Rešeni zadaci iz Uvoda u programiranje

Nenad Radu

1st Edition

Contents

1 Sume i Proizvodi

Sume i proizvodi su svima vama sigurno već vrlo dobro poznati. Ako pogledate prvi zadatak sa bilo kog ispita u poslednjih desetak godina, videćete da se traži rešenje funkcije koja je manje i ili više slična sledećoj:

$$F(x,y,n) = \sum_{k=1}^{n} \frac{k!!x^{k^2} \prod_{j=1}^{k} \frac{y^{2j-1}+k}{\sum_{j=1}^{j} 2^{k+i}x^{3i}}}{1+\sum_{j=1}^{k} \frac{y^j}{(k!)^2}}$$

Na prvi pogled izgleda glomazno i užasno, ali kako budete prelazili ovu knjigu, videćete da će bivati sve jasnije i jasnije...

1.1 Rekurzija

Problemi rešavanja suma i prozvoda su problemi rekurzivne prirode, odnosno koristimo neke prethodne rezultate koji nam mogu pomoći za lakše i brže izračunavanje nekog problema.

Ako recimo treba da izračunamo sumu od n elemenata, najprirodniji način za njeno izračunavanje jeste sledeći: *Na prvi broj dodaćemo drugi, na taj zbir dodaćemo treći, pa na taj zbir četvrti, itd.*

Ako recimo treba da izračunamo proizvod od n elemenata, najprirodniji način za njeno izračunavanje jeste sledeći: *Prvi broj pomnožimo sa drugim, taj proizvod pomnožimo sa trećim, pa taj proizvod sa četvrtim, itd.*

Na isti način potrudićemo se da prilikom računanja naših suma i proizvoda, nemamo bespotrebnih računanja, odnosno da ne računamo ponovo ono što smo već mogli izračunati.

1.1.1 Primer

Ako primer iz prethodnog poglavlja uzmemo i raspisemo za neko konkretno n, recimo n = 2, dobili bi sledeće:

$$F(x,y,2) = \frac{1!!x^{1^2}\frac{y^{2-1}+1}{2^{1+1}x^3}}{1+\frac{y^1}{(1!)^2}} + \frac{2!!x^{2^2}\left(\frac{y^{2-1}+2}{2^{2+1}x^3}\frac{y^{4-1}+k}{2^{2+1}x^3+2^{2+2}x^6}\right)}{1+\frac{y^1}{(2!)^2}+\frac{y^2}{(2!)^2}}$$

Kao što se lako da primetiti u ovoj formuli ima dosta sličnih izraza. Potrudićemo se da imamo što manje računanja kod izraza koji izgledaju ovako slično.

Raspisivanje suma na ovakav način je dobra vežba za razumevanje i shvatanje kako cele ove formule izgledaju i način na koji ćemo ih mi obradjivati. Predlažemo ovaj postupak za što češću vežbu.

1.2 Jednostavne formule

U ovom odeljku ćemo objasniti kako se prave i kako izgledaju rekurente veze i relacije za jednostavnije formule od kojih su gradjene sume i proizvodi.

Rekurentnu vezu pravimo tako što tražimo vezu izmedju dva susedna člana. Kod svakih rekurentnih relacija, moraju postajati i inicijalne ili početne vrednosti.

Primeri koji će biti obradjeni se i najčešće koriste kao podproblemi u rešavanju celog zadatka. Na kraju svakog primera će biti napisan deo programa koji pretstavlja programski zapis matematičkog problema. Ukoliko naidjete na neki primer koji nije obradjen, to znači ili da smo obradili nešto slično ili da ste stekli odredjeni alat sa kojim možete obraditi taj primer.

1.2.1 stepenovanje - x^k

$$stx_k = x^k$$

$$\frac{stx_k}{stx_{k-1}} = \frac{x^k}{x^{k-1}} = x$$

$$stx_k = stx_{k-1} \cdot x$$

$$stx_0 = 1$$

```
stx := stx * x;
```

1.2.2 stepenovanje - x^{ak+b}

$$stx_k = x^{ak+b}$$

$$\frac{stx_k}{stx_{k-1}} = \frac{x^{ak+b}}{x^{ak-a+b}} = x^a$$

$$stx_k = stx_{k-1} \cdot x^a$$
$$stx_0 = x^b$$

```
stx := stx * xnaa;
```

1.2.3 stepenovanje - x^{k^2}

$$stx_k = x^{k^2}$$
$$\frac{stx_k}{stx_{k-1}} = \frac{x^{k^2}}{x^{k^2-2k+1}} = x^{2k-1}$$
$$pstx_k = x^{2k-1}$$
$$stx_k = stx_{k-1} \cdot pstx_k$$
$$stx_0 = 1$$
$$pstx_k = pstx_{k-1} \cdot x^2$$
$$pstx_0 = \frac{1}{x}$$

```
pstx := pstx * xna2;
stx := stx * pstx;
```

1.2.4 stepenovanje - x^{ak^2}

$$stx_k = x^{ak^2}$$
$$\frac{stx_k}{stx_{k-1}} = \frac{x^{ak^2}}{x^{a(k^2-2k+1)}} = x^{a(2k-1)}$$
$$pstx_k = x^{a(2k-1)}$$
$$stx_k = stx_{k-1} \cdot pstx_k$$
$$stx_0 = 1$$
$$pstx_k = pstx_{k-1} \cdot x^{2a}$$
$$pstx_0 = \frac{1}{x^a}$$

```
pstx := pstx * xna2a;
stx := stx * pstx;
```

1.2.5 stepenovanje - x^{n-k}

$$stx_k = x^{n-k}$$

$$\frac{stx_k}{stx_{k-1}} = \frac{x^{n-k}}{x^{n-k+1}} = \frac{1}{x}$$

$$stx_k = stx_{k-1} \cdot \frac{1}{x}$$

$$stx_0 = x^n$$

```
stx := stx / x;
```

1.2.6 Binomni koeficijent - $\binom{n}{k}$

$$bin_k = \binom{n}{k}$$

$$\frac{bin_k}{bin_{k-1}} = \frac{\binom{n}{k}}{\binom{n}{k-1}} = \frac{\frac{n!}{k!(n-k)!}}{\frac{n!}{(k-1)!(n-k+1)!}} = \frac{n+1-k}{k}$$

$$bin_k = bin_{k-1} \cdot \frac{n+1-k}{k}$$

$$bin_0 = 1$$

```
n1 := n + 1;
bin := bin * FLOAT(n1 - k) / FLOAT(k);
```

1.2.7 Binomni koeficijent - $\binom{2k}{k}$

$$bin_k = \binom{2k}{k}$$

$$\frac{bin_k}{bin_{k-1}} = \frac{\binom{2k}{k}}{\binom{2k-2}{k-1}} = \frac{\frac{(2k)!}{k!k!}}{\frac{(2k-2)!}{(k-1)!(k-1)!}} = \frac{(2k)(2k-1)}{k^2} = \frac{4k-2}{k}$$

$$bin_k = bin_{k-1} \cdot \frac{4k-2}{k}$$

$$bin_0 = 1$$

```
bin := bin * FLOAT(4 * k - 2) / FLOAT(k);
```

1.2.8 Binomni koeficijent - $\binom{3k}{k}$

$$bin_k = \binom{3k}{k}$$

$$\frac{bin_k}{bin_{k-1}} = \frac{\binom{3k}{k}}{\binom{3k-3}{k-1}} = \frac{\frac{(3k)!}{k!(2k)!}}{\frac{(3k-3)!}{(k-1)!(2k-2)!}} = \frac{(3k)(3k-1)(3k-2)}{k(2k)(2k-1)} = 3\frac{(3k-1)(3k-2)}{(2k)(2k-1)}$$

$$bin_k = bin_{k-1} \cdot 3 \cdot \frac{(3k-1)(3k-2)}{(2k)(2k-1)}$$

$$bin_0 = 1$$

```
k3 := 3 * k; k2 := 2 * k;
bin := bin * 3.0 * FLOAT((k3 - 1) * (k3 - 2)) \ FLOAT(k2 * (k2 - 1));
```

1.2.9 Faktorijel - $k!$

$$fak_k = k!$$
$$\frac{fak_k}{fak_{k-1}} = \frac{k!}{(k-1)!} = k$$
$$fak_k = fak_{k-1} \cdot k$$
$$fak_0 = 1$$

```
fak := fak * FLOAT(k);
```

1.2.10 Faktorijel - $(2k)!$

$$fak_k = (2k)!$$
$$\frac{fak_k}{fak_{k-1}} = \frac{(2k)!}{(2k-2)!} = (2k)(2k-1)$$
$$fak_k = fak_{k-1} \cdot (2k)(2k-1)$$
$$fak_0 = 1$$

```
k2 := 2 * k;
fak := fak * FLOAT(k2 * (k2 - 1));
```

1.2.11 Faktorijel - $(n-k)!$

$$fak_k = (n-k)!$$
$$\frac{fak_k}{fak_{k-1}} = \frac{(n-k)!}{(n-k+1)!} = \frac{1}{(n+1-k)}$$
$$fak_k = fak_{k-1} \cdot \frac{1}{(n+1-k)}$$

```
n1 := n + 1;
fak := fak / FLOAT(n1 - k);
```

1.2.12 Dupli faktorijel - $k!!$

$$dfak_k = k!!$$
$$pdfak_k = (k-1)!!$$
$$dfak_k = pdfak_{k-1} \cdot k$$
$$pdfak_k = dfak_{k-1}$$
$$dfak_0 = 1$$
$$pdfak_0 = 1$$

```
pom := pdfak;
pdfak := dfak;
dfak := FLOAT(k) * pom;
```

1.2.13 Dupli faktorijel - $(2k)!!$

$$dfak_k = (2k)!!$$
$$\frac{dfak_k}{dfak_{k-1}} = \frac{(2k)!!}{(2k-2)!!} = 2k$$
$$dfak_k = dfak_{k-1} \cdot 2k$$
$$dfak_0 = 1$$

```
dfak := dfak * FLOAT(2 * k);
```

1.2.14 Suma - $\sum A$

$$S_k = \sum A$$
$$S_k = S_{k-1} + A$$
$$S_0 = 0$$

```
S := S + A;
```

1.2.15 Proizvod - $\prod A$

$$P_k = \prod A$$
$$P_k = P_{k-1} \cdot A$$
$$P_0 = 1$$

```
P := P * A;
```

1.3 Korisne transformacije

U zadacima će ponekada biti potrebno da uradimo odredjene ekvivalente transformacije dobijene formule, da bi smo je sveli na nešto što već znamo da rešimo.

Ako su k promenljiva u sumi ili prozivodu od n elemenata, c neka konstanta, a A i B neki izrazi koji zavise od k, onda znamo da važe sledeće jednakosti:

$$\sum cA = c \sum A$$
$$\prod cA = c^n \prod A$$

$$\sum (A + B) = \sum A + \sum B$$
$$\sum (A - B) = \sum A - \sum B$$

$$\prod (AB) = \prod A \prod B$$
$$\prod \frac{A}{B} = \frac{\prod A}{\prod B}$$

Naravno treba primetiti i da se proizvod i količnik dva izraza ne može rastaviti pod sumom, kao ni zbir niti razlika dva izraza pod proizvodom.

1.4 Zadaci

Kada smo objasnili kako se matematički izvode rekurentne veze pojedinih izraza sada možemo da krenemo sa konkretnim rešavanjem nekih ispitnih zadataka.

Zadaci će biti podeljeni u dve grupe: zadaci koji se rešavaju sa jednom FOR petljom i zadaci koji se rešavaju sa dve ili više FOR petlji.

Ako sa brojačem nazovemo vrednosti koje se nalaze u indeksima suma i proizvoda (k, j, i, p), a sa konstantama sve ostale vrednosti (respektivno vrednosti x, y i n su takodje konstante jer su one poznate u toku izračunavanja odredjene funkcije), onda možemo reći i sledeće:

Stav 1. Ako se pod odredjenom sumom ili proizvodom nalaze konstante i od brojača samo onaj koji se nalazi u indeksu uočene sume ili proizvoda, onda se cela ta suma ili proizvod može izračunati preko prethodne sume ili proizvoda.

$$\sum_{k=1}^{n} 2^k k! \prod_{j=1}^{k} \frac{x^j + j}{j!!}$$

Proizvod u našem primeru se može izračunati preko glavne sume, odnosno u istoj petlji u kojoj se i ona računa.

