

PRIAMA REKURZIA

1. Napíšte rekurzívnu procedúru, ktorá pri vstupe čísla n vypíše čísla od 1 po n.

```
procedure vypisl(n:integer);
begin
  if n>0 then
  begin
    vypisl(n-1);
    write(n);
  end;
end;
```

volanie procedúry: vypisl(10)

nerekurzívny prepis

```
procedure vypis(n:integer);
var i: integer;
begin
  for i:=1 to n dopredu write(i);
end;
```

2. Napíšte rekurzívnu procedúru, ktorá pri vstupe čísla n vypíše čísla od n po 1.

```
procedure vypis(n:integer);
begin
  if n>0 then
  begin
    write(n);
    vypis(n-1);
  end;
end;
```

volanie procedúry: vypis(10)

nerekurzívny prepis

```
procedure vypis(n:integer);
var i:integer;
begin
  for i:=n dopredu wnto 1 write(i);
end;
```

3. Napíšte rekurzívnu procedúru, ktorá vypíše zadaný reťazec S obrátene.

```
procedure obrat(s:string);
begin
  if length(s)>0 then
  begin
    obrat(copy(s,2,length(s)-1));
    write(sbeginlend);
  end;
end;
```

volanie procedúry: obrat(ahoj);

nerekurzívny prepis

```
procedure obrat(s:string);
```

```

var i: integer;
begin
  for i:=length(s) to 1 dopredu write(sbeginiend);
end;

```

4. Napíšte rekurzívnu procedúru, ktorá pri vstupe čísla n , znakov a, b, c vypíše. $a^n c b^n$, napr $n=3, a=a, b=b, c=c$, výpis bude $aaacbbb$.

```

procedure retazec1(i:integer; a,b,c:string);
begin
  if i>0 then
  begin
    write(a);
    retazec1(i-1,a,b,c);
    write(b);
  end
  else write(c);
end;

```

volanie procedúry: `retazec1(5,h,i,j);`

nerekurzívny prepis

```

procedure retazec1(n:integer; a,b,c:string);
var i:integer;
begin
  for i:=1 to n dopredu write(a);
  write(c);
  for i:=1 to n dopredu write(b);
end;

```

5. Napíšte rekurzívnu procedúru, ktorá pri vstupe prirodzených čísel n, k, l a znakov a, b, c vypíše $a^{kn} c b^{ln}$, napr $n=3, a=a, b=b, c=c, k=1, l=2, n=3$ výpis bude $aaacbbbbbb$.

```

procedure retazec2(n,k,l:integer; a,b,c:string);
var i:integer;
begin
  if i>0 then
  begin
    for i:=1 to k dopredu write(a);
    retazec2(n-1,k,l,a,b,c);
    for i:=1 to l dopredu write(b);
  end
  else write(c);
end;

```

volanie procedúry: `retazec1(3,1,2,h,i,j);`

nerekurzívny prepis

```

procedure retazec2(n,k,l:integer; a,b,c:string);
var i:integer;
begin
  for i:=1 to n*k dopredu write(a);
  write(c);
  for i:=1 to n*l dopredu write(b);
end;

```

6. Napíšte rekurzívnu procedúru, ktorá pri vstupe čísla n a znakov a, b, c vypíše $a^n b^n$, napr $n=3, a=a, b=b$, výpis bude $aaabbb$.

```

procedure retazec3(i:integer; a,b:string);

```

```

begin
  if i>0 then
    begin
      write(a);
      retazec3(i-1,a,b);
      write(b);
    end
  end;
end;

```

volanie procedúry: `retazec1(5,h,i);`

nerekurzívny prepis

```

rocedure retazec3(n:integer; a,b:string);
var i:integer;
begin
  for i:=1 to n dopredu write(a);
  for i:=1 to n dopredu write(b);
end;

```

7. Napíšte rekurzívnu procedúru, ktorá načíta n čísel, pokiaľ nenačíta 0 a vypíše ich v opačnom poradí v akom boli načítane.

```

procedure citaj;
var n:integer;
begin
  readln(n);
  if n<>0 then citaj;
  write(n);
end;

```

volanie procedúry: `nacitaj();`

8. Napíšte rekurzívnu procedúru, ktorá vypíše reťazec v normálnom tvare a hneď k nemu aj obrátene.

```

procedure tiktak(s:string);
begin
  if length(s)>0 then
    begin
      write(sbeginlend;);
      tiktak(copy(s,2,length(s)-1));
      write(sbeginlend;);
    end;
end;

```

volanie procedúry: `tiktak(ahoj);`

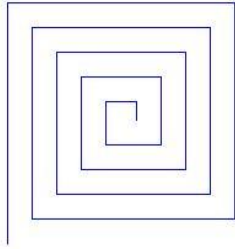
nerekurzívny prepis

```

procedure tiktak(s:string);
var i:integer;
begin
  write(s)
  for i:=length(s) dopredu wnto 1 write(sbeginiend;);
end;

```

9. Napíšte rekurzívnu procedúru, ktorá nakresli štvorcovú špirálu podľa predlohy na obrázku.



```
procedure spirala1(a,stupen:integer);
begin
  if stupen>0 then
    begin
      dopredu(a);
      vlavo(90);
      spirala1(a+10,stupen-1);
    end;
  end;
end;
```

volanie procedúry: spirala1 5 20

nerekurzívny prepis

```
procedure spirala1 (a,stupen:integer);
begin
  while stupen>0 do
    begin
      dopredu(a);
      vlavo(90);
      a:=a+10;
      stupen:=stupen-1;
    end;
  end;
end;
```

- 10. Naprogramujte náhodný pohyb korytnačky a to tak, že korytnačka sa bude otáčať o náhodný uhol, bude meniť náhodne farbu pera a bude sa posúvať dopredu predu krokom 10.**

```
procedure pohyb-nahodny
begin
  vlavo(random(360));
  dopredu(10);
  setcolor(random(16));
  pohyb-nahodny;
end;
```

volanie procedúry: nahodny-pohyb

- 11. Napíšte procedúru, v ktorej korytnačka bude kresliť štvorec n krát, bez použitia príkazu opakuj, ale s náhodnou farbou štvorca a na rôznych miestach.**

```
procedure stvorec3 (a:integer);
var i:integer;
begin
  if a>5 then
    begin
      pero_hore;
      setXYSur(random(640),radnom(480));
      setcolor(random(16));
      pero_dole
      polygon
      begin for i:=1 to 4 do
```

```

begin
  dopredu(50);
  vlavo(90);
end;
end;
stvorec3(a-1);
end;
end;

