TABLICA – ZŁOŻONA STRUKTURA DANYCH

- struktura, która zawiera kilka wartości określonego typu,
- liczba elementów tablicy i ich typ są definiowane przez programistę,
- w danej tablicy typ wszystkich jej elementów jest ten sam,
- indeksy → kolejne wzrastające wartości dowolnego typu porządkowego.

Tablica **Tab** →

10	5	0	100	2
Tab[1]	Tab[2]	Tab[3]	Tab[4]	Tab[5]

Definiowanie typu tablicowego

TYPE

Nazwa_typu = **array [** *indeks_pocz* **..** *indeks_końc* **] of** *typ_elementu*;

{ różne rodzaje indeksów}

```
Tablica   = array [ 1 .. 5 ] of Byte;          { 1, 2, 3, 4, 5 }
Tablica_2 = array [-2 .. 2] of Byte;           { -2, -1, 0, 1, 2 }
Tablica_3 = array [ 'A' .. 'E' ] of Byte;      { 'A', 'B', 'C', 'D', 'E' }
Tablica_4 = array [ '1' .. '5' ] of Byte;      { '1', '2', '3', '4', '5' }

Dni_tygodnia = ( Pon, Wt, Sr, Czw, Pt, Sob, Niedz );
Tablica_5 = array [ Pon .. Pt ] of Byte;       { Pon, Wt, Sr, Czw, Pt }
```

{ różne rodzaje elementów}

```
Tablica_6 = array [ 1 .. 5 ] of Byte;
Tablica_7 = array [ 1 .. 5 ] of Integer;
Tablica_8 = array [ 1 .. 5 ] of Char;
Tablica_9 = array [ 1 .. 5 ] of Dni_tygodnia;
Tablica_10 = array [ 1 .. 5 ] of array [ 1 .. 5 ] of Byte;   { tablica tablic }
```

UWAGA !

Definicja typu → tylko opis struktury danych,
→ nie powoduje fizycznego przydzielenia obszarów w pamięci operacyjnej komputera.

Deklarowanie zmiennych tablicowych

```
TYPE
    Tablica = array [ 1 .. 5 ] of Byte;
VAR
    Tab : Tablica;
    A, B, Tab_wyn : Tablica;
    Tab_C : array [ 1 .. 5 ] of Byte
    Tab_D, Tab_E : array [ 1 .. 5 ] of Byte
```

{ wykorzystanie stałych do definiowania ilości elementów tablicy }

```
CONST
    N = 5;
TYPE
    Tablica = array [ 1 .. N ] of Integer;
VAR
    T_1, T_2 : Tablica;
```

UWAGA !

Maksymalny rozmiar tablicy → 64 kilobajty pamięci ! *{Rozmiar obszaru pamięci zajmowanej przez tablicę = rozmiar elementu × ilość_elementów}*

np.

```
    Tablica = array [ 1 .. 10 922 ] of Real;
```

sizeof(Real) = 6 bajtów
sizeof(Tablica) = 6 × 10 922 = 65 532 *{ 65 536 = 64 × 1024 = 64 Kb}*

Przypisywanie / odczytywanie wartości elementów tablicy

```
TYPE
    Typ_tab = array [ 1 .. 4 ] of  Integer;
VAR
    Tab: Typ_tab;
    i , suma: Integer;
BEGIN
  Tab[ 1 ] := 0;
  Tab[ 2 ] := 10;
  Tab[ 3 ] := -20;
  Tab[ 4 ] := 3;

  { wczytanie wartości z klawiatury }

  Readln( Tab[ 1 ] );
  Readln( Tab[ 2 ], Tab[ 3 ] );

  Write( 'Podaj 4 element tablicy: ' );
  Readln( Tab[ 4 ] );


  { odczytywanie i wyświetlanie zawartości elementów }

  suma := Tab[2] + Tab [2] + Tab [3] + Tab [4];

  Writeln( 'Tab[1] = ' , Tab[1] : 5 );
  Writeln( 'Tab[2] = ' , Tab[2] : 5 );
  Writeln( 'Tab[3] = ' , Tab[3] : 5 );
  Writeln( 'Tab[4] = ' , Tab[4] : 5 );


  { zadawanie wartości indeksu za pośrednictwem zmiennej }
  i := 2;
  Tab[ i ] := 10;
  Tab[ 2 ] := 10;


  Write( 'Podaj który element tablicy chcesz zmienic: ' );
  Readln( i );
  Write( 'Podaj nową wartość Tab[' , i , '] = ' );
  Readln( Tab[ i ] );
END;
```