Stav 2. Ako se pod odredjenom sumom ili proizvodom nalaze konstante i bar dva različita brojača (nije bitno koja) onda se ta suma ili proizvod mora računati zasebno od njene prethodne sume ili proizvoda.

$$\sum_{k=1}^{n} \binom{4k}{k} y^{3k-2} \sum_{j=1}^{k} \frac{x^{j^3}(y^i+3)^k}{(2j-1)^2(y+3)^{3j+2}}$$

Suma u našem primer se ne može izračunati preko glavne sume, odnosno potrebna je bar jos jedna nova petlja za njeno izračunavanje.

Svaki od uradjenih zadataka će biti isto obradjen. Prvo će biti ispisano njegovo matematičko izvodjenje, a zatim sam program, kako treba da izgleda, i na kraju objašnjenje zašto je taj primer izabran, koje su novosti i na šta se treba obratiti pažnja.

1.4.1 1. Zadatak - Jun 1998

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n}\sum_{j=1}^{k}\sum_{i=1}^{j}\frac{ijkx^j}{\binom{n}{i}y^{2i-1}\prod_{m=1}^{i}\binom{2m}{m}}$$

$$F(x,y,n) = \sum_{k=1}^{n}k\sum_{j=1}^{k}jx^j\sum_{i=1}^{j}\frac{i}{\binom{n}{i}y^{2i-1}\prod_{m=1}^{i}\binom{2m}{m}}$$

$$S_k = S_{k-1} + k \cdot S1_k$$

$$S_0 = 0$$

$$S1_k = \sum_{j=1}^{k}jx^j\sum_{i=1}^{j}\frac{i}{\binom{n}{i}y^{2i-1}\prod_{m=1}^{i}\binom{2m}{m}}$$

$$S1_k = S1_{k-1} + k \cdot stx \cdot S2_k$$

$$S1_0 = 0$$

$$stx_k = x^k$$

$$stx_k = stx_{k-1} \cdot x$$

$$stx_0 = 1$$

$$S2_k = \sum_{i=1}^{k}\frac{i}{\binom{n}{i}y^{2i-1}\prod_{m=1}^{i}\binom{2m}{m}}$$

$$S2_k = S2_{k-1} + \frac{k}{cin_k \cdot P_k}$$

$$S2_0 = 0$$

$$cin_k = \binom{n}{k}y^{2k-1}$$

$$cin_k = cin_{k-1} \cdot y^2 \cdot \frac{n+1-k}{k}$$

$$cin_0 = \frac{1}{y}$$

$$P_k = \prod_{m=1}^{k} \binom{2m}{m}$$

$$P_k = P_{k-1} \cdot bin_k$$

$$P_0 = 1$$

$$bin_k = \binom{2k}{k}$$

$$bin_k = bin_{k-1} \cdot \frac{4k-2}{k}$$

$$bin_0 = 1$$

Program

```
MODULE Suma0698;
(* Jun 1998.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 0;
    nggr = 50;
    xgr = 3.0 / 4.0;
    ygr = 1.0;

  VAR
    S, S1, S2, stx, P, cin, bin : REAL;
    k, n1 : CARDINAL;
    yna2 : REAL;
  BEGIN
    IF (ndgr < n) AND (n < nggr) AND (ABS(x) < xgr)
                  AND (ABS(y) < ygr) AND (y # 0.0) THEN
      ok := TRUE;
      S := 0.0; S1 := 0.0; S2 := 0.0;
      stx := 1.0; P := 1.0; bin := 1.0; cin := 1.0 / y;
      n1 := n + 1; yna2 := y * y;
      FOR k := 1 TO n DO
        bin := bin * FLOAT(4 * k - 2) / FLOAT(k);
        P := P * bin;
```

```
        cin := cin * yna2 * FLOAT(n1 - k) / FLOAT(k);
        S2 := S2 + FLOAT(k) / (cin * P);
        stx := stx * x;
        S1 := S1 + FLOAT(k) * stx * S2;
        S := S + FLOAT(k) * S1;
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;

END Suma0698.
```

Objašnjenje

Prilikom rešavanja svakog zadatka pre pisanja programa potrebno je uraditi matematičko izvodjenje. Kao što ćete primetiti program se veoma lako i jednostavno konstruiše na osnovu njega.

U ovom primeru potrebno je primetiti značaj transformacija opisanih u prethodnom poglavlju. Na prvi pogled suma je delovala da se ne moze uraditi u jednoj petlji, jer je više brojača bilo medjusobno povezano. Medjutim, primenom transformacija dobijamo jednostavniji problem. Dalje, korišćenjem jednostavnih formula naš problem delimo dok ne ispisemo sve potrebne formule. Naravno, ponekada neke stvari mozemo grupisati pod jednu, kao sto je opisana promenljiva cin_k. Isti rezultat bi dobili da smo posebno rekurzivno resavali binomni deo, pa stepeni deo i ta dva na kraju pomnožili. Ukoliko nekom ovakvo grupisanje predstavlja problem, predlažemo da radi školski, tj. svaki deo zasebno.

Posle izvodjenja napisan je i program. Program se sastoji iz opšteg dela i procedure F. Opšti deo će biti ispisan samo u prvih nekoliko primera jer je potpuno identičan za svaki zadatak. Procedura se sastoji iz dva dela. Prvi deo je inicijalizacija početnih vrednosti. One se dobijaju tako sto se u formulama umesto k ubaci 0. U ovom delu se takodje izračunaju i neke pomoćne vrednosti koje se kasnije često koriste, kao što su $n1$ i $yna2$. Drugi deo je FOR petlja. U njoj je bitno primetiti redosled naredbi, odnosno da se svaka promenljiva računa tek kada se izračunaju sve njene komponente od kojih se ona sastoji. Npr $S2_k$ se računa tek kada se znaju cin_k i P_k, a P_k tek posle bin_k. Praktično, u programu se prvo ispisuje i računa ono što se u matematičkom izvodjenju poslednje izvede.

1.4.2 2. Zadatak - Januar 2000

Matematičko izvodjenje

$$F(x,y,n)=\sum_{k=1}^{n}2^{n-k}\prod_{j=1}^{k}\frac{\binom{2n}{j}x^{2j-1}\sum_{i=1}^{j}\left(y^{3i-2}j!!\binom{n}{i}\right)}{3+\sum_{p=1}^{j}\frac{x^{p^2}y^p}{(3p)!}}$$

$$F(x,y,n)=\sum_{k=1}^{n}2^{n-k}\prod_{j=1}^{k}\frac{\binom{2n}{j}x^{2j-1}j!!\sum_{i=1}^{j}\left(y^{3i-2}\binom{n}{i}\right)}{3+\sum_{p=1}^{j}\frac{x^{p^2}y^p}{(3p)!}}$$

$$S_k=S_{k-1}+st2_k\cdot P_k$$

$$S_0=0$$

$$st2_k=2^{n-k}$$

$$st2_k = st2_{k-1}\frac{1}{2}$$

$$st2_0 = 2^n$$

$$P_k = \prod_{j=1}^{k} \frac{\binom{2n}{j} x^{2j-1} j!! \sum_{i=1}^{j} \left(y^{3i-2} \binom{n}{i} \right)}{3 + \sum_{p=1}^{j} \frac{x^{p^2} y^p}{(3p)!}}$$

$$P_k = P_{k-1} \cdot \frac{bin_k \cdot stx_k \cdot dfak_k \cdot S1_k}{3 + S2_k}$$

$$P_0 = 1$$

$$bin_k = \binom{2n}{k}$$

$$bin_k = bin_{k-1} \cdot \frac{2n+1-k}{k}$$

$$bin_0 = 1$$

$$stx_k = x^{2k-1}$$

$$stx_k = stx_{k-1} \cdot x^2$$

$$stx_0 = \frac{1}{x}$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_k \cdot k$$

$$pdfak_k = dfak_{k-1}$$

$$dfak_0 = 1$$

$$pdfak_0 = 1$$

$$S1_k = \sum_{i=1}^{k} y^{3i-2} \binom{n}{i}$$

$$S1_k = S1_{k-1} + sab1_k$$

$$S1_0 = 0$$

$$sab1_k = y^{3k-2} \binom{n}{k}$$

$$sab1_k = sab1_{k-1} \cdot y^3 \cdot \frac{n+1-k}{k}$$

$$sab1_0 = \frac{1}{y^2}$$

$$S2_k = \sum_{p=1}^{k} \frac{x^{p^2} y^p}{(3p)!}$$

$$S2_k = S2_{k-1} + sab2_k$$

$$S2_0 = 0$$

$$sab2_k = \frac{x^{k^2} y^k}{(3k)!}$$

$$sab2_k = sab2_{k-1} \cdot \frac{x^{2k-1} \cdot y}{(3k)(3k-1)(3k-2)}$$

$$sab2_k = sab2_{k-1} \cdot stx_k \cdot \frac{y}{(3k)(3k-1)(3k-2)}$$

$$sab2_0 = 1$$

Program

```
MODULE Suma0100;
(* Januar 2000.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 10;
    xgr = 1.0;
    ygr = 1.0;
  VAR
    S, st2, P, bin, S1, dfak, pdfak, S2, sab1, sab2, stx : REAL;
    k, n1, n2, k1 : CARDINAL;
    pom, yna3, xna2 : REAL;

  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                  AND (x # 0.0) AND (y # 0.0) THEN
      ok := TRUE;
```

```
      S := 0.0; S1 := 0.0; S2 := 0.0;
      st2 := 1.0; P := 1.0; sab2 := 1.0;
      bin := 1.0; dfak := 1.0; pdfak := 1.0;
      sab1 := 1.0 / (y * y); stx := 1.0 / x;

      FOR k := 1 TO n DO
        st2 := stx * 2.0;
      END;
      n1 := 2 * n + 1; n2 := n + 1;
      xna2 := x * x; yna3 := y * y * y;
      FOR k := 1 TO n DO
        k1 := 3 * k;
        st2 := st2 / 2.0;
        bin := bin * FLOAT(n1 - k) / FLOAT(k);
        stx := stx * xna2;

        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;

        sab1 := sab1 * yna3 * FLOAT(n2 - k) / FLOAT(k);
        S1 := S1 + sab1;

        sab2 := sab2 * stx * y / FLOAT(k1 * (k1 - 1) * (k1 - 2));
        S2 := S2 + sab2;

        P := P * bin * stx * dfak * S1 / (3.0 + S2);
        S := S + st2 * P;
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
```

```
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;

END Suma0100.
```

Objašnjenje

Ovaj zadatak je tipski potpuno identičan prethodnom. Potrebno je shvatiti da se pre svakog izvodjenja moraju prvo uraditi transformacije. Ponovo imamo grupisanje radi manje računanja, a kao i svakom zadatku mora se paziti na redosled naredbi. Ispisan je radi lakšeg shvatanja i boljeg razumevanja.