```

12. Napíšte rekurzívnu procedúru, ktorá sčíta kladné čísla A a B bez použitia sčítania týchto dvoch čísel.

```

procedure sucet(a,b,integer);
begin
  if a>0 then sucet(a-1,b+1)
  else writeln('sucet cisel je ',b);
end;

```

volanie procedúry: `sucet(125,3658);`

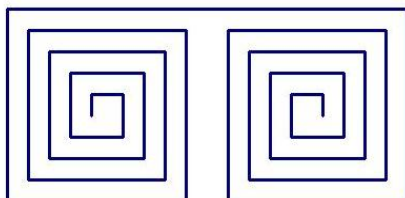
nerekurzívny prepis

```

procedure sucet(a,b,integer);
begin
  while a>0 dopredu
  begin
    a:=a-1;
    b:=b+1;
  end
  writeln('sucet cisel je ',b);
end;

```

13. Napíšte rekurzívnu procedúru, ktorá nakreslí dve štvorcové špirály podľa predlohy na obrázku.



```

procedure spir1 (a,stupen:integer);
begin
  if stupen>0 then
  begin
    dopredu(a);
    vlavo(90);
    spir1 (a+10,stupen-1);
    dopredu(a);
    vlavo(90);
  end
  else
  begin
    vpravo(90);
  end;
end;

```

volanie procedúry: `spir1 5 30`

nerekurzívny prepis

```

procedure spirala1(a,stupen:integer);
begin
  pom:=stupen;
  while pom>0 do
  begin
    dopredu(a);
    vlavo(90);
    a:=a+10;
    pom:=pom-1;
  end;
  vpravo(90);
  while pom<>stupen
  begin
    dopredu(a);
    vlavo(90);
    a:=a-10;
    pom:=pom+1;
  end;
end;

```

14. Napíšte rekurzívnu procedúru, ktorá vypočíta najmenší spoločný násobok čísel m a n .

Dá sa použiť vzorec
$$NSN(a,b) = \frac{a*b}{NSD(a,b)}$$

```

procedure nsn(a,b,c:integer);
begin
  if a=0 then writeln('nsn zadanych čísel je ',b)
  else if b=0 then writeln('nsn zadanych čísel je ',a)
  else if (a>b) then nsn(a-b,b,c)
  else if (b>a) then nsn(a, b-a,c)
  else writeln('nsn zadanych čísel je ',c/a)
end;

```

volanie procedúry: nsn(50,123,50*123);

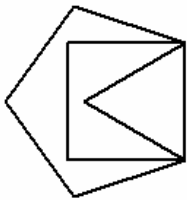
nerekurzívny prepis

```

procedure nsn(a,b:integer);
var a1,b1,pom:integer;
begin
  a1:=a;
  b1:=b;
  if a=0 then writeln('nsn zadanych čísel je ',b)
  else if a=b then writeln('nsn zadanych čísel je ',a)
  else
  begin
    while a<>b dopredu
    begin
      if a>b then a:=a-b
      else b:=b-a;
    end;
    writeln('nsn zadanych čísel je ',a1*b1/a);
  end;
end;

```

15. Pomocou chvostovej rekurzie nakreslite pravidelné n-uholníky, ktoré majú jednu stranu spoločnú a sú dopredu seba vnorené, podľa predlohy na obrázku. Kreslenie sa ukončí trojuholníkom.



```
procedure uholniky(n:integer);
var i:integer;
begin
  if n>2 then
    begin
      for i:=1 to n do
        begin
          dopredu(50);
          vlavo(360/n);
        end;
      uholniky(n-1);
    end;
end;
```

volanie procedúry: uholnik 7

nerekurzívny prepis

```
procedure uholniky(n:integer);
var i,j:integer;
begin
  pom:=n;
  for i:=1 to n-2 do
    begin
      for j:=1 to pom do
        begin
          dopredu(50);
          vlavo(360/pom);
        end;
      pom:=pom-1;
    end;
end;
```

16. Napíšte rekurzívnu procedúru, ktorá vypíše všetky možnosti, ako sa dá rozdeliť tyč s dĺžkou N na diely s dĺžkou 1 a 2.

```
program tyc;
var k,n:integer;
    casti: arraybegin1..50end; of integer;
procedure rozdel(n:integer);
var i:integer;
begin
  if n>1 then
    begin
      k:=k+1;
      castibeginkend:=1;
      rozdel(n-1);
      castibeginkend:=2;
      rozdel(n-2);
      k:=k-1;
    end
  else
    begin
      for i:=1 to k dopredu write(castibeginiend; ' ');
      if n=1 then write('1');
      writeln;
    end;
end;
```

```

    end;
end;

begin
    k:=0;
    readln(n);
    rozdel(n);
    readln;
end.

```

volanie procedúry: rozdel(7);

17. Napíšte rekurzívnu procedúru, ktorá vypíše všetky permutácie čísla n.

```

type císla=set of 0..9;
var mnozina:císla;
    pole:arraybegin1..50end;of integer;
procedure perm(i,vnorenje:integer);
var l,k:integer;
begin
    for k:=1 to i dopredu
    begin
        if not (k in mnozina) then
            begin
                mnozina:=mnozina+beginkend;;
                polebeginvnorenjeend;:=k;
                if vnorenje=i-1 then
                    begin
                        for l:=0 to i-1 dopredu write(polebeginlend;);
                        writeln;
                    end;
                    perm(i,vnorenje+1);
                    mnozina:=mnozina-beginkend;;
                end
            end;
        end;
    end;
end;

```

mnozina:=beginend;;

volanie procedúry: perm(4,0);

18. Napíšte rekurzívnu procedúru, prevedie dekadické číslo do binárnej sústavy.

```

procedure dec(d,i:longint);
begin
    if i>0 then
        begin
            dec(d shr 1,i-1);
            write(d and 1);
        end
    end;
end;

```

volanie procedúry: dec(2,32);

NEPRIAMA REKURZIA

- 1 Pomocou nepriamej rekurzíe nakreslite obrázok podľa predlohy do zadaného stupňa (ak sa zadá 1 tak sa nakreslí len čiara, ak 2 tak sa nakreslí čiara a neviditeľná čiara, atď.).

```

procedure ciara(a:integer);
begin
  if a>0 then
    begin
      dopredu(30);
      nic(a-1);
    end;
end;

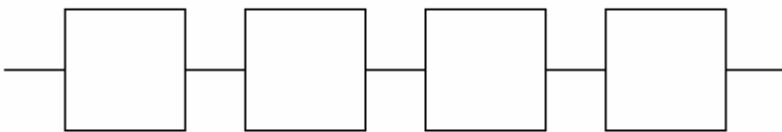
```

```

procedure nic(a:integer);
begin
  if a>0
  begin
    pero_hore;
    dopredu(10);
    pero_dole;
    ciara(a-1);
  end;
end;

