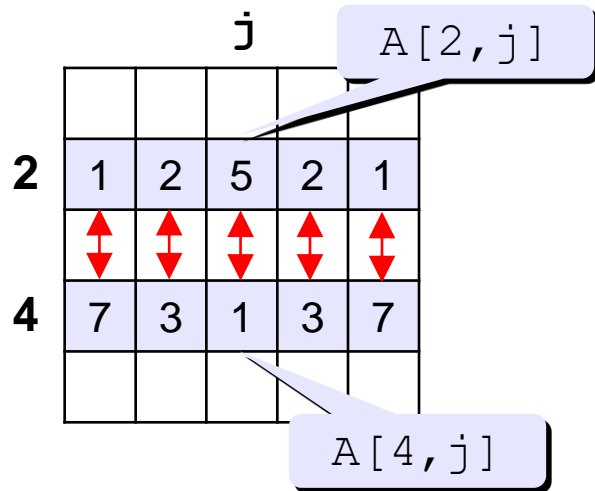
```
End
```

ciclu pentru rânduri

adunăm elementele
necesare a rândului i

Exemple

Problema 4. Interschimbarea rândurilor și coloanelor. În matricea cu N rânduri și M coloane să se schimbe cu locul al 2-lea și al 4-lea rând.



```
For j:=1 to M step 1
  c:= A[2,j]
  A[2,j]:= A[4,j]
  A[4,j]:= c
End
```

Problema 5. La coloana a treia să se adune elementele coloanei șase.

```
For i:=1 to N step 1
  A[i,3]:= A[i,3] + A[i,6]
End
```