1.4.3 3. Zadatak - April 1999

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{x^{2k+1}}{(2k+1)!} \sum_{i=1}^{k} \sum_{j=1}^{i} \frac{2ij^2(x^{j^2} + i!!)}{\binom{2k+1}{3}}$$

$$F(x,y,n) = 2 \sum_{k=1}^{n} \frac{x^{2k+1}}{\binom{2k+1}{3}(2k+1)!} \sum_{i=1}^{k} i \left(\sum_{j=1}^{i} j^2 x^{j^2} + i!! \sum_{j=1}^{i} j^2 \right)$$

.

$$F(x,y,n) = 2 \cdot S_k$$

$$S_k = S_{k-1} + cin_k \cdot S1_k$$

$$S_1 = \frac{1}{6} \cdot x^3(x+1)$$

$$cin_k = \frac{x^{2k+1}}{\binom{2k+1}{3}(2k+1)!}$$

$$cin_k = cin_{k-1} \cdot \frac{(2k-2)(2k-3)}{(2k+1)^2(2k)^2} \cdot x^2$$

$$cin_1 = \frac{x^3}{6}$$

$$S1_k = \sum_{i=1}^{k} i \left(\sum_{j=1}^{i} j^2 x^{j^2} + i!! \sum_{j=1}^{i} j^2 \right)$$

$$S1_k = S1_{k-1} + k(S2_k + dfak_k \cdot S3_k)$$

$$S1_1 = x + 1$$

$$S2_k = \sum_{j=1}^{k} j^2 x^{j^2}$$

$$S2_k = S2_k + k^2 \cdot stx_k$$

$$S2_1 = x$$

$$stx_k = x^{k^2}$$

$$stx_k = stx_{k-1} \cdot x^{2k-1}$$

$$pstx_k = x^{2k-1}$$

$$stx_k = stx_{k-1} \cdot pstx_k$$

$$stx_1 = x$$

$$pstx_k = pstx_{k-1} \cdot x^2$$

$$pstx_1 = x$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_{k-1} \cdot k$$

$$pdfak_k = dfak_{k-1}$$

$$dfak_1 = 1$$

$$pdfak_1 = 1$$

$$S3_k = \sum_{j=1}^{k} j^2$$

$$S3_k = S3_{k-1} + k^2$$

$$S3_1 = 1$$

Program

```
MODULE Suma0499;
(* April 1999.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 50;
    xgr = 1.0;
    ygr = 1.0;
  VAR
    S, cin, S1, S2, dfak, pdfak, S3, stx, pstx : REAL;
    k, k1, k2 : CARDINAL;
    xna2, pom : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr)
                  AND (ABS(y) <= ygr) THEN
      ok := TRUE;
      cin := x * x * x / 6.0;
      pstx := x; stx := x;
      S3 := 1.0; S2 := x;
      S1 := S2 + S3;
      S := cin * S1;
      dfak := 1.0; pdfak := 1.0;
      xna2 := x * x;
      FOR k := 2 TO n DO
        pstx := xna2 * pstx;
        stx := pstx * stx;
        k2 := k * k; k1 := 2 * k;
```

```
        S2 := S2 + FLOAT(k2) * stx;
        S3 := S3 + FLOAT(k2);

        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;

        S1 := S1 + FLOAT(k) * (S2 + dfak * S3);
        cin := xna2 * FLOAT((k - 1) * (k1 - 3)) * cin
                    / FLOAT((k1 + 1) * (k1 + 1) * k1 * k);
        S := S + cin * S1;
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN 2.0 * S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0499.
```

Objašnjenje

Ovaj zadatak je izabran jer se na prvi pogled ne vidi da se radi sve u jednoj petlji. Potrebna je transformacija sume koju smo spominjali u prethodnom odeljku. Takodje, sama inicijalizacija ili pocetne vrednosti kreću od jedinice, a ne od nule kako smo navikli jer se u rekurentnoj vezi promenljive *cin* izmedju ostalog nalazi i izraz $(2k-2)$, gde ako bi zamenili sa k = 1, dobili bi nulu i samim tim ceo proizvod bi stalno bio nula. Zbog toga se naša glavna petlja kreće u programu od 2 do n. Ovakva specificna inicijalizacija zavisi samo od izraza oblika $(k-1)$, $(2k-2)$, $(3k-3)$, ... a oni se jedino mogu dobiti kada se pravi rekurentna veza od dela formule koji sadrži binomni koeficijent ili faktorijel. Zato se predlaže da se uvek prvo proveri rekurentna veza ovih elemenata (ako ih ima) i na osnovu njih odredi da li je inicijalizacija normalna ili specifična.

1.4.4 4. zadatak - Juni 2000

Matematičko izvodjenje

$$F(x,y,n)=\sum_{k=1}^{n}\frac{k!!-\sum\limits_{i=1}^{n-k}x^{k^2}\dfrac{1-i!}{x^i+y^{2i}}}{1+\prod\limits_{j=1}^{2k}(x^j-j)}$$

$$F(x,y,n)=\sum_{k=1}^{n}\frac{k!!-x^{k^2}\sum\limits_{i=1}^{n-k}\dfrac{1-i!}{x^i+y^{2i}}}{1+\prod\limits_{j=1}^{2k}(x^j-j)}$$

$$S_k=S_{k-1}+\frac{dfak_k-stx_k\cdot S1_k}{1+P_k}$$

$$S_0=0$$

$$dfak_k=k!!$$

$$pdfak_k=(k-1)!!$$

$$dfak_k=pdfak_{k-1}\cdot k$$

$$pdfak_k=dfak_{k-1}$$

$$dfak_0=1$$

$$pdafk_0=1$$

$$stx1_k = x^{k^2}$$
$$stx1_k = stx1_{k-1} \cdot x^{2k-1}$$
$$pstx1_k = x^{2k-1}$$
$$stx1_k = stx1_{k-1} \cdot pstx1_k$$
$$stx1_0 = 1$$
$$pstx1_k = pstx1_{k-1} \cdot x^2$$
$$pstx1_0 = \frac{1}{x}$$
$$S1_k = \sum_{i=1}^{n-k} \frac{1-i!}{x^i + y^{2i}}$$
$$S1_k = S1_{k-1} - \frac{1 - fak_k}{stx2_k + sty_k}$$
$$S1_0 = \sum_{i=1}^{n} \frac{1-i!}{x^i + y^{2i}}$$
$$fak_k = (n-k+1)!$$
$$fak_k = fak_{k-1} \cdot \frac{1}{n+2-k}$$
$$fak_0 = (n+1)!$$
$$stx2_k = x^{n-k+1}$$
$$stx2_k = stx2_{k-1} \cdot \frac{1}{x}$$
$$stx2_0 = x^{n+1}$$
$$sty_k = y^{2(n-k+1)}$$
$$sty_k = sty_{k-1} \cdot \frac{1}{y^2}$$
$$sty_0 = y^{2n+2}$$
$$P_k = \prod_{j=1}^{2k} (x^j - j)$$
$$\frac{P_k}{P_{k-1}} = \frac{\prod_{j=1}^{2k} (x^j - j)}{\prod_{j=1}^{2(k-1)} (x^j - j)}$$

$$\frac{P_k}{P_{k-1}} = (x^{2k} - 2k)(x^{2k-1} - (2k-1))$$

$$P_k = P_{k-1}(x^{2k} - 2k)(x^{2k-1} - (2k-1))$$

$$P_k = P_{k-1}(stx3_k - 2k)(stx4_k - (2k-1))$$

$$P_0 = 1$$

$$stx3_k = x^{2k}$$

$$stx3_k = stx3_{k-1} \cdot x^2$$

$$stx3_0 = 1$$

$$stx4_k = x^{2k-1}$$

$$stx4_k = pstx1_k$$

Program

```
MODULE Suma0600;
(* Juni 2000.  1. zadatak *)
FROM IO IMPORT WrStr, WrReal, WrLn, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 23;
    xgr = 0.2;
    ygr = 0.4;
  VAR
    S, dfak, pdfak, S1, stx1, pstx1, fak, stx2, sty, P, stx3 : REAL;
    k, k1 : CARDINAL;
    pom, yna2, xna2, n1 : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (xgr <= ABS(x))
                  AND (ABS(y) <= ygr) AND (x # 0.0) THEN
      ok := TRUE;
      S := 0.0; S1 := 0.0;
      stx1 := 1.0; fak := 1.0; stx2 := 1.0; sty := 1.0; P := 1.0;
      pstx1 := 1.0 / x; dfak := 1.0; pdfak := 1.0; stx3 := 1.0;
      yna2 := y * y; xna2 := x * x; n1 := n + 2;
```

```
      FOR k := 1 TO n DO
        fak := FLOAT(k) * fak;
        stx2 := x * stx2;
        sty := yna2 * sty;
        S1 := S1 + (1.0 - fak) / (stx2 + sty);
      END;
      fak := FLOAT(n + 1) * fak;
      stx2 := x * stx2;
      sty := yna2 * sty;

      FOR k := 1 TO n DO
        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;

        pstx1 := xna2 * pstx1;
        stx1 := pstx1 * stx1;

        fak := fak / FLOAT(n1 - k);
        stx2 := stx2 / x;
        sty := sty / yna2;
        S1 := S1 - (1.0 - fak) / (stx2 + sty);

        stx3 := xna2 * stx3; k1 := 2 * k;
        P := P * (stx3 - FLOAT(k1)) * (pstx1 - FLOAT(k1 - 1));

        S := S + (dfak - stx1 * S1) / (1.0 + P);
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;


VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
BEGIN
```

```
    WrStr('Unesite x: ');
    x := RdReal(); WrLn;
    WrStr('Unesite y: ');
    y := RdReal(); WrLn;
    WrStr('Unesite n: ');
    n := RdCard(); WrLn;

    rez := F(x, y, n, ok);
    IF ok THEN
      WrStr('F(x, y, n) = ');
      WrReal(rez, 8, 15);
    ELSE
      WrStr('Uneli ste nedozvoljene argumente');
    END;
  END Suma0600.
```

Objašnjenje

U ovom primeru je opisano rešavanje suma i proizvoda sa nestandardnim indeksima.

U prvom slučaju imali smo $S1_k$ sumu koja ide do $n-k$. Kada je $k=0$ ona ima n elemenata, $k=1$ tada $n-1$, a za $k=2$ ona ima $n-2$ elemenata. Očigledno je da se vrši oduzimanje, a za opšte k oduzima se $n-k+1$ elemenat. Takodje, pošto je njena početna vrednost suma od n elemenata, morali smo ubaciti jednu pomoćnu FOR petlju za njeno izračunavanje. Kako inicijalne vrednosti pojedinih delova sume idu do n + 1, na kraju je bilo potrebno još jedno množenje.

U drugom slučaju imali smo P_k proizvod koji ide do $2k$. Rekurzivnim deljenjem, dobijamo da se sve skrati izuzev $2k$ i $2k-1$ člana. Slično se izvodi suma koja ide do $2k$.

1.4.5 5. Zadatak - Januar 1997

Matematičko izvodjenje

$$F(x,n)=\sum_{k=0}^{n}\frac{\dfrac{\binom{4k}{k}\binom{n}{k}}{3^k}\displaystyle\sum_{j=0}^{k}\frac{(-1)^j x^{j+1}}{2+(3k-1)^2}}{1+\displaystyle\prod_{j=0}^{k}\frac{x^{4j}}{(2k+1)^2}}$$

$$F(x,n)=\sum_{k=0}^{n}\frac{\dfrac{\binom{4k}{k}\binom{n}{k}}{3^k}\sum_{j=0}^{k}(-1)^j x^{j+1}}{\Big(2+(3k-1)^2\Big)\left(1+\dfrac{\prod_{j=0}^{k}x^{4j}}{\prod_{j=0}^{k}(2k+1)^2}\right)}$$

$$S_k=S_{k-1}+\frac{cin_k\cdot S1_k}{(2+(3k-1)^2)\cdot\left(1+\dfrac{P1_k}{P2_j}\right)}$$

$$S_0=\frac{x}{6}$$

$$cin_k=\frac{\binom{4k}{k}\binom{n}{k}}{3^k}$$

$$cin_k=cin_{k-1}\cdot 4\cdot\frac{(4k-1)(4k-2)(4k-3)(n+1-k)}{(3k)^2(3k-1)(3k-2)}$$

$$cin_0=1$$

$$S1_k=\sum_{j=0}^{k}(-1)^j x^{j+1}$$

$$S1_k=S1_{k-1}+stx1_k$$

$$S1_0=x$$

$$stx1_k=(-1)^k x^{k+1}$$

$$stx1_k=-stx1_{k-1}\cdot x$$

$$stx_0=x$$

$$P1_k=\prod_{j=0}^{k}x^{4j}$$

$$P1_k=P1_{k-1}\cdot stx2_k$$

$$P1_k=1$$

$$stx2_k=x^{4k}$$

$$stx2_k=stx2_k\cdot x^4$$

$$stx2_0=1$$

$$P2_j = \prod_{j=0}^{k} (2k+1)^2$$

$$P2_j = P2_{j-1} \cdot (2k+1)^2$$

$$P2_0 = 1$$

Program

```
MODULE Suma0197;
(* Januar 1997.   1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 50;
    xgr = 1.0;
  VAR
    S, cin, S1, P1, stx2, P2, stx1 : REAL;
    k, j, k1, k2, k3, k4, n1 : CARDINAL;
    xna4 : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) THEN
      ok := TRUE;
      S := x / 6.0; S1 := x; stx1 := x;
      cin := 1.0; P1 := 1.0; stx2 := 1.0;
      n1 := n + 1; xna4 := x * x * x * x;

      FOR k := 1 TO n DO
        k1 := 4 * k; k2 := 3 * k;
        cin := cin * FLOAT(4 * (k1 - 1) * (k1 - 2) * (k1 - 3) * (n1 - k)) /
                     FLOAT(k2 * k2 * (k2 - 1) * (k2 - 2));

        stx1 := -stx1 * x;
        S1 := S1 + stx1;

        stx2 := stx2 * xna4;
        P1 := P1 * stx2;
```

```
        k3 := (2 * k + 1); k3 := k3 * k3;
        P2 := 1.0;

        FOR j := 0 TO k DO
          P2 := P2 * FLOAT(k3);
        END;

        k4 := k2 - 1; k4 := k4 * k4;
        S := S + cin * S1 / ((2.0 + FLOAT(k4)) * (1.0 + P1 / P2));
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;


VAR
  x, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;
  rez := F(x, n, ok);
  IF ok THEN
    WrStr('F(x, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0197.
```

Objašnjenje

U ovom zadatku zanimljiva nam je promenljiva $P2_j$. Naime, u njenom proizvodu se izmedju ostalih nalazi i k. Pošto se nalazi i u indeksu proizvoda taj proizvod ne možemo odmah izračunati. Moramo uvesti dodatnu promenljivu j, tako da sada k posmatramo kao konstantu, a ne brojač odnosno promenljivu. Ostatak radimo kao do sada. Jedino treba voditi računa da se inicijalne vrednosti promenljivih unutar ove druge petlje nalaze odmah pre nje, tačnije unutar prve petlje.