```

2 Pomocou nepriamej rekurzije nakreslite obrázok podľa predlohy do zadaného stupňa (ak sa zadá 1 tak sa nakreslí len čiara, ak 2 tak sa nakreslí čiara a štvorec, atď.).



```

procedure ciara(a:integer);
begin
  if a>0 then
    begin
      pero_dole;
      dopredu(10);
      stvor(a-1);
    end;
end;

```

```

procedure stvor(a:integer);
var i:integer;
begin
  if a>0 then
    begin
      setcolor(0);
      vpravo(90);
      dopredu(25);
      vlavo(90);
      for i:=1 to 3 do
        begin
          dopredu(50);
          vlavo(90);
        end;
      dopredu(25);
      vlavo(90);
      pero_hore;
      dopredu(25);
      vypln;
      dopredu(25);
      pero_dole;
      ciara(a-1);
    end;
end;

```

3 Pomocou nepriamej rekurzije nakreslite obrázok podľa predlohy do zadaného stupňa (ak sa zadá 1 tak sa nakreslí len kruh, ak 2 tak sa nakreslí kruh a štvorec, atď.).



```

procedure kruhs(a:integer);
begin
  if a>0 then
    begin

```

```

procedure stvor(a:integer);
varf i:integer;
begin
  if a>0 then

```

```

nechfv "cierna
dopredu(25);
circle(getx,gety,50);
dopredu(25);
stvor(a-1);
end;
end;

```

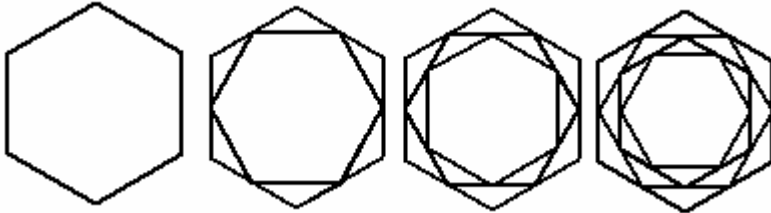
```

begin
  nechfv "biela;
  vpravo(90);
  dopredu(25);
  vlavo(90);
  for i:=1 to 3 do
  begin
    dopredu(50)
    vlavo(90)
  end;
  dopredu(25);
  vlavo(90);
  pero_hore;
  dopredu(25);
  vypln;
  dopredu(25);
  pero_dole;
  kruhs(a-1);
end;
end;

```

volanie procedúry: kruhs 10

- 4** Pomocou nepriamej rekurzcie nakreslite obrázok podľa predlohy do zadaného stupňa (nakreslite šesťuholníky umiestnene v sebe, tak že sa budú dotýkať vrcholy menšieho šesťuholníka v strede strán väčšieho šesťuholníka. Šesťuholníky budú mať dĺžku hrany zadané číslo a počet šesťuholníkov v sebe bude druhé zadané číslo. Jedna procedúra bude kresliť šesťuholníky a druhá bude otáčať korytnačku do správneho smeru a nastaví ju na miesto odkiaľ sa bude kresliť ďalší šesťuholník).



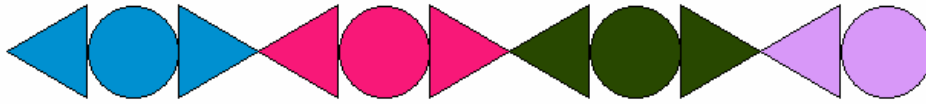
```

procedure
sestuholnik(a:real,n:integer);
var i:integer;
  if n>0 then
  begin
    for i:=1 to 6 do
    begin
      dopredu(a);
      vlavo(60);
    end;
    otoc(a,n-1);
  end;
end;

procedure otoc (a:real,n:integer);
begin
  dopredu(a/2);
  vlavo(30);
  sestuholnik((a*sqrt(3))/2,n);
end;

```

- 5** Pomocou nepriamej rekurzcie nakreslite obrázok podľa predlohy do zadaného stupňa (nakresli trojuholník jedným smerom, k nemu kruh a k nemu druhým smerom trojuholník. Každá takáto trojica bude mať rovnakú náhodnú farbu).



```

procedure trojr(a:integer);
begin
  if a>0 then
  begin
    vlavo(30);
    dopredu(60); vpravo(120);
    dopredu(60); vpravo(120);
    dopredu(60); vpravo(120);
    vpravo(30);
    pero_hore;
    dopredu((sqrt(60*60-30*30))/2);
    pero_dole;
    nechfv ?;
    vypln;
    pero_hore;
    dopredu((sqrt(60*60-30*30))/2);
    pero_dole;
    kruhy(a-1);
  end;
end;
procedure kruhy :a
begin
  if a>0 then
  begin
    pero_hore;
    dopredu(30);
    pero_dole;
    circle(getx,gety,60);
    pero_hore;
    dopredu(30);
    pero_dole;
    troj(a-1);
  end;
end;

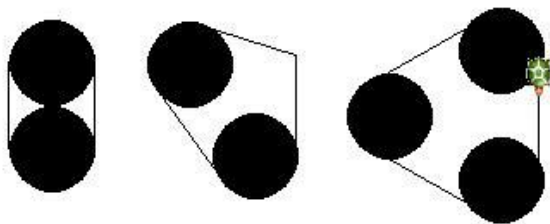
```

```

procedure troj (a:integer);
begin
  if a>0 then
  begin
    vlavo(90);
    dopredu(30); vpravo(120);
    dopredu(60); vpravo(120);
    dopredu(60); vpravo(120);
    dopredu(30); vpravo(90);
    pero_hore;
    dopredu((sqrt(60*60-30*30))/2);
    pero_dole;
    vypln;
    pero_hore;
    dopredu((sqrt(60*60-30*30))/2);
    pero_dole;
    trojr(a-1);
  end;
end;

```

6 Pomocou nepriamej rekurzie nakreslite obrázok podľa predlohy do zadaného stupňa (nakreslí sa n-uholník tak, že jedna strana je čiara a druhá je kruh).



```

procedure kruhy(i,a,n:integer);
begin
  if n>0 then
  begin
    pero_hore; dopredu(a/2);
    pero_dole; circle(getx,gety,a);
    pero_hore; dopredu(a/2);
    vlavo(360*(i-1)/i);
    stvorec(i,a,n-1);
  end;
end;

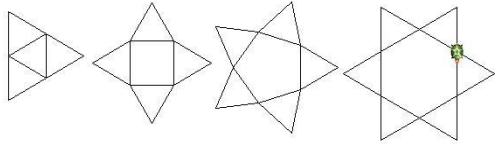
```

```

procedure stvorec(i,a,n:integer);
begin
  if n>0 then
  begin
    pero_dole; dopredu(a);
    vlavo(360*(i-1)/i);
    kruhy(i,a,n-1);
  end;
end;

