Úlohy ke zkoušce z předmětu Programování pro fyziky (OFY056)

30. 1. 2007

1. Nachystejte Pascalův trojúhelník tak, aby následné volání funkce `Binom(n,k)` vracelo binomické číslo s konstantní časovou složitostí. Když už budete mít Pascalův trojúhelník po ruce, připravte i funkci realizující součet $F_{n+1} = \sum_{k=0}^{m}$ (n–k nad k), kde m je celá část n/2, a nazvěte ji `Fibonacci`.
   Víme, že: Pascalův trojúhelník obsahuje na n-tém řádku (n = 0, 1, ...) n+1 binomických čísel (n nad 0), (n nad 1), ..., (n nad n) a k jeho sestrojení se užívají vzorce (n nad 0) = (n nad n) = 1, (n nad k) = = (n–1 nad k–1) + (n–1 nad k), k = 1, 2, ..., n–1.

2. Zjistěte, k čemu slouží následující procedura.

```
procedure P(const x:real;const c:array of real;var a,b:real);
var j:integer;
begin
  a:=c[High(c)];
  b:=0.;
  for j:=High(c)-1 downto Low(c) do begin
    b:=b*x+a;
    a:=a*x+c[j];
  end;
end;
```

3. Mějme pole ponožek neboli strukturovanou proměnnou typu `tArrPonozka`,

```
type
  tPonozka = record
    velikost : tVelikost;
    barva    : tBarva;
    material : tMaterial;
    dira     : boolean;
  end;
  tArrPonozka = array of tPonozka;
```

   Napište funkci

```
function PocetParu(const Ponozky : tArrPonozka) : integer;
```

   která spočítá, kolik párů stejných ponožek (párů ponožek se stejnými položkami `velikost`, `barva` a `material`) je v poli k dispozici. Levé a pravé ponožky se nerozlišují, děravé ponožky se nezapočítávají, typy `tVelikost`, `tBarva` a `tMaterial` jsou ordinální. Aplikujte algoritmus o lineární časové složitosti, tedy cyklus

```
for i:=Low(Ponozky) to High(Ponozky) do
```

   použijte právě jednou. (Stejně jako ponožky – použít jednou a dost.)

4. a) Newtonův interpolační polynom n-tého stupně, procházející n+1 interpolačními uzly (x_p, y_p), p = 0, 1, ..., n, se zapisuje ve tvaru

   (*) $N_n(z) = a_0 + a_1 (z–x_0) + a_2 (z–x_0)(z–x_1) + ... + a_n (z–x_0)(z–x_1)...(z–x_{n–1})$.

   Napište funkci

```
function Newton(const z: real; const a,x: array of real) : real;
```

   která pro dané z vypočte hodnotu Newtonova interpolačního polynomu $N_n(z)$, určeného n+1 koeficienty a_p, p = 0, 1, ..., n, a prvními n souřadnicemi x_p interpolačních uzlů.

   b) Dosadíte-li do (*) (n+1)krát interpolační podmínky, $N_n(x_p) = y_p$, získáte soustavu n+1 lineárních algebraických rovnic pro koeficienty a_p. Matice M_{pq} této soustavy je dolní trojúhelníková, a soustavu tak lze řešit postupným vyčíslováním $a_p = (1/M_{pp}) (y_p – \sum_{q=0}^{p–1} M_{pq} a_q)$, p = 0, 1,..., n. Zpracujte proceduru

```
procedure EvalNewton(const x,y: array of real; var a: array of real);
```

   pro výpočet koeficientů a_p z daných souřadnic interpolačních uzlů.