Potrebno je primetiti da se ovakva suma ne bi radila u dodatnoj petlji jer bi cela suma bila jednaka $k \cdot (2k+1)^2$, što se lako izračunava.

1.4.6 6. Zadatak - Januar 2002

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{k!!x^{k^2} \prod_{j=1}^{k} \frac{y^{2j-1}+k}{\sum_{i=1}^{j} 2^{k+i} x^{3i}}}{1 + \sum_{j=1}^{k} \frac{y^j}{(k!)^2}}$$

$$F(x,y,n) = \sum_{k=1}^{n} \frac{k!! \left(\frac{x}{2}\right)^{k^2} \left(\frac{\prod_{j=1}^{k} y^{2j-1}+k}{\prod_{j=1}^{k} \sum_{i=1}^{j} 2^i x^{3i}} \right)}{1 + \frac{1}{(k!)^2} \sum_{j=1}^{k} y^j}$$

$$S_k = S_{k-1} + \frac{dfak_k \cdot stx_k \cdot P1_j}{P2_k \cdot \left(1 + \frac{S1_k}{fak_k^2}\right)}$$

$$S_0 = 0$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_{k-1} \cdot k$$

$$pdfak_k = dfak_{k-1}$$

$$dfak_0 = 1$$

$$pdfak_0 = 1$$

$$stx_k = \left(\frac{x}{2}\right)^{k^2}$$

$$stx_k = stx_{k-1} \cdot \left(\frac{x}{2}\right)^{2k-1}$$

$$pstx_k = \left(\frac{x}{2}\right)^{2k-1}$$

$$stx_k = stx_{k-1} \cdot pstx_k$$

$$stx_0 = 1$$

$$pstx_k = pstx_{k-1} \cdot \left(\frac{x}{2}\right)^2$$

$$pstx_0 = \frac{2}{x}$$

$$P1_j = \prod_{j=1}^{k} (y^{2j-1} + k)$$

$$P1_j = P1_{j-1} \cdot (sty_j + k)$$

$$P1_0 = 1$$

$$sty_j = y^{2j-1}$$

$$sty_j = sty_{j-1} \cdot y^2$$

$$sty_0 = \frac{1}{y}$$

$$P2_k = \prod_{j=1}^{k} \sum_{i=1}^{j} 2^i x^{3i}$$

$$P2_k = P2_{k-1} \cdot S2_k$$

$$P2_0 = 1$$

$$S2_k = \sum_{i=1}^{k} 2^i x^{3i}$$

$$S2_k = S2_{k-1} + sab_k$$

$$S2_0 = 0$$

$$sab_k = 2^k x^{3k}$$

$$sab_k = sab_{k-1} \cdot 2x^3$$

$$sab_0 = 1$$

$$S1_k = \sum_{j=1}^{k} y^j$$

$$S1_k = S1_{k-1} + sty2_k$$

$$S1_0 = 0$$

$$sty2_k = y^k$$

$$sty2_k = sty2_{k-1} \cdot y$$

$$sty2_0 = 1$$

$$fak_k = k!$$

$$fak_k = fak_{k-1} \cdot k$$

$$fak_0 = 1$$

Program

```
MODULE Suma0102;
(*  Januar 2002.   1. zadatak  *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 10;
    xdgr = 0.1;
    xggr = 1.0;
    ydgr = 0.0;
    yggr = 1.0;
  VAR
    S, dfak, pdfak, stx, pstx, P1, P2, S1, S2, fak, sty1, sab, sty2 : REAL;
    k, j : CARDINAL;
    pom, xna2, xna3, yna2 : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (xdgr <= x) AND (x <= xggr)
                  AND (ydgr <= y) AND (y <= yggr) AND (y # 0.0) THEN
      ok := TRUE;
      S := 0.0; S1 := 0.0; S2 := 0.0;
      stx := 1.0; dfak := 1.0; pdfak := 1.0; P2 := 1.0;
```

```
      sab := 1.0; sty2 := 1.0; fak := 1.0; pstx := 2.0 / x;
      xna2 := x * x / 4.0; xna3 := 2.0 * x * x * x; yna2 := y * y;

      FOR k := 1 TO n DO
        pom := pdfak;
        pdfak := dfak;
        dfak := pom * FLOAT(k);

        pstx := pstx * xna2;
        stx := stx * pstx;

        sty1 := 1.0 / y;
        P1 := 1.0;

        FOR j := 1 TO k DO
          sty1 := sty1 * yna2;
          P1 := P1 * (sty1 + FLOAT(k));
        END;

        sab := sab * xna3;
        S2 := S2 + sab;
        P2 := P2 * S2;

        sty2 := sty2 * y;
        S1 := S1 + sty2;

        fak := fak * FLOAT(k);

        S := S + dfak * stx * P1 / (P2 * (1.0 + S1 / (fak * fak)));
      END;
    ELSE
      ok := FALSE;
      S := 0.0
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
```

```
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0102.
```

Objašnjenje

U ovom zadatku se takodje jedan deo radi pod drugom petljom. Opis je slican prethodnom.

Takodje, $2^{k+i} = 2^k \cdot 2^i$. Prvi deo ide ispred proizvoda pa postaje 2^{k^2}, a onda se spaja zbog istog stepena i postaje $\left(\frac{x}{2}\right)^{k^2}$.

1.4.7 7. Zadatak - Avgust 2000

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{y^{2k^2} 2^{k+2} \sum_{j=1}^{k} \left(\sum_{i=1}^{j} k^2 \left(\frac{x}{2}\right)^{2i} \right)}{k!! \prod_{j=1}^{k} \binom{2k}{j}}$$

$$F(x,y,n) = \sum_{k=1}^{n} \frac{y^{2k^2} 2^{k+2} k^2 \sum_{j=1}^{k} \left(\sum_{i=1}^{j} \left(\frac{x}{2}\right)^{2i} \right)}{k!! \prod_{j=1}^{k} \binom{2k}{j}}$$

$$S_k = S_{k-1} + \frac{cin_k \cdot k^2 \cdot S1_k}{dfak_k \cdot P_j}$$

$$S_0 = 0$$

$$cin_k = y^{2k^2} 2^{k+2}$$

$$cin_k = cin_{k-1} \cdot 2 \cdot y^{4k-2}$$

$$pcin_k = y^{4k-2}$$

$$cin_k = cin_{k-1} \cdot 2 \cdot pcin_k$$

$$cin_0 = 1$$

$$pcin_k = pcin_{k-1} \cdot y^4$$

$$pcin_0 = \frac{1}{y^2}$$

$$S1_k = \sum_{j=1}^{k} \left(\sum_{i=1}^{j} \left(\frac{x}{2} \right)^{2i} \right)$$

$$S1_k = S1_{k-1} + S2_k$$

$$S1_0 = 0$$

$$S2_k = \sum_{i=1}^{k} \left(\frac{x}{2} \right)^{2i}$$

$$S2_k = S2_{k-1} + stx_k$$

$$S2_0 = 0$$

$$stx_k = \left(\frac{x}{2} \right)^{2k}$$

$$stx_k = stx_{k-1} \cdot \left(\frac{x}{2} \right)^{2}$$

$$stx_0 = 1$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_{k-1} \cdot k$$

$$pdfak_k = dfak_{k-1}$$

$$dfak_0 = 1$$

$$pdfak_0 = 1$$

$$P_j = \prod_{j=1}^{k} \binom{2k}{j}$$

$$P_j = P_{j-1} \cdot bin_j$$

$$P_0 = 1$$

$$bin_j = \binom{2k}{j}$$

$$bin_j = bin_{j-1} \cdot \frac{2k+1-j}{j}$$

$$bin_0 = 1$$

Program

```
MODULE Suma0800;
(* Avgust 2000. 1. zadatak *)
FROM IO IMPORT WrStr, WrLn, RdReal, WrReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 10;
    xgr = 1.0;
    ygr = 1.0;

  VAR
    S, cin, pcin, S1, S2, stx, P, bin, dfak, pdfak : REAL;
    yna4, xna2, pom : REAL;
    k, j, k1 : CARDINAL;

  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                  AND (x # 0.0) AND (y # 0.0) THEN
      ok := TRUE;
      S := 0.0; S1 := 0.0; S2 := 0.0;
      stx := 1.0; dfak := 1.0; pdfak := 1.0;
      cin := 4.0; pcin := 1.0 / (y * y);
      yna4 := y * y * y * y; xna2 := x * x / 4.0;
```

```
      FOR k := 1 TO n DO
          pcin := pcin * yna4;
          cin := cin * 2.0 * pcin;

          stx := stx * xna2;
          S2 := S2 + stx;
          S1 := S1 + S2;

          P := 1.0; bin := 1.0;
          k1 := 2 * k + 1;
          FOR j := 1 TO k DO
            bin := bin * FLOAT(k1 - j) / FLOAT(j);
            P := P * bin;
          END;

          pom := pdfak;
          pdfak := dfak;
          dfak := FLOAT(k) * pom;

          S := S + cin * FLOAT(k * k) * S1 / (dfak * P);
        END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  ok : BOOLEAN;
  x, y, rez : REAL;
  n : CARDINAL;

BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;
```

```
    rez := F(x, y, n, ok);
    IF ok THEN
      WrStr('F(x, y, n) = ');
      WrReal(rez, 8, 15);
    ELSE
      WrStr('Pogresno ste uneli argumente')
    END;
  END Suma0800.
```

Objašnjenje

P_j je drugi tip proizvoda za koji su potrebne dve FOR petlje. Razlog je isti, uvodi se nova promenljiva da bi se stara u njoj ponašala kao konstanta, a sa tim već znamo da radimo.