```

- 7 Pomocou nepriamej rekurzcie nakreslite obrázok podľa predlohy do zadaného stupňa (nakreslí sa n-uholník tak, že namiesto strany bude nakreslený trojuholník. Použite dve procedúry, jedna bude kresliť trojuholník z vonkajšej strany uholníka a druhá bude otáčať korytnačku dopredu správneho smeru).**



```

procedure otoc(i,a,n:integer);
begin
  dopredu(a);
  vlavo(360*(i-1)/i);
  uholnik(i,a,n-1);
end;

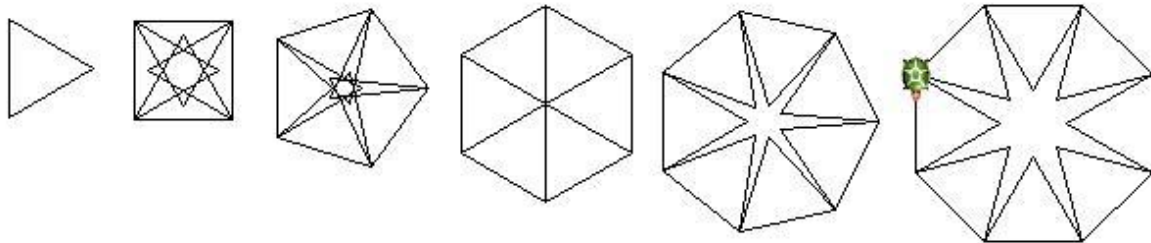
```

```

procedure uholnik(i,a,n:integer);
var i:integer;
begin
  if n >0 then
    begin
      for i:=1 to 3 do
        begin
          dopredu(a);
          vlavo(120);
        end;
      otoc(i,a,n-1);
    end;
end;

```

- 8 Pomocou nepriamej rekurzcie nakreslite obrázok podľa predlohy do zadaného stupňa (nakreslí sa n-uholník tak, že namiesto strany bude nakreslený trojuholník. Použite dve procedúry, jedna bude kresliť trojuholník z vnútornej strany uholníka a druhá bude otáčať korytnačku dopredu správneho smeru).**



```

procedure otoc2 (i,a,n:integer);
begin
  dopredu(a);
  vpravo( 360*(i-1)/i);
  uholnik(i,a,n-1);
end;

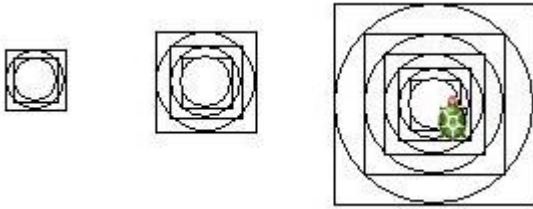
```

```

procedure uholnik(i,a,n:integer);
var i:integer;
begin
  if n >0
  begin
    for i:=1 to 3 do
      begin
        dopredu(a);
        vlavo(120);
      end;
    otoc(i,a,n);
  end;
end;

```

- 9 Napíšte rekurzívny podprogram, ktorý nakreslí zadaný obrázok podľa predlohy tak, aby najmenšia hrana nemala viac ako 20.**



```

procedure kruhl(a:real,n:integer);
begin
  if n>0 then
    begin
      pero_hore;
      vlavo(45);
      dopredu(a*sqrt(2))/2;
      pero_dole;
      ellipse(getx,gety,a,a);
      pero_hore ;
      dozadu(a/2);
      vpravo(45);
      pero_dole;
      stvorec(a*(sqrt 2))/2,n-1);
    end;
  end;
end;

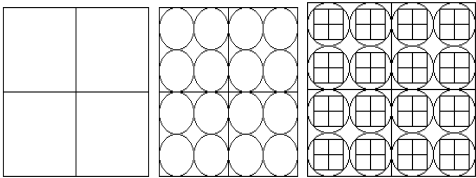
```

```

procedure
stvorec(a:real,n:integer);
var i:integer;
begin
  if n>0 then
    begin
      for i:=1 to 4 do
        begin
          dopredu(a);
          vlavo(90);
        end;
        kruhl(a,n-1);
      end;
    end;
end;

```

10 Napíšte rekurzívny podprogram, ktorý nakreslí zadaný obrázok podľa predlohy dopredu zadaného stupňa (ak je stupeň 0 tak sa nakreslia 4 štvorce, ak je 1 tak v každom dopredu m štvorci 4 kruhy...).



```

procedure kruh4(a:real,n:integer);
begin
  if n>0 then
    begin
      vlavo(135);
      pero_hore;
      dopredu(a/sqrt(2))/2;
      pero_dole;
      ellipse(getx,gety,a/2,a/2);
      vlavo(45);
      stvorec4( a*sqrt(2))/8,n-1);
      vpravo(45);
      pero_hore;
      dozadu(a/sqrt(2))/2);
      vpravo(135);
    end;
  end;
end;

```

```

procedure
stvorec4 (a:real,n:integer);
var j,i:intger;
begin
  if n>0 then
    begin
      for i:=1 to 4 do
        begin
          for j:=1 to 4 do
            begin
              pero_dole;
              dopredu(a);
              kruh4(a,n-1);
              vlavo(90);
            end;
          end;
        end;
      end;
    end;
end;

```

REKURZIVNE FUNKCIE

1 Napíšte rekurzívnu funkciu, ktorá vypíše súčin čísel od 1 po n (faktoriál čísla n).

```

function sucin(n:integer):integer;
begin
  if n>1 then sucin:=n*sucin(n-1)

```

```
    else then sucin:=1;
end;
```

volanie funkcie: `sucin(10);`

nerekurzívny prepis

```
function faktorial(n:integer):integer;
var i, suc:integer;
begin
    suc:=1;
    for i:=2 to n dopredu    suc:=suc*i;
    faktorial:=suc;
end;
```

2 Napíšte rekurzívnu funkciu, ktorá vypíše súčet čísel od 1 po n.

```
function sucet(n:integer)integer;
begin
    if n>0 then sucet:=n+ sucet(n-1)
    else then sucet:=0;
end;
```

volanie funkcie: `sucet(10);`

nerekurzívny prepis

```
function sucet(n:integer):integer;
var i,suc:integer;
begin
    suc:=0;
    for i:=1 to n dopredu    suc:=suc+i;
    sucet:=suc;
end;
```

