

11. Rekurzívne číselné funkcie

Všetky tieto rekurzívne funkcie by sa dali zapísať aj bez rekurzíe, napr. pomocou cyklov - našou úlohou je nacvičiť sa riešiť tieto úlohy rekurzívne, hoci mnohokrát sú to veľmi neefektívne riešenia

<i>k-ta mocnina čísla n</i>	
triviálny prípad: $n^0 = 1$	
inak: $n^k = n * n^{k-1}$	
<i>fibonacciho postupnosť</i>	
triviálny prípad: $f(0) = f(1) = 1$	<pre>f1:=1; f2:=1; while n>1 do begin i:=f1; f1:=f2; f2:=i+f2; dec(n); end; Result:=f2;</pre>
inak: $f(n) = f(n-1) + f(n-2)$	
úlohu riešte takýmto postupom: - najprv zapíšte pomocou while-cyklu - prepíšte while do chvostovej rekurzíe (opačným postupom, ako sa chvostová rekurzia prepíše do while-cyklu)	<pre>function fib(n:integer):integer; function fib1(n,f1,f2:integer); begin ... end; begin ... end;</pre>
<i>kombinačné číslo</i>	
pripomeňme si n nad k :	<pre>function C(n,k:integer):integer; begin ... end;</pre>
triviálny prípad: $C(n,0) = C(n,n) = 1$	
inak: $C(n,k) = C(n-1,k-1) + C(n-1,k)$	

Ďalšie úlohy

- rekurzívne násobenie dvoch celých čísel pomocou súčtu
- podobne sa dá optimalizovať aj súčet
- rekurzívne (euklidovým algoritmom) vypočítajte NSD - najväčší spoločný deliteľ

rekurzia a znakové reťazce

s reťazcami sa veľmi zriedka pracuje pomocou rekurzíe - väčšinou v praxi sa používajú nerekurzívne algoritmy (pomocou cyklov)

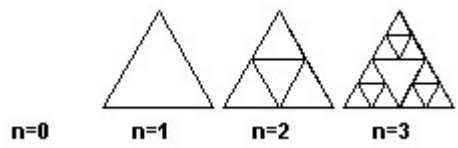
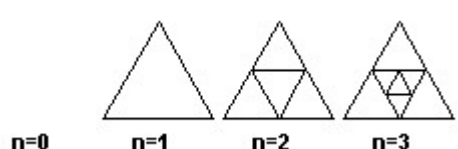
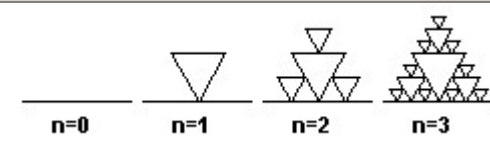
<i>palindrom</i>	
<p>rekurzívna funkcia zistí, či je reťazec palindrom.</p> <p>Triviálny prípad: prázdny alebo jednoznakový reťazec.</p> <p>Inak: prvý znak sa rovná poslednému a zvyšok je tiež palindrom</p>	<pre>function palindrom(s:string):boolean; begin ... end;</pre>
<i>prevrátenie reťazca</i>	
<p>rekurzívna funkcia prevráti reťazec</p> <p>triviálny prípad: prázdny alebo jednoznakový reťazec</p> <p>inak: prevrátenie reťazca bez prvého a posledného znaku a k tomu sa pridá posledný (na začiatok) a prvý znak (na koniec)</p>	<pre>function prevrat(s:string):string; begin ... end;</pre>
<i>prevrátenie poradia riadkov v súbore</i>	
<p>triviálny prípad: súbor je prázdny</p> <p>inak: prečítame 1. riadok, prevrátime zvyšok súboru (rekurzívne), zapíšeme zapamätaný prvý riadok (na koniec)</p> <p>premyslite, čo by sa stalo, keby lokálna premenná s nebola lokálnou v prevrat1, ale o úroveň vyššie - bola by lokálnou v prevratSubor</p>	<pre>procedure prevratSubor(meno:string); var t1,t2:TextFile; procedure prevrat1; begin ... end; begin ... end;</pre>
<i>prevrátenie obsahu súboru 1.verzia</i>	
<p>v predchádzajúcom príklade budeme meniť len vnorenú procedúru prevrat1</p> <p>1. verzia riešenia je veľmi podobná prevracaniu poradia riadkov - len sa číta a zapisuje po jednom znaku - problém vzniká s dvojicou znakov #13#10</p>	<pre>procedure prevrat1; var z:char; begin ... end;</pre>
<i>prevrátenie obsahu súboru 2.verzia</i>	
<p>2. verzia využíva reťazcovú funkciu prevrat, ktorá prevracia znakový reťazec</p>	<pre>procedure prevrat1; var s:string; begin ... end;</pre>