1.4.8 8. Zadatak - Maj 1998

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{\binom{2n}{k} n^k}{3^k (k!!)} \prod_{j=1}^{k} \left(\frac{\frac{(-1)^j x^{j^2}}{j!!}}{\binom{n+3}{k}} \right) y^{3k-2}$$

$$F(x,y,n) = \sum_{k=1}^{n} \frac{\binom{2n}{k} n^k y^{3k-2}}{3^k (k!!)} \prod_{j=1}^{k} \left(\frac{(-1)^j x^{j^2}}{j!!} \right) \prod_{j=1}^{k} \binom{n+3}{k}$$

$$S_k = S_{k-1} + \frac{cin_k}{dfak_k} \cdot P1_k \cdot P2_j$$

$$S_0 = 0$$

$$cin_k = \frac{\binom{2n}{k} n^k y^{3k-2}}{3^k}$$

$$cin_k = cin_{k-1} \cdot \frac{ny^3}{3} \cdot \frac{2n+1-k}{k}$$

$$cin_0 = \frac{1}{y^2}$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_{k-1} \cdot k$$

$$pdfak_k = dfak_{k-1}$$

$$dfak_0 = 1$$

$$pdfak_0 = 1$$

$$P1_k = \prod_{j=1}^{k} \left(\frac{(-1)^j x^{j^2}}{j!!} \right)$$

$$P1_k = P1_{k-1} \cdot \frac{stx_k}{dfak_k}$$

$$P1_0 = 1$$

$$stx_k = (-1)^k x^{k^2}$$

$$stx_k = -stx_{k-1} \cdot x^{2k-1}$$

$$pstx_k = x^{2k-1}$$

$$stx_k = -stx_{k-1} \cdot pstx_k$$

$$stx_0 = 1$$

$$pstx_k = pstx_{k-1} \cdot x^2$$

$$pstx_0 = \frac{1}{x^2}$$

$$P2_j = \prod_{j=1}^{k} \binom{n+3}{k}$$

$$P2_j = P2_{j-1} \cdot bin_k$$

$$P2_0 = 1$$

$$bin_k = \binom{n+3}{k}$$

$$bin_k = bin_{k-1} \cdot \frac{n+4-k}{k}$$

$$bin_0 = 1$$

Program

```
MODULE Suma0598;
(* Maj 1998. 1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 20;
    xgr = 1.0;
    ygr = 0.5;
  VAR
    S, cin, dfak, pdfak, P1, P2, stx, pstx, bin : REAL;
    k, j, n1, n2 : CARDINAL;
    pom, xna2, yna3 : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                   AND (y # 0.0) AND (x # 0.0) THEN
      ok := TRUE;
      S := 0.0;
      P2 := 1.0; bin := 1.0; stx := 1.0; pstx := 1.0 / x;
      cin := 1.0 / (y * y); dfak := 1.0; pdfak := 1.0;
      xna2 := x * x; yna3 := y * y * y; n1 := 2 * n + 1; n2 := n + 4;

      FOR k := 1 TO n DO
        cin := FLOAT((n1 - k) * n) * yna3 * cin / FLOAT(3 * k);

        bin := FLOAT(n2 - k) * bin / FLOAT(k);
        P1 := 1.0;
        FOR j := 1 TO k DO
          P1 := P1 * bin;
        END;

        pstx := xna2 * pstx;
        stx := - pstx * stx;
```

```
        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;
        P2 := P2 * stx / dfak;

        S := S + cin * P1 * P2 / dfak;
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0598.
```

Objašnjenje

Novi tip proizvoda.

1.4.9 9. Zadatak - Decembar 2000

Matematičko izvodjenje

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{n+3}{k}x^{2k^2}+\sum_{i=1}^{k}\frac{y^3 i}{2i^2+3ki}}{(k!!)^k}$$

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{n+3}{k}x^{2k^2}+\sum_{i=1}^{k}\frac{y^3 i}{i(2i+3k)}}{\prod_{j=1}^{k}(k!!)}$$

$$S_k=S_{k-1}+\frac{sab_k+S1_j}{P_j}$$

$$S_0=0$$

$$sab_k=\binom{n+3}{k}x^{2k^2}$$

$$sab_k=sab_{k-1}\cdot x^{4k-2}\frac{n+4-k}{k}$$

$$psab_k=x^{2k-1}$$

$$sab_k=sab_{k-1}\cdot psab\cdot\frac{n+4-k}{k}$$

$$sab_0=1$$

$$psab_k=psab_{k-1}\cdot x^4$$

$$psab_0=\frac{1}{x^2}$$

$$S1_j=\sum_{j=1}^{k}\frac{y^{3j}}{j(2j+3k)}$$

$$S1_j=S1_{j-1}+\frac{sty_j}{j\cdot(2j+3k)}$$

$$S1_0=0$$

$$sty_j=y^{3k}$$

$$sty_j=sty_{j-1}\cdot y^3$$

$$sty_0=1$$

$$P_j = \prod_{j=1}^{k} (k!!)$$

$$P_j = P_{j-1} \cdot dfak_k$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_{k-1} \cdot k$$

$$pdfak_k = dfakk - 1$$

$$dfak_0 = 1$$

$$pdfak_0 = 1$$

Program

```
MODULE Suma1200;
(* Decembar 2000.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 10;
    xgr = 1.0;
    ygr = 1.0;

  VAR
    S, sab, S1, P, psab, sty, dfak, pdfak : REAL;
    k, j, n1 : CARDINAL;
    pom, xna4, yna3 : REAL;

  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                  AND (x # 0.0) THEN
      ok := TRUE;
      S := 0.0;
      sab := 1.0; dfak := 1.0; pdfak := 1.0;
      psab := 1.0 / (x * x);
```

```
      xna4 := x * x * x * x; yna3 := y * y * y; n1 := n + 4;

      FOR k := 1 TO n DO
        psab := psab * xna4;
        sab := sab* psab * FLOAT(n1 - k) / FLOAT(k);

        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;

        P := 1.0; S1 := 0.0; sty := 1.0;

        FOR j := 1 TO k DO
          sty := sty * yna3;
          S1 := S1 + sty / FLOAT(j * (2 * j + 3 * k));

          P := P * dfak;
        END;

        S := S + (sab + S1) / P;
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;
```

```
    rez := F(x, y, n, ok);
    IF ok THEN
      WrStr('F(x, y, n) = ');
      WrReal(rez, 8, 15);
    ELSE
      WrStr('Uneli ste nedozvoljene argumente');
    END;
  END Suma1200.
```

Objašnjenje

Novi tip proizvoda, takodje i prvi primer sume u drugoj petlji.

1.4.10 10. Zadatak - Maj 1999

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{\binom{2n}{k}(2x)^{2k+1}\sum_{p=1}^{2k} k\frac{y^{3p-1}}{1+p!}}{(k!!)^{2k}(3+x)^{3k}\prod_{p=1}^{k}(2+y)^p}$$

$$F(x,y,n) = \sum_{k=1}^{n} \frac{\binom{2n}{k}(2x)^{2k+1}k\sum_{p=1}^{2k} \frac{y^3p-1}{1+p!}}{\prod_{j=1}^{k}(k!!)^{2}(3+x)^{3k}\prod_{p=1}^{k}(2+y)^p}$$

$$S_k = S_{k-1} + \frac{cin_k \cdot k \cdot S1_k}{P1_j \cdot P2_k}$$

$$S_0 = 0$$

$$cin_k = \frac{\binom{2n}{k}(2x)^{2k+1}}{(3+x)^{3k}}$$

$$cin_k = cin_{k-1} \cdot \frac{4x^2}{(3+x)^3} \cdot \frac{2n+1-k}{k}$$

$$cin_0 = 2 \cdot x$$

$$S1_k = \sum_{p=1}^{2k} \frac{y^{3p}-1}{1+p!}$$

$$S1_k - S1_{k-1} = \sum_{p=1}^{2k} \frac{y^{3p}-1}{1+p!} - \sum_{p=1}^{2(k-1)} \frac{y^{3p}-1}{1+p!}$$

$$S1_k - S1_{k-1} = \frac{y^{6k-1}}{1+(2k)!} + \frac{y^{6k-4}}{1+(2k-1)!}$$

$$S1_k = S1_{k-1} + \frac{sty1_k}{1+fak1_k} + \frac{sty2_k}{1+fak2_k}$$

$$S1_0 = 0$$

$$sty1_k = y^{6k-1}$$

$$sty1_k = sty1_{k-1} \cdot y^6$$

$$sty1_0 = \frac{1}{y}$$

$$fak1_k = (2k)!$$

$$fak1_k = fak1_{k-1} \cdot (2k)(2k-1)$$

$$fak1_0 = 1$$

$$sty2_k = y^{6k-4}$$

$$sty2_k = sty2_{k-1} \cdot y^6$$

$$sty2_0 = \frac{1}{y^4}$$

$$fak2_k = (2k-1)!$$

$$fak2_k = fak2_{k-1} \cdot (2k-1)(2k-2)$$

$$fak2_0 = 1$$

$$fak2_k = fak1_{k-1} \cdot (2k-1)$$

$$fak1_k = fak2_{k-1} \cdot 2k$$

$$P_j = \prod_{j=1}^{k} (k!!)^2$$

$$P_j = P_{j-1} \cdot dfak_k^2$$

$$P_0 = 1$$

$$dfak_k = k!!$$

$$pdfak_k = (k-1)!!$$

$$dfak_k = pdfak_{k-1} \cdot k$$

$$pdfak_k = dfak_{k-1}$$

$$dfak_0 = 1$$

$$pdfak_0 = 1$$

$$P2_k = \prod_{p=1}^{k} (2+y)^p$$

$$P2_k = P2_{k-1} \cdot sty3_k$$

$$P2_0 = 1$$

$$sty3_k = (2+y)^k$$

$$sty3_k = sty3_{k-1} \cdot (2+y)$$

$$sty3_0 = 1$$

Program

```
MODULE Suma0599;
(* Maj 1999.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 50;
    xgr = 1.0;
    ygr = 1.0;

  VAR
    S, cin, S1, P1, P2, sty1, fak1, sty2, fak2, dfak, pdfak, sty3 : REAL;
    k, j, n1 : CARDINAL;
    x1, yna6, pom, y1, dfakna2 : REAL;

  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                 AND (y # 0.0) THEN
```

```
      ok := TRUE;
      S := 0.0; S1 := 0.0;
      fak1 := 1.0; fak2 := 1.0; P2 := 1.0;
      sty3 := 1.0; dfak := 1.0; pdfak := 1.0;
      cin := 2.0 * x; sty1 := 1.0 / y; sty2 := 1.0 / (y * y * y * y);
      x1 := 3.0 + x; x1 := x1 * x1 * x1; x1 := 4.0 * x * x / x1;
      yna6 := y * y; yna6 := yna6 * yna6 * yna6; y1 := y + 2.0;
      n1 := 2 * n + 1;

      FOR k := 1 TO n DO
        cin := cin * x1 * FLOAT(n1 - k) / FLOAT(k);

        sty2 := sty2 * yna6;
        fak2 := fak1 * FLOAT(2 * k - 1);
        sty1 := yna6 * sty1;
        fak1 := fak2 * FLOAT(2 * k);
        S1 := S1 + sty1 / (1.0 + fak1) + sty2 / (1.0 + fak2);

        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;
        P1 := 1.0; dfakna2 := dfak * dfak;
        FOR j := 1 TO k DO
          P1 := P1 * dfakna2;
        END;

        sty3 := y1 * sty3;
        P2 := P2 * sty3;

        S := S + cin * FLOAT(k) * S1 / (P1 * P2);
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
```

```
    ok : BOOLEAN;

  BEGIN
    WrStr('Unesite x: ');
    x := RdReal(); WrLn;
    WrStr('Unesite y: ');
    y := RdReal(); WrLn;
    WrStr('Unesite n: ');
    n := RdCard(); WrLn;

    rez := F(x, y, n, ok);
    IF ok THEN
      WrStr('F(x, y, n) = ');
      WrReal(rez, 8, 15);
    ELSE
      WrStr('Uneli ste nedozvoljene argumente');
    END;
  END Suma0599.
```

Objašnjenje

Suma koja ide do $2k$ i već vidjeni proizvod u drugoj petlji. Zadatak za vežbu.