3 Napíšte rekurzívnu funkciu, ktorá vypočíta n-tý člen Fibonacciho postupnosti.

```
function fibo(n:integer):integer;
begin
    if n>1 then fibo:=fibo(n-1)+ fibo(n-2)
    else fibo:=1
end;
```

volanie funkcie: `fibo(10);`

nerekurzívny prepis

```
function fib(a:integer):integer;
var i,x,y,z:integer;
begin
    x:=1;
    y:=1;
    z:=1;
    for i:=3 to a dopredu
    begin
        z:=x+y;
        x:=y;
        y:=z;
    end;
    fib:=z;
end;
```

4 Napište rekurzívnu funkciu, ktorá vypočíta hodnotu McCarthyho 91 funkcie pri zadaní čísla n, ak procedure e, že je daná predpisom:

$$M(n) = \begin{cases} M(M(n+11)) & \text{pre } n \leq 100 \\ n-10 & \text{pre } n > 100 \end{cases}$$

```
function mccarthy (n:integer):integer;
begin
  if n<=100 then mccarthy:=mccarthy(mccarthy(n+11))
  else mccarthy:=n-10;
end;
```

volanie funkcie: mccarthy(501);

5 Napište rekurzívnu funkciu, ktorá vypočíta hodnotu F(x, y) funkcie pri zadaní čísla x a y, ak procedure e, že je daná predpisom:

$$F(x, y) = \begin{cases} F(x-1, F(x, y-1)) & \\ F(0, y) = y+1 & \\ F(1, 0) = 2 & \\ F(2, 0) = 0 & \\ F(x, 0) = 1 \text{ pre } x > 2 & \end{cases}$$

```
function f(x,y:integer):longint;
begin
  if (x=0) and (y<>0) then f:=y+1
  else if (y=0) and (x=1) then f:=2
  else if (y=0) and (x=2) then f:=0
  else if (y=0) and (x>2) then f:=1
  else f:=f(x-1,f(x,y-1));
end;
```

volanie funkcie: f(10,3);

6 Napište rekurzívnu funkciu, ktorá vypočíta hodnotu Ackermanovej funkcie pri zadaní čísla x a y, ak procedure e, že je daná predpisom:

$$A(x, y) = \begin{cases} y+1 & \text{ak } x=0 \\ A(x-1, 1) & \text{ak } y=0 \\ A(x-1, A(x, y-1)) & \text{inak} \end{cases}$$

```
function a(x,y:integer):longint;
begin
  if x=0 then a:=y+1
  else if y=0 then a:=a(x-1,1)
  else a:=a(x-1,a(x,y-1));
end;
```

komentár: maximálne hodnoty, pre ktoré zbehne výpočet na počítači sú A(4,0),A(3,7).

volanie funkcie: a(2,2);

7 Napište rekurzívnu funkciu, ktorá vypočíta hodnotu Legendrovho polynómu

$$P_n(z) = \begin{cases} 1 & \text{ak } n=0 \\ z & \text{ak } n=1 \\ ((2n-1)zP_{n-1}(z) - (n-1)P_{n-2}(z))/2 & \end{cases}$$

```
function legen(n,z:integer):real;
begin
  if n=0 then legen:=1
  else if n=1 then legen:=z
  else
    legen:=(((2*n-1)*z*legen(n-1,z))-((n-1)*legen(n-2,z)))/2
  end
end
```

volanie funkcie: legen(10,-2);

nerekurzívny prepis

```
function legen(n,z:integer):real;
var i,x,y,a:integer;
begin
  if n=0 then legen:=1
  else if n=1 then legen:=z
  else
    begin
      x:=1; {n=0}
      y:=z; {n=1}
      a:=0; {dalsi clen}
      for i:=2 to a dopredu
      begin
        a:=((2*i-1)*z*y-(i-1)*x)/2;
        x:=y;
        y:=a;
      end;
      legen:=a;
    end;
  end;
end;
```

8 Napíšte rekurzívnu funkcie na výpočet ciferného súčtu čísla A.

```
function cifsucet(n:longint):integer;
begin
  if n>=10 then cifsucet:=( n mod 10)+ cifsucet ( n div 10);
  else cifsucet:=n;
end;
```

9 Napíšte rekurzívnu funkcie na výpočet najväčšieho spoločného deliteľa prirodzených čísel A a B.

```
function nsd(a,b:integer):integer;
begin
  if a>b then nsd:=nsd(a-b,b)
  else if b>a then nsd:= nsd(a,b-a)
  else nsd:=a;
end;
```

volanie funkcie: nsd(6,1156);

nerekurzívny prepis

```
function nsd(a,b:integer):integer;
var k,i,j:integer;
deli:boolean;
```

```

begin
  deli:=false;
  i:=a;
  j:=b;
  if i>j then k:=j
  else k:=i;
  while not deli dopredu
  begin
    if ((i mod k=0) and (j mod k=0))then deli:=true
    else k:=k-1;
  end;
  nsd:=k
end;

```

alebo prepis Euklidopredu vho algoritmu pomocou cyklu

```

function nsd(a,b:integer):integer;
var i,j:integer;
begin
  i:=a;
  j:=b;
  while a<>b dopredu
  begin
    if a>b then a:=a-b
    else if b>a then b:=b-a
  end
  nsd:=a
end;

```

10 Napíšte rekurzívnu funkcie na výpočet najmenšieho spoločného násobku prirodzených čísel A a B (pomocou nájdenia NSD).

```

function nsd(a,b,c:integer):integer;
begin
  if a>b then nsd:=nsd(a-b,b,c)
  else if b>a then nsd:= nsd(a,b-a,c)
  else nsd:=c/a;
end;

```

volanie funkcie: nsd(50,30,50*30);

11 Napíšte rekurzívnu funkcie na výpočet najmenšieho spoločného násobku prirodzených čísel A a B (bez nájdenia NSD).

```

function nsn(a,b,c:integer):integer;
begin
  if c=0 then nsn:=nsn(a,b,c+a)
  else if (c div b<>0) then nsn:=nsn(a,b,c+a)
  else nsn:=c
end;

```

volanie funkcie: nsn(10,236,0);

12 Napíšte rekurzívnu funkciu na výpočet súčiny prirodzených čísel A a B pomocou sčítania.

```

function sucinAB(a,b:integer):integer;
begin
  if a>1 then sucinAB:=b+sucinAB(a-1,b)
  else sucinAB:=b;
end;

```

end;

13 Napíšte rekurzívnu funkciu na výpočet rozdielu kladných čísel A a B bez odčítania tých dvoch čísel (postupne sa odčítava 1 pokiaľ je B>0).

```
function rozdielAB(a,b:integer):integer;
begin
  if b>0 then rozdielAB:=rozdielAB(a-1,b-1)
  else rozdielAB:=a;
end;
```

