

PP. Laboratorium 3

Zadanie pokazujące wykorzystanie procedur i funkcji, oraz różnych sposobów przekazywania parametrów.

TREŚĆ:

Napisz program który:

- Uzupełni tablicę jedno wymiarową losowymi liczbami z zakresu podanego przez użytkownika
- Wyświetli tablicę na ekranie
- Wypisze MEDIANE (bez zmiany kolejności liczb w tablicy)
- Wypisze jeszcze raz tablicę na ekranie (aby sprawdzić, czy dane dalej nie są posortowane)
- Posortuje i wyświetli tablicę na ekranie

PRZYKŁADOWY EFEKT DZIAŁANIA PROGRAMU:

```
      8   3   3   5   2   7   3   5  10   1   6
Mediana: 5.000
      8   3   3   5   2   7   3   5  10   1   6
posortowane
      1   2   3   3   3   5   5   6   7   8  10
```

POTRZEBNE PROCEDURY I FUNKCJE:

1. Procedury:
 - a. Uzupełniająca tablicę losowymi liczbami
 - b. Wyświetlająca tablicę
 - c. Sortująca
2. Funkcje:
 - a. Zwracająca medianę

PROGRAM GŁÓWNY

```
program ProceduryFunkcje;
uses Tablice;
var
  tab:Ttable1;
begin
  randFill(tab,0,10);
  showTable(tab);
  writeln('Mediana: ', median(tab):0:3);
  showTable(tab);
  writeln('Posortowane');
  bubbleSort(tab);
  showTable(tab);
  readln;
end.
```

MODUŁ:

```
unit Tablice;  
  
interface
```

Zadeklarowanie stałej oznaczającej maksymalny index tablicy:

```
const  
    max = 10;
```

Deklaracja własnego typu danych – typ tablicowy

```
type  
    TtableI = array[0..max] of Integer;
```

Deklaracja nagłówków potrzebnych procedur i funkcji

```
procedure showTable(const tab:TtableI);  
procedure randFill(var tab:TtableI;min,max:Integer);  
procedure bubbleSort(var tab:TtableI);  
function median(tab:TtableI):Real;
```

Implementacja zadeklarowanych procedur i funkcji:

```
implementation
```

Procedura do wyświetlania tablicy.

Procedura przyjmuje jako parametr tablicę (zadeklarowanego typu) przekazywaną za pomocą modyfikatora **CONST** (przekazujemy ‘oryginalna’ zmienną, ale bez możliwości jej zmiany w procedurze).

Do iteracji po tablicy wykorzystujemy funkcje **LOW** i **HIGH**.

```
procedure showTable(const tab: TtableI);  
var  
    i:Integer;  
begin  
    for i:=low(tab) to High(tab) do  
        write(tab[i]:4);  
        writeln;  
    end;
```

Procedura uzupełniająca tablicę losowymi wartościami z zakresu

Procedura przyjmuje jako parametr tablicę (zadeklarowanego typu) przekazywaną za pomocą modyfikatora **VAR** (przekazujemy ‘oryginalną’ zmienną z możliwością jej modyfikacji).

Do iteracji po tablicy wykorzystujemy funkcje **LOW** i **HIGH**.

```
procedure randFill(var tab: TtableI;min,max:Integer);  
var  
    i:Integer;  
begin  
    randomize;  
    for i:=Low(tab) to High(tab) do  
        begin  
            tab[i]:=Random(max-min+1)+min;  
        end;  
    end;
```

Procedura sortująca liczby w tabeli

Procedura przyjmuje jako parametr tablicę (zadeklarowanego typu) przekazywaną za pomocą modyfikatora **VAR** (przekazujemy 'oryginalną' zmienną z możliwością jej modyfikacji).

Do iteracji po tablicy wykorzystujemy funkcje **LOW** i **HIGH**.

Jest to jeden ze sposobów implementacji algorytmu **sortowania bąbelkowego**

W pętli for porównujemy wybrany element z następnym i jeżeli obecny jest większy od następnego zamieniamy je miejscami (pętla iterujemy od pierwszego do przedostatniego indeksu tablicy).

Pętla for znajduje się w pętli WHILE która będzie się wykonywać dopóki jakkolwiek zamiana w danym obiegu pętli for zajdzie.

```
procedure bubbleSort(var tab: TtableI);
var
  i,tmp:Integer;
  sorted:Boolean;
begin
  sorted:=false;
  while not sorted do
  begin
    sorted:=true;
    for i:=low(tab) to high(tab)-1 do
    begin
      if(tab[i]>tab[i+1]) then
      begin
        tmp:=tab[i];
        tab[i]:=tab[i+1];
        tab[i+1]:=tmp;
        sorted:=false;
      end;
    end;
  end;
end;
```

Funkcja zwracająca medianę

Funkcja przyjmuje jako parametr tablicę (zadeklarowanego typu) przekazywaną **BEZ** modyfikatora (powoduje to utworzenie wewnątrz funkcji kopii tablicy – jest to niezbędne gdyż do wyznaczenia mediany potrzebujemy uporządkowanej tablicy, ale nie chcemy modyfikować oryginalnej tablicy).

W przypadku nieparzystej ilości liczb zwraca środkową wartość, w przeciwnym średnią z dwóch środkowych wartości (dlatego zwraca typ **REAL**);

Zwracanie wartości funkcji za pomocą dwóch różnych sposobów (**EXIT** i **nazwa:=**)

```
function median(tab: TtableI): Real;
var
  i:Integer;
begin
  i:=Length(tab);
  bubbleSort(tab);
  if(i mod 2 = 0) then
    median:=((tab[i div 2]+tab[(i div 2)-1]))/2)
  else
    exit(tab[(i div 2)]);
end;
```

Koniec modułu

```
| End.
```