1.4.11 11. Zadatak - Januar 2001

Matematičko izvodjenje

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{n}{k}y^{2k-1}\prod_{j=1}^{k}\frac{x^j}{k}\sum_{i=1}^{j}(x+y)^i}{4k+\sum_{p=1}^{k}\frac{(p+k)^p}{y^{2p+1}}}$$

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{n}{k}y^{2k-1}\prod_{j=1}^{k}x^j\sum_{i=1}^{j}(x+y)^i}{\left(4k+\sum_{p=1}^{k}\frac{\prod_{i=1}^{p}(p+k)}{y^{2p+1}}\right)\prod_{j=1}^{k}k}$$

$$S_k = S_{k-1} + \frac{cin_k \cdot P1_k}{(4k + S1_j) \cdot P2_j}$$

$$S_0 = 0$$

$$cin_k = \binom{n}{k} y^{2k-1}$$

$$cin_k = cin_{k-1} \cdot y^2 \cdot \frac{n+1-k}{k}$$

$$cin_0 = \frac{1}{y}$$

$$P1_k = \prod_{j=1}^{k} x^j \sum_{i=1}^{j} (x+y)^i$$

$$P1_k = P1_{k-1} \cdot stx_k \cdot S2_k$$

$$P1_0 = 1$$

$$stx_k = x^k$$

$$stx_k = stx_{k-1} \cdot x$$

$$stx_0 = 1$$

$$S2_k = \sum_{i=1}^{k} (x+y)^i$$

$$S2_k = S2_{k-1} + stxy_k$$

$$S2_0 = 0$$

$$stxy_k = (x+y)^k$$

$$stxy_k = stxy_{k-1} \cdot (x+y)$$

$$stxy_0 = 1$$

$$S1_j = \sum_{j=1}^{k} \frac{\prod_{i=1}^{j} (j+k)}{y^{2j+1}}$$

$$S1_j = S1_{j-1} + \frac{P3_i}{sty_j}$$

$$S1_0 = 0$$

$$P3_i = \prod_{i=1}^{j} (j+k)$$

$$P3_i = P3_{i-1} \cdot (j + k)$$
$$P3_0 = 1$$
$$sty_j = y^{2j+1}$$
$$sty_j = sty_{j-1} \cdot y^2$$
$$sty_0 = y$$
$$P2_j = \prod_{j=1}^{k} k$$
$$P2_j = P2_{j-1} \cdot k$$
$$P2_0 = 1$$

Program

```
MODULE Suma0101;
(* Januar 2001.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 10;
    xdgr = 0.5;
    xggr = 1.0;
    ygr = 1.0;

  VAR
    S, cin, P1, P2, S1, stx, S2, stxy, P3, sty : REAL;
    k, j, i, n1, kj : CARDINAL;
    yna2, xy : REAL;

  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (xdgr <= x) AND (x <= xggr)
                  AND (ABS(y) <= ygr) AND (y # 0.0) THEN
      ok := TRUE;
      S := 0.0; S2 := 0.0;
      P1 := 1.0; stx := 1.0; stxy := 1.0;
```

```
      cin := 1.0 / y;
      n1 := n + 1; yna2 := y * y; xy := x + y;

      FOR k := 1 TO n DO
        cin := cin * yna2 * FLOAT(n1 - k) / FLOAT(k);

        stx := stx * x;
        stxy := stxy * xy;
        S2 := S2 + stxy;
        P1 := P1 * stx * S2;

        P2 := 1.0; S1 := 0.0; sty := y;
        FOR j := 1 TO k DO
          P2 := P2 * FLOAT(k);

          P3 := 1.0; kj := k + j;
          FOR i := 1 TO j DO
            P3 := P3 * FLOAT(kj);
          END;

          sty := sty * yna2;
          S1 := S1 + P3 / sty;
        END;

        S := S + cin * P1 / (P2 * (FLOAT(4 * k) + S1));
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
```

```
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0101.
```

Objašnjenje

Primer sa tri petlje. U proizvodu učestvuju dva brojača, a da bi se njima obraćali kao konstantama moramo uvesti i treći.

1.4.12 12. Zadatak - Februar 1999

Matematičko izvodjenje

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{2n+1}{k+1}(n-k)!\cdot y^{3k-2}}{1\cdot 2+2\cdot 3\cdot 4+\ldots+k\cdot(k+1)\cdot\ldots\cdot 2k}\prod_{j=1}^{k}\frac{(1-3x^{3j})^k}{13+j!!}$$

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{2n+1}{k+1}(n-k)!\cdot y^{3k-2}}{1\cdot 2+2\cdot 3\cdot 4+\ldots+k\cdot(k+1)\cdot\ldots\cdot 2k}\cdot\frac{\prod_{j=1}^{k}(1-3x^{3j})^k}{\prod_{j=1}^{k}(13+j!!)}$$

$$S_k=S_{k-1}+\frac{cin_k}{S1_k}\cdot\frac{P1_k}{P2_k}$$

$$S_0=0$$

$$cin_k=\binom{2n+1}{k+1}(n-k)!\cdot y^{3k-2}$$

$$cin_k = cin_{k-1} \cdot y^3 \cdot \frac{2n+1-k}{(n+1-k)(k+1)}$$

$$cin_0 = \frac{(2n+1) \cdot n!}{y^2}$$

$$S1_k = 1 \cdot 2 + 2 \cdot 3 \cdot 4 + \ldots + k \cdot (k+1) \cdot \ldots \cdot 2k$$

$$S1_k = \sum_{j=1}^{k} \frac{(2j)!}{(j-1)!}$$

$$S1_k = \sum_{j=1}^{k} \frac{(2j)!}{(j-1)!} \cdot \frac{j}{j}$$

$$S1_k = \sum_{j=1}^{k} j \cdot \frac{(2j)!}{j!}$$

$$S1_k = S1_{k-1} + k \cdot \frac{fak2_k}{fak_k}$$

$$S1_0 = 0$$

$$fak2_k = (2k)!$$

$$fak2_k = fak_{k-1} \cdot (2k)(2k-1)$$

$$fak2_0 = 1$$

$$fak_k = k!$$

$$fak_k = fak_{k-1} \cdot k$$

$$fak_0 = 1$$

$$P1_k = \prod_{j=1}^{k} (1-3x^{3j})^k$$

$$\frac{P1_k}{P1_{k-1}} = \frac{(1-3x^3)^k (1-3x^6)^k \cdot \ldots \cdot (1-3x^{3(k-1)})^k (1-3x^{3k})^k}{(1-3x^3)^{k-1} (1-3x^6)^{k-1} \cdot \ldots \cdot (1-3x^{3(k-1)})^{k-1}}$$

$$P1_k = P1_{k-1} \cdot (1-3x^3)(1-3x^6) \cdot \ldots \cdot (1-3x^{3(k-1)})(1-3x^{3k})^k$$

$$P1_k = P1_{k-1} \cdot \prod_{j=1}^{k} (1-3x^{3j}) \cdot \prod_{j=1}^{k-1} (1-3x^{3k})$$

$$P1_k = P1_{k-1} \cdot P3_k \cdot P4_j$$

$$P1_0 = 1$$

$$P3_k = \prod_{j=1}^{k} (1-3x^{3j})$$

$$P3_k = P3_{k-1} \cdot (1 - stx_k)$$
$$P3_0 = 1$$
$$stx_k = 3x^{3k}$$
$$stx_k = stx_{k-1} \cdot x^3$$
$$stx_0 = 3$$
$$P4_j = \prod_{j=1}^{k-1}(1 - 3x^{3k})$$
$$P4_j = P4_{j-1} \cdot (1 - stx_k)$$
$$P4_0 = 1$$
$$P2_k = \prod_{j=1}^{k}(13 + j!!)$$
$$P2_k = P2_{k-1} \cdot (13 + dfak_k)$$
$$P2_0 = 1$$
$$dfak_k = k!!$$
$$pdfak_k = (k-1)!!$$
$$dfak_k = pdfak_{k-1} \cdot k$$
$$pdfak_k = dfak_{k-1}$$
$$dfak_0 = 1$$
$$pdfak_0 = 1$$

Program

```
MODULE Suma0299;
(* Februar 1999.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 50;
    xgr = 1.0;
```

```
  ygr = 1.0;

VAR
  S, cin, S1, P1, P2, P3, P4, fak2, fak, stx, dfak, pdfak : REAL;
  k, j, k1, n1, n2 : CARDINAL;
  pom, yna3, xna3 : REAL;

BEGIN
  IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                 AND (y # 0.0) THEN
    ok := TRUE;
    S := 0.0; S1 := 0.0; P2 := 1.0; P3 := 1.0;
    fak2 := 1.0; fak := 1.0; P1 := 1.0; stx := 3.0;
    dfak := 1.0; pdfak := 1.0; cin := 1.0;
    n1 := n + 1; n2 := 2 * n + 1;

    FOR k := 1 TO n DO
      cin := cin * FLOAT(k);
    END;
    cin := cin * FLOAT(n2) / (y * y);

    yna3 := y * y * y; xna3 := x * x * x;

    FOR k := 1 TO n DO
      cin := FLOAT(n2 - k) * yna3 * cin / FLOAT((n1 - k) * (k + 1));

      fak := FLOAT(k) * fak;
      k1 := 2 * k;
      fak2 := FLOAT(k1 * (k1 - 1)) * fak2;
      S1 := S1 + FLOAT(k) * fak2 / fak;

      stx := xna3 * stx;

      pom := pdfak;
      pdfak := dfak;
      dfak := FLOAT(k) * pom;
      P2 := P2 * (13.0 + dfak);
      P3 := P3 * (1.0 - stx);

      P4 := 1.0;
      FOR j := 1 TO k - 1 DO
```

```
        P4 := P4 * (1.0 - stx);
      END;

      P1 := P1 * P4 * P3;
      S := S + cin * P1 / (S1 * P2);
    END;
  ELSE
    ok := FALSE;
    S := 0.0;
  END;
  RETURN S;
END F;


VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unestite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);

  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0299.
```

Objašnjenje

U ovom zadatku imamo raspisanu sumu koju treba pretvoriti u nešto poznato. Ovde nema nikakvog pravila i šablona. Pokušava se prepoznati čemu je slično, kako doći do toga. Ali uvek se završava pogadjanjem. Množenje i deljenje sa j je uradjeno da bi se uradila inicijalizacija za $k = 0$. Ako bismo radili bez toga, zbog izraza $(k-1)!$ morali bi da pomerimo početne vrednosti za $k = 1$.

Sa druge strane imamo zanimljiv proizvod $P1_k$ koji deluje veoma zamršeno. Medjutim, jednostavnim raspisom se lako vidi da se mnoge stvari skrate i da se samo mali deo radi u dodatnoj petlji.

1.4.13 13. Zadatak - Januar 1999

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{17^{n-k}\left(1+\dfrac{x^{3k-1}}{k^3}\right)\dbinom{5k-4}{k-1}}{\sum\limits_{j=1}^{k} y^{3j^2-3j+1}(1+j!!+(1+k)!!)} \prod_{j=1}^{k} \frac{(7-13jx)^{k-7j}}{1+y^{j^3}}$$

$$F(x,y,n) = \sum_{k=1}^{n} \frac{17^{n-k}\left(1+\dfrac{x^{3k-1}}{k^3}\right)\dbinom{5k-4}{k-1}}{\sum\limits_{j=1}^{k} y^{3j^2-3j+1}(1+j!!)+(1+k)!!\sum\limits_{j=1}^{k} y^{3j^2-3j+1}} \cdot \frac{\prod\limits_{j=1}^{k}(7-13jx)^{k-7j}}{\prod\limits_{j=1}^{k}(1+y^{j^3})}$$

$$S_k = S_{k-1} + \frac{cin_k \cdot \left(1+\dfrac{stx_k}{k^3}\right)}{S1_k + dfak_k \cdot S2_k} \cdot \frac{P1_k}{P2_k}$$

$$S_1 = \frac{17^{n-1}(1+x^2)}{4 \cdot y} \cdot \frac{(7-13x)^{-6}}{1+y}$$

$$cin_k = 17^{n-k}\binom{5k-4}{k-1}$$

$$cin_k = cin_{k-1} \cdot \frac{5}{17} \cdot \frac{(5k-4)(5k-6)(5k-7)(5k-8)}{(4k-3)(4k-4)(4k-5)(4k-6)}$$

$$cin_1 = 17^{n-1}$$

$$stx_k = x^{3k-1}$$
$$stx_k = stx_{k-1} \cdot x^3$$
$$stx_1 = x^2$$
$$dfak_k = (k+1)!!$$
$$pdfak_k = k!!$$
$$dfak_k = pdfak_{k-1} \cdot (k+1)$$
$$pdfak_k = dfak_{k-1}$$
$$dfak_1 = 2$$
$$pdfak_1 = 1$$
$$S1_k = \sum_{j=1}^{k} y^{3j^2-3j+1}(1+j!!)$$
$$S1_k = S1_{k-1} + sty1_k \cdot (1 + pdfak_k)$$
$$S1_1 = 2 \cdot y$$
$$sty1_k = y^{3j^2-3j+1}$$
$$sty1_k = sty1_{k-1} \cdot y^{6j-6}$$
$$psty1_k = y^{6j-6}$$
$$sty1_k = sty1_{k-1} \cdot psty1_k$$
$$sty1_1 = y$$
$$psty1_k = psty1_{k-1} \cdot y^6$$
$$psty1_1 = 1$$
$$S2_k = \sum_{j=1}^{k} y^{3j^2-3j+1}$$
$$S2_k = S2_{k-1} + sty1_k$$
$$S2_1 = y$$
$$P1_k = \prod_{j=1}^{k} (7-13jx)^{k-7j}$$
$$\frac{P1_k}{P1_{k-1}} = \frac{(7-13x)^{k-7} \cdot (7-13 \cdot 2 \cdot x)^{k-7 \cdot 2} \cdot \ldots \cdot (7-13 \cdot (k-1) \cdot x)^{k-7 \cdot (k-1)} \cdot (7-13kx)^{k-7k}}{(7-13x)^{(k-1)-7} \cdot (7-13 \cdot 2 \cdot x)^{(k-1)-7 \cdot 2} \cdot \ldots \cdot (7-13 \cdot (k-1) \cdot x)^{(k-1)-7 \cdot (k-1)}}$$
$$P1_k = P1_{k-1} \cdot (7-13x) \cdot (7-13 \cdot 2 \cdot x) \cdot \ldots \cdot (7-13 \cdot (k-1) \cdot x) \cdot (7-13kx)^{-6k}$$