14 Napíšte rekurzívnu funkciu na výpočet súčtu dvoch prirodzených čísel pomocou jednej premennej v parametri funkcie.

```
function sucetAB(a:integer):integer;
begin
  if a>0 then sucetAB:=1+sucetAB(a-1)
  else
    begin
      readln(a);
      sucetAB:=a;
    end;
end;
```

15 Napíšte rekurzívnu funkciu na výpočet determinantu matice typu MxM (pomocou rozvoja determinantu – Cramerovo pravidlo).

```
function determinant (n:integer;mat:matica):integer;
var pomocna : matica;
  riadopredu k,j,k:integer;
  s:integer;
begin
  s:=0;
  if n=2 then determinant:= matbegin1,1end;*matbegin2,2end;-
  matbegin1,2end;*matbegin2,1end;
  else
    begin
      for riadopredu k:=1 to n dopredu
      begin
        for j:=1 to riadopredu k-1 dopredu
          for k:=2 to n dopredu pomocnabeginj,k-1end;:=matbeginj,kend;;

          for j:=riadopredu k+1 to n dopredu
            for k:=2 to n dopredu pomocnabeginj-1,k-1end;:=matbeginj,kend;;

          if ((riadopredu k+1) mod 2)=0
            then s:=s+matbeginriadopredu k,1end;*determinant(n-1,pomocna)
            else s:=s-matbeginriadopredu k,1end;*determinant(n-1,pomocna);
          end;
        end;
      end;
      determinant:=s;
    end;
end;
```

volanie funkcie: $\text{matica}(4, \begin{pmatrix} 3 & 0 & 4 & -1 \\ 5 & 1 & -9 & 15 \\ 6 & 5 & 5 & 8 \\ 7 & 9 & -3 & 2 \end{pmatrix})$;

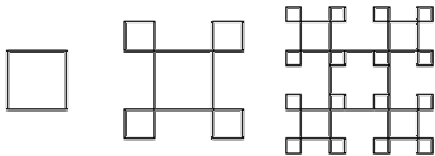
16 Napíšte rekurzívnu funkciu na prevod arabského zápisu čísla na rímsky.

```
function rim(i:integer):string;
begin
  if i>1000 then rim:='M'+rim(i-1000)
  else if i>=900 then rim:='CM'+rim(i-900)
  else if i>=500 then rim:='D'+rim(i-500)
  else if i>=400 then rim:='CD'+rim(i-400)
  else if i>=100 then rim:='C'+rim(i-100)
  else if i>=90 then rim:='XC'+rim(i-90)
  else if i>=50 then rim:='L'+rim(i-50)
  else if i>=40 then rim:='XL'+rim(i-40)
  else if i>=10 then rim:='X'+rim(i-10)
  else if i=9 then rim:='IX'+rim(i-9)
  else if i>=5 then rim:='V'+rim(i-5)
  else if i=4 then rim:='IV'+rim(i-4)
  else if i>0 then rim:='I'+rim(i-1)
  else rim:='';
end;
```

volanie funkcie: rim(1596);

FRAKTÁLY

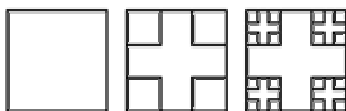
- 1 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.



```
procedure stvorec (a:real,stupen:integer);
var i:integer;
begin
  if stupen>0
  begin
    for i:=1 to 4 do
    begin
      dopredu(a);
      vpravo(90);
      stvorec(a/2,stupen-1);
      vlavo(90);
      vlavo(90);
    end;
  end;
end;
```

volanie procedúry: stvorec 50 3

- 2 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.



```
procedure stvorice (a,stupen:integer);
var i:integer;
begin
```

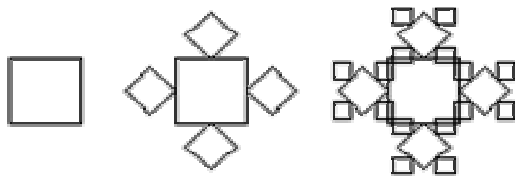
```

if stupen>0
begin
  for i:=1 to 4 do
  begin
    dopredu(a);
    vlavo(90);
    stvorce(a*2/5,stupen-1);
  end;
end;
end;

```

volanie procedúry: stvorce 50 3

3 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.

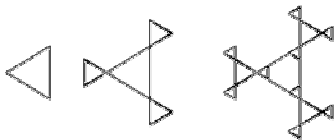


```

procedure stvorec1 (a,stupen:integer);
var i:integer;
begin
  if stupen>0
  begin
    for i:=1 to 4 do
    begin
      dopredu(a/2);
      vpravo(135);
      stvorec1(a/2,stupen-1);
      vlavo(135);
      dopredu(a/2);
      vlavo(90);
    end;
  end;
end;

```

4 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.



```

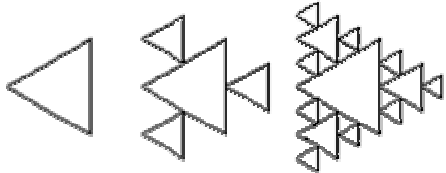
procedure troj (a,stupen:integer);
var i:integer;
begin
  if stupen>0
  begin
    for i:=1 to 3 do
    begin
      dopredu(a);
      vpravo(60);
      troj(a/2,stupen-1);
      vlavo(60);
      vlavo(120);
    end;
  end;
end;

```

```
end;  
end;
```

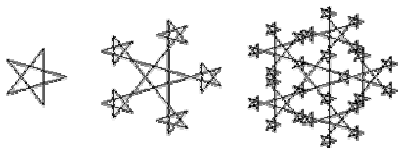
volanie procedúry: troj 50 3

- 5 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.**



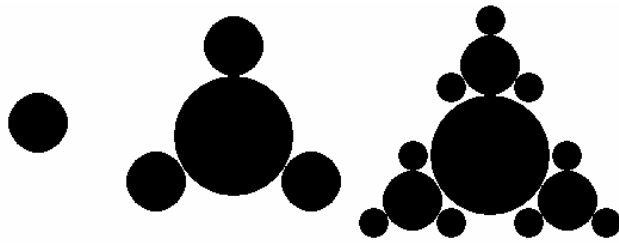
```
procedure trojuholnik (a,stupen:integer);  
var i:integer;  
begin  
  if stupen >0  
  begin  
    for i:=1 to 3 do  
    begin  
      dopredu(a/2);  
      vpravo(120);  
      trojuholnik(a/2,stupen-1);  
      vlavo(120);  
      dopredu(a/2);  
      vlavo(120);  
    end;  
  end;  
end;
```