Zastosowanie instrukcji repetycyjnej "for" do operacji na tablicach

```
CONST
    IL_ELEM = 100 ;
TYPE
    Typ_tab = array [ 1 .. IL_ELEM ] of  Integer;
VAR
    Tab: Typ_tab;
    i , suma: Integer;

BEGIN
    { inicjowanie zawartości tablicy }
    for i := 1 to IL_ELEM do
        Tab[ i ] := 0;

    { wczytanie wartości z klawiatury }
    for i := 1 to IL_ELEM do
        begin
            Write( 'Podaj Tab[' ,  i  , '] = ' );
            Readln( Tab[ i ] );
        end;

    { wyświetlenie zawartości elementów }
    for i := 1 to IL_ELEM do
        Writeln( 'Tab[' ,  i  , '] = ' , Tab[ i ] : 5);

    { zsumowanie wartości elementów }
    suma := 0;
    for i := 1 to IL_ELEM do
        suma := suma + Tab[ i ];
    Writeln( 'Suma wartości elementów = ' , suma );

END;
```

Dyrektywa kompilatora {$R+} powoduje, że w czasie wykonania programu odbywa się kontrola zakresu indeksów tablicy.

Tablica dwuwymiarowa

= jednowymiarowa tablica, której elementami są także tablice.

np. **Macierz**

	Macierz[1,1]	Macierz[1,2]	Macierz[1,3]	Macierz[1,4]
Macierz[1] →	**10**	**0**	**2**	**16**
Macierz[2] →	**1**	**4**	**100**	**255**
	Macierz[2,1]	Macierz[2,2]	Macierz[2,3]	Macierz[2,4]

Definiowanie tablicy dwuwymiarowej

```
TYPE
    T_jednowym = array[ 1..4 ] of Byte;
    T_dwuwym = array[ 1..2 ] of  T_jednowym;
VAR
    Macierz : T_dwuwym ;
```

..... *1 sposób*

```
TYPE
    T_dwuwym =array[ 1..2 ] of  array[ 1..4 ] of Byte;
VAR
    Macierz : T_dwuwym;
```

..... *2 sposób*

```
TYPE
    T_dwuwym =array[ 1..2 , 1..4 ] of Byte;
VAR
    Macierz : T_dwuwym;
```

..... *3 sposób*

```
program Tabliczka_Mnozenia;
{$R+}
uses Crt;
const N = 10;
type   Mac = array [ 1..N, 1..N ] of Word;
varTabliczka : Mac;

procedure LICZ( var T:Mac );
    var wiersz, kolumna : Integer;
begin
    for wiersz := 1 to N do
        for kolumna := 1 to N  do
            T[ wiersz, kolumna ] := wiersz ### kolumna;
end; {---------------------------------------------------------- LICZ }

procedure WYSWIETL( T:Mac );
    var wiersz, kolumna : Integer;
begin
    for wiersz := 1 to N  do
        begin
            for kolumna := 1 to N  do
                Write( T[ wiersz, kolumna ] : 4 );
            Writeln;
        end;
end; {-----------------------------------------------------WYSWIETL}



begin                                          {program główny}
    ClrScr;
    LICZ( Tabliczka );
    WYSWIETL( Tabliczka );
    GotoXY(1,25);
    Write('<Enter> konczy program.');
    Readln;
    ClrScr;
end.
```