$$P1_k = P1_{k-1} \cdot \frac{\prod_{j=1}^{k-1}(7-13jx)}{\left(\prod_{j=1}^{k}(7-13kx)\right)^6}$$

$$P1_k = P1_{k-1} \cdot \frac{P3_k}{P4_j^6}$$

$$P1_1 = \frac{1}{(7-13x)^6}$$

$$P3_k = \prod_{j=1}^{k-1}(7-13jx)$$

$$P3_k = P3_{k-1} \cdot (7 - 13 \cdot (k-1) \cdot x)$$

$$P3_1 = 1$$

$$P4_j = \prod_{j=1}^{k}(7-13kx)$$

$$P4_j = P4_{j-1} \cdot (7-13kx)$$

$$P4_0 = 1$$

$$P2_k = \prod_{j=1}^{k}(1+y^{j^3})$$

$$P2_k = P2_{k-1} \cdot (1 + sty2_k)$$

$$P2_1 = 1 + y$$

$$sty2_k = y^{k^3}$$

$$sty2_k = sty2_{k-1} \cdot y^{3k^2-3k+1}$$

$$sty2_k = sty2_{k-1} \cdot sty1_k$$

$$sty2_1 = y$$

Program

```
MODULE Suma0199;
(* Januar 1999. 1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdCard, RdReal;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
CONST
  ndgr = 1;
  nggr = 50;
  xgr = 1.0;
  ygr = 1.0;
VAR
  S, S1, S2, P1, P2, P3, P4, stx, sty1, psty1, sty2, cin, dfak, pdfak : REAL;
  k, j, k1, k2 : CARDINAL;
  xna3, yna6, pom, pom1, P4na6 : REAL;
BEGIN
  IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr) THEN
    ok := TRUE;

    cin := 1.0;
    FOR k := 1 TO n - 1 DO
      cin := cin * 17.0
    END;

    stx := x * x;
    dfak := 2.0; pdfak := 1.0;
    S1 := 2.0 * y;
    sty1 := y; psty1 := 1.0;
    S2 := y;
    P1 := 7.0 - 13.0 * x; P1 := P1 * P1 * P1; P1 := P1 * P1;
    P3 := 1.0;
    P2 := 1.0 + y;
    sty2 := y;
    S := (cin * (1.0 + stx) * P1) / ((S1 + dfak * S2) * P2);

    xna3 := x * x * x; yna6 := y * y * y; yna6 := yna6 * yna6;
```

```
    FOR k := 2 TO n DO
      k1 := 5 * k; k2 := 4 * k;
      cin := cin * 5.0 * FLOAT((k1 - 4) * (k1 - 6) * (k1 - 7) * (k1 - 8)) /
                 (17.0 * FLOAT((k2 - 3) * (k2 - 4) * (k2 - 5) * (k2 - 6)));
      stx := stx * xna3;

      pom := dfak;
      dfak := pdfak;
      pdfak := pom * FLOAT(k + 1);

      psty1 := psty1 * yna6;
      sty1 := sty1 * psty1;
      S1 := S1 + sty1 * (1.0 + pdfak);
      S2 := S2 + sty1;

      P3 := P3 * (7.0 - 13.0 * FLOAT(k - 1) * x);
      pom1 := 7.0 - 13.0 * FLOAT(k) * x;
      P4 := 1.0;
      FOR j := 1 TO k DO
        P4 := P4 * pom1
      END;
      P4na6 := P4 * P4 * P4; P4na6 := P4na6 * P4na6;
      P1 := P1 * P3 / P4na6;

      sty2 := sty2 * sty1;
      P2 := P2 * (1.0 + sty2);

      S := S + cin * (1.0 + stx / FLOAT(k * k * k)) * P1 / ((S1 + dfak * S2) * P2)
    END
  ELSE
    ok := FALSE;
    S := 0.0;
  END;
  RETURN S;
END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
```

```
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 5, 8)
  ELSE
    WrStr('Uneli ste nedozvoljene argumente')
  END
END Suma0199.
```

Objašnjenje

Zanimljiva je donja suma koju je potrebno podeliti na dve sume.

Inače, zadatak je tipski, isti kao i prethodni, proizvod koji se raspisuje i zatim jednostavno rešava. U ovom slučaju se sreć emo i sa proizvodom koji ide do $k-1$. On se lako računa tako što se umesto trenutnog brojača piše $k-1$. Naravno, prilikom inicijalizacije, za $k=1$ $P3_1 = 0$, a za $k=0$ $P3_0 = 7^{-1} = \frac{1}{7}$.

1.4.14 14. Zadatak - Oktobar 2001

Matematičko izvodjenje

$$F(x,y,n) = \sum_{k=1}^{n} \frac{\binom{2n-1}{k} y^{k^2} x^{k+n} \sum_{j=1}^{k} \binom{2j}{j} \prod_{i=1}^{j} \left((i+j)^j x^j\right)}{k!! + \sum_{p=1}^{k} x^p y^{-p+1}}$$

$$F(x,y,n) = x^n \sum_{k=1}^{n} \frac{\binom{2n-1}{k} y^{k^2} x^{k} \sum_{j=1}^{k} \binom{2j}{j} x^{j^2} \prod_{i=1}^{j} (i+j)^j}{k!! + \sum_{p=1}^{k} x^p y^{-p+1}}$$

$$S_k = S_{k-1} + \frac{cin1_k \cdot S1_k}{dfak_k + S2_k}$$
$$S_0 = 0$$
$$cin1_k = \binom{2n-1}{k} y^{k^2} x^k$$
$$cin1_k = cin1_{k-1} \cdot \frac{2n-k}{k} \cdot y^{2k-1} \cdot x$$
$$pcin1_k = y^{2k-1}$$
$$cin1_k = cin1_{k-1} \cdot pcin1_k \cdot \frac{2n-k}{k} \cdot x$$
$$cin1_0 = 1$$
$$pcin1_k = pcin1_{k-1} \cdot y^2$$
$$pcin1_0 = \frac{1}{y}$$
$$S1_k = \sum_{j=1}^{k} \binom{2j}{j} x^{j^2} \prod_{i=1}^{j} (i+j)^j$$
$$S1_k = S1_{k-1} + cin2_k \cdot P1_j$$
$$S1_0 = 0$$
$$cin2_k = \binom{2k}{k} x^{k^2}$$
$$cin2_k = cin2_{k-1} \cdot x^{2k-1} \cdot \frac{4k-2}{k}$$
$$pcin2_k = x^{2k-1}$$
$$cin2_k = cin2_{k-1} \cdot pcin2_k \cdot \frac{4k-2}{k}$$
$$cin2_0 = 1$$
$$pcin2_k = pcin2_{k-1} \cdot x^2$$
$$pcin2_0 = \frac{1}{x}$$
$$P1_j = \prod_{j=1}^{k} (k+j)^k$$
$$P1_j = \prod_{j=1}^{k} \prod_{i=1}^{k} (k+i)$$

$$P1_j = P1_{j-1} \cdot P2_k$$
$$P1_0 = 1$$
$$P2_k = \prod_{j=1}^{k} (k+j)$$
$$\frac{P2_k}{P2_{k-1}} = \frac{(k+1)\cdot(k+2)\cdot\ldots\cdot(k+k-1)\cdot(k+k)}{(k-1+1)\cdot(k-1+2)\cdot\ldots\cdot(k-1+k-1)}$$
$$P2_k = P2_{k-1} \cdot \frac{(2k-1)\cdot 2k}{k}$$
$$P2_k = P2_{k-1} \cdot (4k-2)$$
$$P2_0 = 1$$
$$dfak_k = k!!$$
$$pdfak_k = (k-1)!!$$
$$dfak_k = pdfak_{k-1} \cdot k$$
$$pdfak_k = dfak_{k-1}$$
$$dfak_0 = 1$$
$$pdfak_0 = 1$$
$$S2_k = \sum_{p=1}^{k} x^p y^{-p+1}$$
$$S2_k = S2_{k-1} + cin3_k$$
$$S2_0 = 0$$
$$cin3_k = x^k y^{-k+1}$$
$$cin3_k = cin3_{k-1} \cdot \frac{x}{y}$$
$$cin3_0 = y$$

Program

```
MODULE Suma1001;
(* Oktobar 2001.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 50;
    xgr = 1.0;
    ygr = 1.0;
  VAR
    niz : red;
    S, P1, P2, cin1, cin2, S1, S2, dfak, pdfak, pcin1, pcin2, cin3, xnan : REAL;
    k, j, n1, k1, k2 : CARDINAL;
    pom, xy, xna2, yna2 : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                  AND (x # 0.0) AND (y # 0.0) THEN
      ok := TRUE;
      S := 0.0; cin1 := 1.0; pcin1 := 1.0 / y; S2 := 0.0; S1 := 0.0;
      dfak := 1.0; pdfak := 1.0; cin3 := y; xnan := 1.0; P2 := 1.0;
      cin2 := 1.0; pcin2 := 1.0 / x;

      FOR k := 1 TO n DO
        xnan := x * xnan;
      END;
      yna2 := y * y; xna2 := x * x; n1 := 2 * n; xy := x / y;

      FOR k := 1 TO n DO
        pcin1 := yna2 * pcin1;
        cin1 := pcin1 * x * FLOAT(n1 - k) * cin1 / FLOAT(k);
        cin3 := cin3 * x / y;
```

```
        S2 := S2 + cin3;

        pom := pdfak;
        pdfak := dfak;
        dfak := FLOAT(k) * pom;

        k1 := 4 * k - 2;
        pcin2 := xna2 * pcin2;
        cin2 := FLOAT(k1) * pcin2 * cin2 / FLOAT(k);

        P2 := FLOAT(k1) * P2; P1 := 1.0;
        FOR j := 1 TO k  DO
          P1 := P2 * P1;
        END;
        S1 := S1 + cin2 * P1;

        S := S + cin1 * S1 / (dfak + S2);
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S * xnan;
  END F;

VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
    WrReal(rez, 8, 15);
```

```
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma1001.
```

Objašnjenje

Zadatak je tipski slican prethodnom.