- 6 Napíšte rekurzívnu procedúru, pomocou ktorej vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.**



```
procedure uholnik (a,stupen:integer);  
begin  
  if stupen>0  
  begin  
    for i:=1 to 5 do  
    begin  
      dopredu(a);  
      vpravo(40);  
      uholnik(a/2,stupen-1);  
      vlavo(40);  
      vlavo(360*2/5);  
    end;  
  end;  
end;
```

- 7 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.**



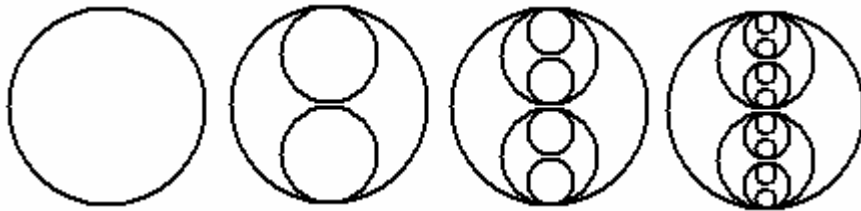
```

procedure bodky(a:real,n:integer);
var i:integer;
begin
  if n=0
  begin
    circle(getx,gety,a)
    break;
  end;

  circle(getx,gety,a);
  for i:=1 to 3 do
  begin
    dopredu(a/2+a/4);
    bodky(a/2,n-1);
    dozadu(a/2+a/4);
    vlavo(120);
  end;
end;

```

8 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.



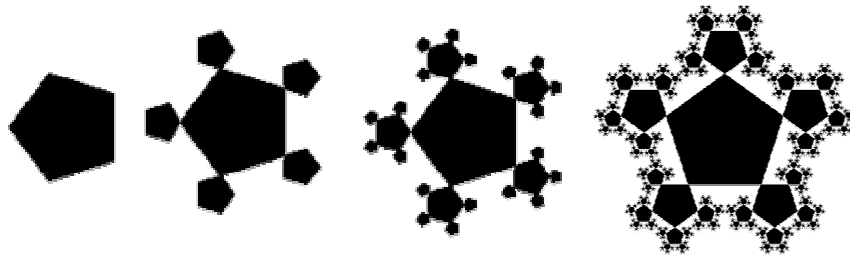
```

procedure kruhy (a:real,n:integer);
begin
  if n=0
  begin
    circle(getx,gety, a);
    break;
  end;
  pero_hore;
  circle(getx,gety, a);

  dopredu(a/4);
  kruhy (a/2,n-1);
  dozadu(a/2);
  kruhy (a/2,n-1);
  dopredu(a/4);
end;

```

9 Napíšte rekurzívnu procedúru, pomocou ktorého sa ktorej "Penta flake distance" podľa predlohy na obrázku, dopredu zvoleného stupňa.

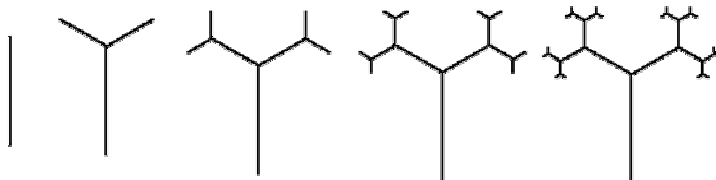


```

procedure pental (prve,d:real,n:integer);
var i:integer;
begin
  if n>0 then
  begin
    if prve=1
    begin
      for i:=1 to 5 do
      begin
        dopredu(d);
        vpravo(108);
        pental(0,d/3,n-1);
        vlavo(108);
        vlavo(72);
      end;
    end;
  else begin
    for i:=1 to 4 do
    begin
      dopredu(d);
      vpravo(108);
      pental(0,d/3,n-1);
      vlavo(108);
      vlavo(72);
    end;
    dopredu(d);
    vlavo(72);
  end;
  vlavo(52);
  pero_hore;
  dopredu(d/2);
  vypln
  dozadu(d/2);
  pero_dole;
  vpravo(52);
end;
end;

```

10 Napište rekurzivnu procedúru, ktorá nakresli binárny strom podľa predlohy na obrázku, dopredu zvoleného stupňa. Vetvy stromu sa skracojú na polovicu vo vyšších úrovniach. Ako parameter sa zadá dĺžka najdlhšej strany a stupeň vykreslenia.



```

procedure strom (a,stupen:integer);
begin
  if :stupen>=0
  begin
    dopredu(a);

```

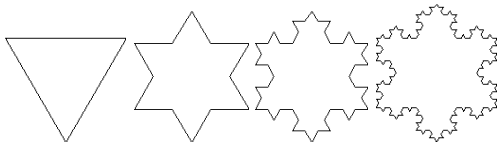
```

    vlavo(60);
    strom (a/2,stupen-1);
    vpravo(120);
    strom (a/2,stupen-1);
    vlavo(60);
    dozadu(a);
end;
end;

```

11 Napíšte rekurzívnu procedúru na kreslenie "Kochovej vložky" (r.1904) podľa predlohy na obrázku, dopredu zvoleného stupňa.

(krivka je daná vzťahom : iniciátor: F++F++F, generátor: F->F-F++F-F, uhol= 60°).



```

procedure
  vlockal(n:integer;s:real);
var i:integer;
begin
  for i:=1 to 3 do
    begin
      vlocka(n,s);
      vlavo(120);
    end;
  end;
end;

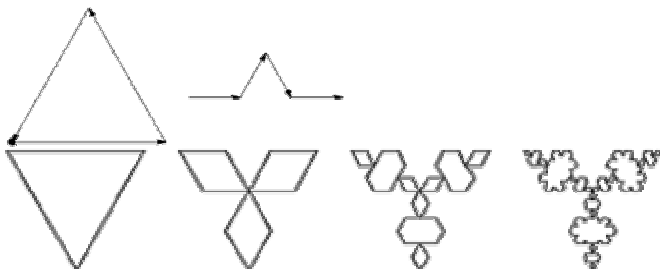
```

```

procedure vlocka (n:integer;s:real);
begin
  if n=0
  begin
    dopredu(s);
  end;
  else
  begin
    procedure vlocka
      (n:integer;s:real);
    begin
      if n=0
      begin
        dopredu(s);
      end;
      else
      begin
        vpravo(60);
        vlocka(n-1,s/3);
        vlavo(120);
        vlocka(n-1,s/3);
        vpravo(60);
        vlocka(n-1,s/3);
      end;
    end;
  end;
end;

```

12 Napíšte rekurzívnu procedúru, pomocou ktorého sa vykreslí "Kochova antivložku" podľa predlohy na obrázku, dopredu zvoleného stupňa.



```

procedure
  vlockal(n:integer;s:real);
var i:integer;
begin

```

```

  procedure vlocka (n:integer;s:real);
  begin
    if n=0
    begin

```

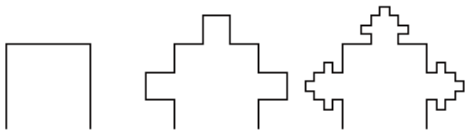
```

for i:=1 to 3 do
begin
  vlocka(n,s);
  vlavo(120);
end;
end;

dopredu(s);
end;
else
begin
  vlocka(n-1,s/3);
  vlavo(60);
  vlocka(n-1,s/3);
  vpravo(120);
  vlocka(n-1,s/3);
  vlavo(60);
  vlocka(n-1,s/3);
end;
end;

```

13 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.