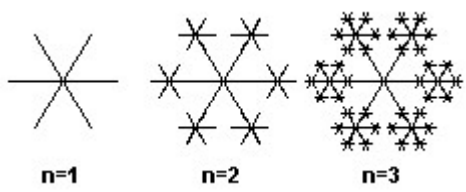
Ďalšie úlohy

- napíšte rekurzívnu funkciu, ktorá vygeneruje znakový reťazec 2^n hviezdíčiek, napr. $hv(3)='*****'$;
- napíšte rekurzívnu funkciu, ktorá pre dané n vráti znakový reťazec zložený z toľkých hviezdíčiek, ako je n -té fibonacciho číslo - vo funkcii nepoužívajte iné číselné premenné okrem formálneho parametra n

rekurzívne krivky

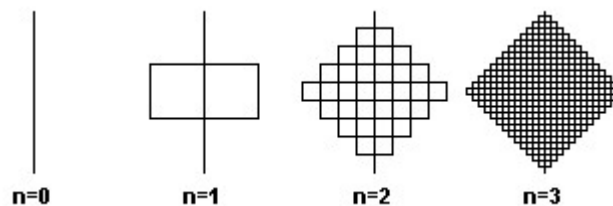
- všetky krivky kreslíme pomocou korytnačky, ktorá je zadefinovaná v globálnej premennej k a je vytvorená niekde v strede plochy
- pri riešení úloh sa snažte v útvere úrovne n nájsť útvar úrovne $n-1$ a tiež postup, ako to využijete

trojuholníky	
<p>triviálny prípad: pre $n=0$ je nič</p>  <p>$n=0$ $n=1$ $n=2$ $n=3$</p>	<pre> procedure trojuhol1(n:integer; s:real); var i:integer; begin if n=0 then // nič else for i:=1 to 3 do begin ... end; end; </pre>
vpísané trojuholníky	
<p>triviálny prípad: pre $n=0$ je nič</p>  <p>$n=0$ $n=1$ $n=2$ $n=3$</p>	<pre> procedure trojuhol2(n:integer; s:real); begin ... end; </pre>
trojuholníková pyramída	
 <p>$n=0$ $n=1$ $n=2$ $n=3$</p> <p>uvedomte si, že tento obrázok vznikol jedným ťahom (ako by ste mohli využiť tento algoritmus na nakreslenie rekurzívneho obrázka trojuhol1 jedným ťahom?)</p>	<pre> procedure trojuhol3(n:integer; s:real); var i:integer; begin ... end; </pre>

<i>trojuhelníková pyramída</i>	
rovnaký útvar trochu iným postupom	<pre> procedure trojuhol3(n:integer; s:real); begin ... end; </pre>
<i>snehová vločka</i>	
triviálny prípad: pre $n=0$ je nič	<pre> procedure vlocka(n:integer; s:real); var i:integer; begin if n=0 then // nič else ... end; </pre>
	

Ďalšie úlohy

- pokúste sa niektorú z týchto úloh vyriešiť bez korytnačky - len štandardnými príkazmi grafickej plochy (MoveTo a LineTo)
- nájdite postup pre



- nasledujúci rekurzívny obrázok vznikol rovnakým postupom, len namiesto zatočenia vľavo, resp. vpravo o 90 stupňov, sa najprv otočí o 45, potom urobí malý krok a potom opäť o 45:

