

```
PROGRAM Komunikat; {procedura}
```

```
USES
```

```
Crt;
```

```
PROCEDURE Wypisz_Komunikat; {Definicja procedury – bez parametrow}
```

```
BEGIN
```

```
Writeln('To jest procedura, ktora wypisuje komunikat na ekranie');
```

```
Writeln('To jest komunikat');
```

```
END; {Koniec procedury}
```

```
BEGIN {main}
```

```
ClrScr;
```

```
Writeln('Przed wywołaniem procedury');
```

```
Wypisz_Komunikat; {wywołanie procedury}
```

```
Writeln('Po wywołaniu procedury');
```

```
REPEAT UNTIL KeyPressed;
```

```
END.{main}
```

```
PROGRAM Przekazywanie_parametrow_przez_wartosc; {Procedura}
```

```
USES
```

```
Crt;
```

```
PROCEDURE Wypisz_liczbe(liczba : INTEGER);
```

```
BEGIN
```

```
Writeln('Wartosc parametru wynosi ', liczba);
```

```
END;
```

```
BEGIN
```

```
ClrScr;
```

```
Wypisz_liczbe(10);
```

```
Wypisz_liczbe(100);
```

```
REPEAT UNTIL KeyPressed;
```

```
END.
```

```
PROGRAM Obliczanie_silni; {Funkcja}
```

```
USES Crt;
```

```
VAR
```

```
liczba, i, n : INTEGER;
```

```
FUNCTION Silnia(liczba : INTEGER) : LONGINT;
```

```
BEGIN
```

```
IF liczba > 1 THEN
```

```
Silnia:=liczba*Silnia(liczba-1)
```

```
ELSE
```

```
Silnia:=1;
```

```
END; {Silnia}
```

```
BEGIN {main}
ClrScr;
Writeln('Obliczanie silni z danej liczby calkowitej');
Writeln('Podaj liczbe');
Readln(n);
FOR i:=0 TO n DO
Writeln(' ', i, ' != ', Silnia(i));

REPEAT UNTIL KeyPressed;
END. {main}
```

program **CzytanieHasla**; {Funkcja, typ łańcuchowy, ReadKey}

```
uses Crt;
```

```
type
```

```
string10 = string[10]; {Deklaracja typu łańcuch o dłuę 10}
```

```
function Haslo:string10; {Funkcja}
```

```
var
```

```
i : integer; { licznik znakow }
```

```
c : char;
```

```
s : string; { bufor na haslo }
```

```
begin {funkcji Haslo}
```

```
s:=""; { zaczynamy od pustego lancucha }
```

```
writeln('Podaj haslo:');
```

```
for i:=1 to 10 do { czytaj kolejne znaki }
```

```
begin
```

```
c:=ReadKey; { czytaj znak }
```

```
if c = #13 then Break { Enter - koniec }
```

```
else s:=s+c { dodaj do hasla }
```

```
end;
```

```
Haslo:=s
```

```
end; {funkcji Haslo}
```

```
Begin {main}
```

```
writeln(Haslo);
```

```
readln
```

```
End.
```

PROGRAM **Przekazywanie_procedurom_tablic**;

```
USES Crt;
```

```
CONST rozmiar = 10;
```

```
TYPE macierz = ARRAY[1..rozmiar, 1..rozmiar] OF INTEGER;
```

```
VAR tablica : macierz;
```

```
PROCEDURE Zapelnij_tablice(VAR tablica : macierz; rozmiar : INTEGER);
```

```
VAR
```

```
i,j :INTEGER;
```

```
BEGIN
```

```
Randomize;
```

```
FOR i:=1 TO rozmiar DO
```

```
BEGIN
```

```
FOR j:=1 TO rozmiar DO
```

```
BEGIN
```

```
IF i =j THEN tablica[i,j]:=Random(10) {Wpisywanie liczb pseudolosowych}
```

```
{od 1 do 9 na przekatnej tablicy}
```

```
ELSE tablica[i,j]:=0;
```

```
END; {j}
```

```
END; {i}
```

```
END;
```

```
PROCEDURE Wydruk_tablicy(VAR tablica : macierz; rozmiar : INTEGER);
```

```
VAR
```

```
i,j :INTEGER;
```

```
BEGIN
```

```
Writeln('Wydruk zawartosci tablicy');
```

```
Writeln;
```

```
FOR i:= 1 TO rozmiar DO
```

```
BEGIN
```

```
FOR j:= 1 TO rozmiar DO
```

```
BEGIN
```

```
Write(tablica[i, j], ' ');
```

```
END;
```

```
Writeln;
```

```
END;
```

```
END;
```

```
PROCEDURE Suma_na_przekatnej(VAR tablica : macierz; rozmiar : INTEGER);