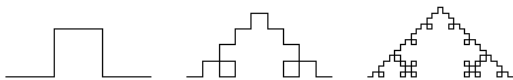


```

procedure krivka (a:integer;b:real);
var i:integer;
begin
  for i:=1 to 3 do
  begin
    if a=0
    begin
      dopredu(b/2);
    end;
    else
    begin
      dopredu(b/2 );
      vlavo(90);
      krivka(a-1,b/2);
      vpravo(180);
      dopredu(b/2);
    end;
    vpravo(90);
  end;
end;

```

14 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí krivka podľa predlohy na obrázku, dopredu zvoleného stupňa.



```

procedure klobuk (n:integer;a:real);
begin
  if :n=0 begin dopredu(a); end;
  else
  begin
    klobuk (n-1,a/2);
    vlavo(90);
    klobuk (n-1,a/2);
    vpravo(90);
    klobuk (n-1,a/2);
    vpravo(90);
    klobuk (n-1,a/2);
    vlavo(90);
  end;
end;

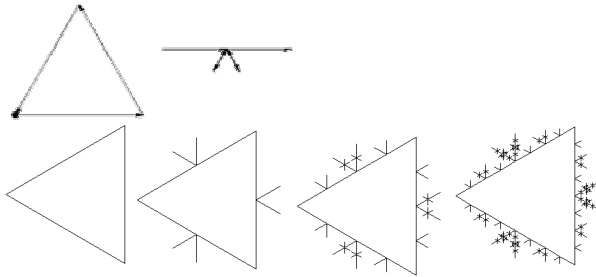
```

```

    klobuk (n-1,a/2);
  end;
end;

```

15 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí "Ice triangel" podľa predlohy na obrázku, dopredu zvoleného stupňa.



```

procedure
  Ictroj (a:integer;b:real);
var i:integer;
begin
  for i:=1 to 3 do
  begin
    krivka(a,b);
    vlavo(120);
  end;
end;

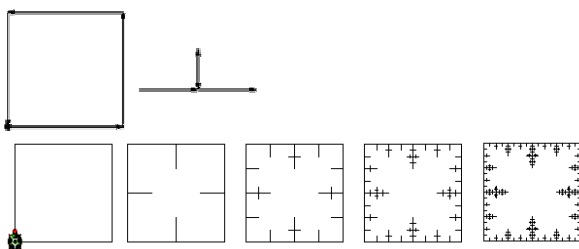
```

```

procedure krivka (a:integer;b:real);
begin
  if a=0 begin dopredu(b); end;
  else
  begin
    krivka (a-1,b/2);
    vpravo(120);
    krivka (a-1,b/5);
    vlavo(180);
    krivka (a-1,b/5);
    vpravo(120);
    krivka (a-1,b/5);
    vlavo(180);
    krivka (a-1,b/5);
    vpravo(120);
    krivka (a-1,b/2);
  end;
end;

```

16 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí "Antilce square" podľa predlohy na obrázku, dopredu zvoleného stupňa.



```

procedure
  Ictroj (a:integer;b:real);
var i:integer;
begin
  for i:=1 to 4 do
  begin
    krivka(a,b);
    vlavo(90);
  end;
end;

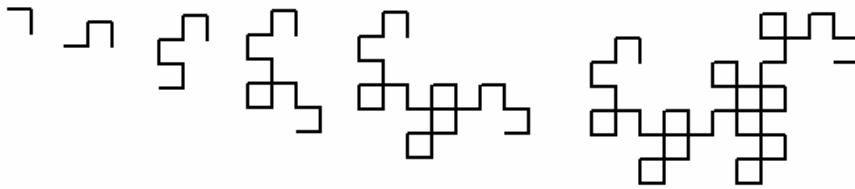
```

```

procedure (a:integer;b:real);
begin
  if a=0 begin dopredu(b); end;
  begin
    krivka (a-1,b/2);
    vpravo(90);
    krivka (a-1,b/4);
    vlavo(180);
    krivka (a-1,b/4);
    vpravo(90);
    krivka (a-1,b/2);
  end;
end;

```

17 Napíšte rekurzívnu procedúru, pomocou ktorej sa vykreslí "Dračiu krivku" podľa predlohy na obrázku, dopredu zvoleného stupňa.



```

procedure pravy (d:real,n:integer);
begin
  if n=0
  begin
    dopredu(d);
    break;
  end;
  lavy(d,n-1);
  vpravo(90);
  pravy(d,n-1);
end;

procedure lavy (d:real,n:integer);
begin
  if n=0
  begin
    dopredu(d);
    break;
  end;
  lavy(d,n-1);
  vlavo(90);
  pravy(d,n-1);
end;

```

18 Napíšte rekurzívnu procedúru, pomocou ktorých sa vykreslí "Cantorovu množinu" podľa predlohy na obrázku, dopredu zvoleného stupňa.

(krivka je daná vzťahom: $X \rightarrow XYX$, $Y \rightarrow YYY$)



```

procedure
(d:real,n:integer);
  ak :n=0   begindopredu
  ukonciend;
  cantorA :d/3 :n-1
  cantorB :d/3 :n-1
  cantorA :d/3 :n-1
end;

cantorA procedure
cantorB (d:real,n:integer);
:d begin
  if n=0
  begin
    pero_hore
    dopredu(d);
    pero_dole;
    break;
  end;
  cantorB(d/3,n-1);
  cantorB(d/3,n-1);
  cantorB(d/3,n-1);
end;

```

alebo

```

procedure
cantorA (d:real,n:integer);
begin
  if n=0
  begin
    dopredu(d)
    break;
  end;
  cantorA(d/3,n-1);
  cantorB(d/3,n-1);
  cantorA(d/3,n-1);
end;

procedure
cantorB (d:real,n:integer);
begin
  pero_hore
  dopredu(d);
  pero_dole
  break;
end;
cantorB(d,n-1);
end;

```