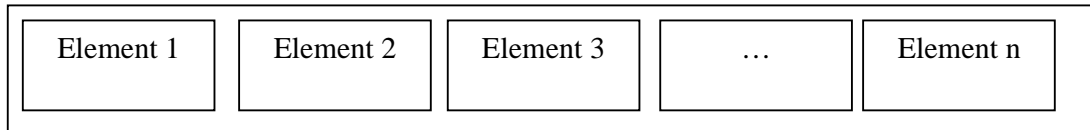


## Tablice

---

**Tablica** służy do przechowywania danych tego samego typu. Tablice składają się z elementów różniących się indeksami.



Deklaracja zmiennej tablicowej:

```
var tablica : array [ poczatek .. koniec ] of typ_danych;
```

Na przykład, deklaracja tablicy 4-elementowej składającej się z liczb całkowitych:

```
var tablica : array[1..4] of integer;
```

Aby odwołać się do elementu w tablicy, podajemy nazwę zmiennej oraz indeks elementu w nawiasach kwadratowych [ ], np. wyświetlenie czwartego elementu powyższej tablicy:

```
writeln('Czwarty element: ', tablica[4]);
```

### Przykładowy program

---

Program wczytuje w pętli elementy do tablicy, a następnie wyświetla jej zawartość:

```
program tablica1;
uses crt;
var i :integer;
    tab : array[1..10] of integer;
begin
  clrscr;
  for i:=1 to 10 do
  begin
    write('Wprowadź wartość elementu nr ', i, ': ');
    readln(tab[i]);
  end;

  writeln('Zawartosc tablicy: ');
  for i:=1 to 10 do
    write(tab[i], ' ');

  readkey;
end.
```

## Procedury i funkcje

---

**Procedura** jest grupą instrukcji, którą wywołujemy poprzez jej nazwę. Procedury stosujemy, aby nie powtarzać wielokrotnie tego samego kodu źródłowego.

Deklaracja procedury:

```
procedure nazwa_procedury (parametry);  
var zmienna : typ;      { <- deklaracje zmiennych }  
begin  
    instrukcja1;  
    instrukcja2;  
    ...  
end;
```

### Przykład (bez parametrów):

Procedura wyświetlająca dane adresowe:

```
procedure adres;  
begin  
    writeln('Jan Kowalski');  
    writeln('ul. Akacjowa 10');  
    writeln('00-010 Warszawa');  
end;
```

### Przykład (z parametrem):

Procedura wyświetlająca ciąg arytmetyczny n liczb:

```
procedure ciag_aryt(n:integer);  
var i:integer;  
begin  
    for i:=1 to n do  
        write(i, ' ');  
    writeln;  
end;
```

**Funkcja** przypomina działaniem procedurę, ale dodatkowo zwraca wartość przez swoją nazwę.

Deklaracja funkcji:

```
function nazwa(parametry) : typ_wyniku;  
var zmienne:typ;  
begin  
    instrukcje;  
    nazwa := przypisanie_wyniku;  
end;
```

**Przykład:**

Funkcja obliczająca iloczyn dwóch liczb całkowitych:

```
function iloczyn(x:integer; y:integer) : integer;
begin
    iloczyn := x*y;
end;
```

**Przykład:**

Funkcja obliczająca sumę elementów w tablicy:

```
function suma(tab:array of integer; n:integer) of real;
var i,s:integer;
begin
    s:=0;
    for i:=1 to n do
        s:=s+tab[i];
    suma:=s;
end;
```

**Zadanie**

---

Napisz program, w którym oceny wprowadzane przez użytkownika (procedura *wczytaj\_oceny*) są wprowadzane do tablicy (max. 10 ocen). Program ma za zadanie:

- wyświetlić wprowadzone oceny, od najmniejszej do największej (procedura *wyświetl\_oceny*)
- obliczyć średnią ocen (funkcja *srednia*)
- podać najwyższą ocenę (funkcja *najwyzsza*),
- podać najniższą ocenę (funkcja *najnizsza*).

Procedura sortująca:

```
procedure sortuj(var tab: array of integer; n:integer);
var i,j,tmp:integer;
begin
    for i := 1 to n do
        for j := n downto i do
            if tab[j-1] > tab[j]
            begin
                tmp = tab[j];
                tab[j] = tab[j-1];
                tab[j-1] = tmp;
            end;
        end;
    end;
```