```

```
VAR
```

```
i, suma : INTEGER;
```

```
BEGIN
```

```
suma:=0;
```

```
FOR i:=1 TO rozmiar DO
```

```
BEGIN
```

```
suma:=suma + tablica[i,i]; {-- Sumowanie elementow na przekatnej --}
```

```
END;
```

```
Writeln;
```

```
Writeln('Suma elementow na przekatnej = ',suma);
```

```
END;
```

```
BEGIN {main}
ClrScr;
Zapelnij_tablice(tablica,rozmiar);
Wydruk_tablicy(tablica,rozmiar);
Suma_na_przekatnej(tablica,rozmiar);

REPEAT UNTIL KeyPressed;
END. {main}
```

```
PROGRAM KULA; { naglowek programu}
{ poczatek BLOKu DEKLARACJI}
```

```
CONST PI=3.14;
VAR r : real; {promien - typu rzeczywistego}
v : real; {objetosc - typu rzeczywistego}
i: integer;
```

```
PROCEDURE dane (VAR x: real);
```

```
begin
write('Promien=');
readln(x);
end; {dane}
```

```
PROCEDURE obliczenia ( x: real; VAR y: real);
```

```
{ x,y - zmienne lokalne
x-promien - przekazywany przez wartosc
z programu glownego
y-objetosc - przekazywany przez zmienna
na zewnatrz procedury do programu glown }
```

```
var x3: real; { r*r*r - promien do potegi 3 - zmienna lokalna}
```

```
Begin {pocz bloku operacji procedury obliczenia}
x3:=x*x*x;
y:=4.0/3.0*pi*x3;
End;
```

```
PROCEDURE wyniki(promien,objetosc: real);
```

```
Begin
writeln;
writeln('OBJETOSC KULI o promieniu ',promien:7:3,' wynosi ',objetosc:10:3);
writeln;
writeln('...Nacisnij ENTER...');
```

```
readln;
End; {wyniki}

{ Koniec bloku deklaracji}

{ Blok operacji - zacznie sie}

BEGIN { Kula – program główny}
dane(r);
obliczenia(r,v); { r wchodzi do procedury, a wychodzi z niej v}
wyniki(r,v);
END. { Kula}
{Koniec bloku operacji}
```

```
PROGRAM KULA1; { naglowek programu}

{ poczatek BLOKu DEKLARACJI}

USES DOS,CRT;

CONST ESC=#27; {stala}

{Zmienne globalne}

VAR r : real; {promien - typu rzeczywistego}
v : real; {objetosc - typu rzeczywistego}
i: integer; { zmienna typu calkowitego}
z: char; {zmienna typu znak}

PROCEDURE dane (VAR x: real);

begin
textcolor(red);
writeln('Obliczenie NR ',i);
writeln;
textcolor(yellow);
write('Promien=');
readln(x);
end; {dane}

PROCEDURE obliczenia ( x: real; VAR y: real);
{ x,y - zmienne lokalne
x-promien - przekazywany przea wartosc
z programu glownego
y-objetosc - przekazywany przez zmienna-
na zewnatrz procedury do programu glown }
var x3: real; { r*r*r - promien do potegi 3 - zmienna lokalna}

Begin {pocz bloku operacji procedury obliczenia}
x3:=x*x*x;
```

```
y:=4.0/3.0*pi*x3;  
End;
```

```
PROCEDURE wyniki(promien,objetosc: real);  
Begin  
writelN;  
textcolor(green);  
writelN('PI=',pi);  
writelN;  
textcolor(blue);  
write('OBJETOSC KULI o promieniu ');  
textcolor(magenta);  
writelN(promien:7:3);  
textcolor(green);  
write(' wynosi ');  
textcolor(lightcyan);  
writelN(objetosc:10:3);  
textcolor(white);  
writelN;  
writelN;  
write('BY zakonczyc obliczenia nacisnij');  
highvideo;  
textcolor(yellow+blink);  
writelN(' ESC...!');  
textcolor(white);  
z:=Readkey;  
normvideo;  
writelN;  
End; {wyniki}
```

```
{ Koniec bloku deklaracji}
```

```
{ Blok operacji - zacznie sie}
```

```
BEGIN { Kula }  
clrscr;  
textmode(c40);  
i:=0;  
repeat  
i:=i+1;  
dane(r);  
obliczenia(r,v); { r wchodzi do procedury, a wychodzi z niej v }  
wyniki(r,v);  
until (z=ESC);  
END. { Kula }  
{Koniec bloku operacji}
```