1.4.15 15. Zadatak - Septembar 2001

Matematičko izvodjenje

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{2n+3}{k+1}x^{2k}\left(y^k\right)^k\prod_{i=1}^{k}\frac{(i+1)^{k+1}}{y^i}}{k!+\binom{2k}{k}\sum_{j=1}^{k}x^{2j}\prod_{p=1}^{j}y^{(2j+p)}}$$

$$F(x,y,n)=\sum_{k=1}^{n}\frac{\binom{2n+3}{k+1}x^{2k}y^{k^2}\frac{\prod_{i=1}^{k}(i+1)^{k+1}}{\prod_{i=1}^{k}y^i}}{k!+\binom{2k}{k}\sum_{j=1}^{k}x^{2j}y^{2j^2}\prod_{p=1}^{j}y^p}$$

$$S_k=S_{k-1}+\frac{cin1_k\cdot\frac{P1_j}{P2_k}}{fak_k+bin_k\cdot S1_k}$$

$$S_0=0$$

$$cin1_k=\binom{2n+3}{k+1}x^{2k}y^{k^2}$$

$$cin1_k=cin1_{k-1}\cdot\frac{2n+3-k}{k+1}\cdot x^2\cdot y^{2k-1}$$

$$pcin1_k=y^{2k-1}$$

$$cin1_k=cin1_{k-1}\cdot\frac{2n+3-k}{k+1}\cdot pcin1_k\cdot x^2$$

$$cin1_0 = 2n + 3$$
$$pcin1_k = pcin1_{k-1} \cdot y^2$$
$$pcin1_k = \frac{1}{y}$$
$$P1_j = \prod_{i=1}^{k} (i+1)^{k+1}$$
$$P1_j = \prod_{j=1}^{k+1} \prod_{i=1}^{k} (i+1)$$
$$P1_j = \prod_{j=1}^{k+1} (k+1)!$$
$$P1_j = P1_{j-1} \cdot fak1_k$$
$$P1_0 = 1$$
$$fak1_k = (k+1)!$$
$$fak_k = k!$$
$$fak1_k = fak_k \cdot (k+1)$$
$$fak_k = fak1_{k-1}$$
$$fak1_0 = 1$$
$$fak_0 = 1$$
$$P2_k = \prod_{i=1}^{k} y^i$$
$$P2_k = P2_{k-1} \cdot sty_k$$
$$P2_0 = 1$$
$$sty_k = y^k$$
$$sty_k = sty_{k-1} \cdot y$$
$$sty_0 = 1$$
$$bin_k = \binom{2k}{k}$$
$$bin_k = bin_{k-1} \cdot \frac{4k-2}{k}$$
$$S1_k = \sum_{j=1}^{k} x^{2j} y^{2j^2} \prod_{p=1}^{j} y^p$$

$$S1_k = S1_{k-1} + cin2_k \cdot P2_k$$
$$S1_0 = 0$$
$$cin2_k = x^{2k} y^{2k^2}$$
$$cin2_k = cin2_{k-1} \cdot x^2 \cdot y^{4k-2}$$
$$cin2_k = cin2_{k-1} \cdot pcin1_k^2 \cdot x^2$$
$$cin2_0 = 1$$

Program

```
MODULE Suma0901;
(* Septembar 2001.  1. zadatak *)
FROM IO IMPORT WrStr, WrLn, WrReal, RdReal, RdCard;

PROCEDURE F(x, y : REAL; n : CARDINAL; VAR ok : BOOLEAN) : REAL;
  CONST
    ndgr = 1;
    nggr = 50;
    xgr = 1.0;
    ygr = 1.0;
  VAR
    S, cin1, pcin1, P1, P2, sty, fak, fak1, bin, S1, cin2, sty2 : REAL;
    k, j, n1 : CARDINAL;
    yna2, xna2, yna4 : REAL;
  BEGIN
    IF (ndgr <= n) AND (n <= nggr) AND (ABS(x) <= xgr) AND (ABS(y) <= ygr)
                  AND (y # 0.0) THEN
      ok := TRUE;
      S := 0.0; S1 := 0.0;
      P2 := 1.0; fak := 1.0; fak1 := 1.0; bin := 1.0; sty := 1.0; cin2 := 1.0;
      pcin1 := 1.0 / y; sty2 := 1.0 / (y * y); n1 := 2 * n + 3; cin1 := FLOAT(n1);
      yna2 := y * y; xna2 := x * x; yna4 := yna2 * yna2;
      FOR k := 1 TO n DO
        pcin1 := yna2 * pcin1;
        cin1 := FLOAT(n1 - k) * pcin1 * xna2 * cin1 / FLOAT(k + 1);

        fak := fak1;
```

```
        fak1 := FLOAT(k + 1) * fak;

        sty := y * sty;
        P2 := P2 * sty;

        bin := FLOAT(4 * k - 2) * bin / FLOAT(k);

        sty2 := yna4 * sty2;
        cin2 := xna2 * sty2 * cin2;
        S1 := S1 + cin2 * P2;

        P1 := 1.0;
        FOR j := 1 TO k + 1 DO
          P1 := P1 * fak1;
        END;

        S := S + cin1 * P1 / (P2 * (fak + bin * S1));
      END;
    ELSE
      ok := FALSE;
      S := 0.0;
    END;
    RETURN S;
  END F;


VAR
  x, y, rez : REAL;
  n : CARDINAL;
  ok : BOOLEAN;
BEGIN
  WrStr('Unesite x: ');
  x := RdReal(); WrLn;
  WrStr('Unesite y: ');
  y := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;

  rez := F(x, y, n, ok);
  IF ok THEN
    WrStr('F(x, y, n) = ');
```

```
    WrReal(rez, 8, 15);
  ELSE
    WrStr('Uneli ste nedozvoljene argumente');
  END;
END Suma0901.
```

Objašnjenje

Isto kao i prethodni.

2 Polinomi

2.1 Procedure

U ovom odeljku će biti opisane procedure koje se najčešće sreću u ispitnim zadacima. Oni su podeljeni u 4 grupe:

PolinomN.mod - nalaze se procedure za rad sa realnim brojevima, one su samo ispisane, a njihova objašnjenja se nalaze u knjizi.

PolinomK.mod - nalaze se procedure za rad sa kompleksnim brojevima, kao i objašnjenja kako su procedure napravljene.

PolinomR.mod - nalaze se procedure za rad sa racionalnim brojevima, kao i objašnjenja kako su procedure napravljene.

ProcPoli.mod - nalaze se ostale procedure koje se često sreću u ispitnim zadacima.

2.1.1 PolinomN.def

```
DEFINITION MODULE PolinomN;
FROM IO IMPORT WrStr, WrLn, WrCard, WrReal, WrInt, RdReal, RdCard;

CONST
  maxSt = 100;

TYPE
  Polinom = RECORD
              k : ARRAY[0..maxSt] OF REAL;
              st : [-1..maxSt];
            END;
```

```
PROCEDURE Anuliraj(VAR p : Polinom);

PROCEDURE Izracunaj(x : REAL; p : Polinom; VAR rez : REAL);

PROCEDURE Stampaj(p : Polinom; d, h : CARDINAL);

PROCEDURE Stampaj2(p : Polinom; d, h : CARDINAL);

PROCEDURE Saberi(p1, p2 : Polinom; VAR zbir : Polinom);

PROCEDURE Oduzmi(p1, p2 : Polinom; VAR raz : Polinom);

PROCEDURE Puta(p1, p2 : Polinom; VAR proiz : Polinom; VAR ok :
BOOLEAN);

PROCEDURE BrojPuta(c : REAL; p : Polinom; VAR rez : Polinom);

PROCEDURE Deli(p1, p2 : Polinom; VAR kol, ost : Polinom; VAR ok :
BOOLEAN);

END PolinomN.
```

2.1.2 PolinomN.mod

```
IMPLEMENTATION MODULE PolinomN;
(* procedure za rad sa realnim brojevima *)

PROCEDURE Anuliraj(VAR p : Polinom);
VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    st := -1;
    FOR i := 0 TO maxSt DO
      k[i] := 0.0
    END
  END
END Anuliraj;
```

```
PROCEDURE NadjiStepen(VAR p : Polinom);
CONST
  eps = 1.E-5;
BEGIN
  WITH p DO
    st := maxSt;
    WHILE (st > -1) AND (ABS(k[st] - 0.0) < eps) DO
      DEC(st)
    END
  END
END NadjiStepen;

PROCEDURE Izracunaj(x : REAL; p : Polinom; VAR rez : REAL);
VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    IF st = -1 THEN
      rez := 0.0
    ELSE
      rez := k[st];
      FOR i := st - 1 TO 0 BY -1 DO
        rez := rez * x + k[i]
      END
    END
  END
END Izracunaj;

PROCEDURE Stampaj(p : Polinom; d, h : CARDINAL);
VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    IF st > 0 THEN
      IF k[st] < 0.0 THEN
        WrStr(' - ')
      END;
      IF ABS(k[st]) <> 1.0 THEN
        WrReal(ABS(k[st]), d, h)
      END;
      IF st > 1 THEN
```

```
        WrStr('x^'); WrCard(st, 1)
      ELSE
        WrStr('x ')
      END;

      FOR i := st - 1 TO 1 BY -1 DO
        IF k[i] <> 0.0 THEN
          IF k[i] > 0.0 THEN
            WrStr(' + ')
          ELSE
            WrStr(' - ')
          END;
          IF ABS(k[i] <> 1.0) THEN
            WrReal(ABS(k[i]), d, h)
          END;
          IF i > 1 THEN
            WrStr('x^'); WrCard(i, 1)
          ELSIF i = 1 THEN
            WrStr('x ')
          END
        END
      END;

      IF k[0] <> 0.0 THEN
        IF k[0] > 0.0 THEN
          WrStr(' + '); WrReal(k[0], d, h)
        ELSE
          WrStr(' - '); WrReal(ABS(k[0]), d, h)
        END
      END
    ELSE
      IF k[0] < 0.0 THEN
        WrStr(' - '); WrReal(ABS(k[0]), d, h)
      ELSE
        WrReal(k[0], d, h)
      END
    END
  END;
  WrLn
END Stampaj;
```

```
PROCEDURE Stampaj2(p : Polinom; d, h : CARDINAL);
VAR
  i : CARDINAL;
BEGIN
  FOR i := p.st TO 0 BY -1 DO
    WrReal(p.k[i], d, h);
    WrStr('x^');
    WrCard(i, 1);
  END
END Stampaj2;

TYPE
  znak = (plus, minus);

PROCEDURE Sab(p1, p2 : Polinom; op : znak; VAR zbir : Polinom);
VAR
  i : CARDINAL;
BEGIN
  Anuliraj(zbir);
  IF p1.st > p2.st THEN
    zbir.st := p1.st
  ELSE
    zbir.st := p2.st
  END;
  WITH zbir DO
    IF op = plus THEN
      FOR i := 0 TO st DO
        k[i] := p1.k[i] + p2.k[i]
      END
    ELSE
      FOR i := 0 TO st DO
        k[i] := p1.k[i] - p2.k[i]
      END
    END
  END;
  NadjiStepen(zbir)
END Sab;

PROCEDURE Saberi(p1, p2 : Polinom; VAR zbir : Polinom);
BEGIN
  Sab(p1, p2, plus, zbir);
```

```
END Saberi;

PROCEDURE Oduzmi(p1, p2 : Polinom; VAR raz : Polinom);
BEGIN
  Sab(p1, p2, minus, raz);
END Oduzmi;

PROCEDURE BrojPuta(c : REAL; p : Polinom; VAR rez : Polinom);
VAR
  i : CARDINAL;
BEGIN
  Anuliraj(rez);
  WITH p DO
    IF (st <> -1) AND (c <> 0.0) THEN
      rez.st := st;
      FOR i := 0 TO st DO
        rez.k[i] := c * k[i]
      END
    END
  END
END BrojPuta;

PROCEDURE Puta(p1, p2 : Polinom; VAR proiz : Polinom; VAR ok :
BOOLEAN); VAR
  i, j : CARDINAL;
BEGIN
  ok := TRUE;
  Anuliraj(proiz);
  IF (p1.st <> -1) AND (p2.st <> -1) THEN
    i := p1.st + p2.st;
    IF i > maxSt THEN
      ok := FALSE
    ELSE
      WITH proiz DO
        st := i;
        FOR i := 0 TO p1.st DO
          FOR j := 0 TO p2.st DO
            k[i + j] := k[i + j] + p1.k[i] * p2.k[j]
          END
        END
      END
```

```
    END
  END
END Puta;

PROCEDURE Deli(p1, p2 : Polinom; VAR kol, ost : Polinom; VAR ok :
BOOLEAN);

VAR
  i, j, m, l : CARDINAL;
BEGIN
  IF p2.st = -1 THEN
    ok := FALSE
  ELSE
    ok := TRUE;
    Anuliraj(kol);
    ost := p1;
    IF p1.st >= p2.st THEN
      kol.st := p1.st - p2.st;
      FOR j := p1.st TO p2.st BY -1 DO
        i := j - CARDINAL(p2.st);
        kol.k[i] := ost.k[j] / p2.k[p2.st];
        FOR m := 0 TO p2.st DO
          l := m + i;
          ost.k[l] := ost.k[l] - kol.k[i] * p2.k[m]
        END
      END;
      NadjiStepen(ost)
    END
  END
END Deli;

END PolinomN.
```

2.1.3 PolinomK.def

```
DEFINITION MODULE PolinomK; FROM IO IMPORT WrStr, WrLn, WrCard,
WrReal, WrInt, RdReal, RdCard;

CONST
```

```
  maxSt = 100;

TYPE
  KompBroj = RECORD
               Re, Im : REAL
             END;
  KPolinom = RECORD
               k : ARRAY[0..maxSt] OF KompBroj;
               st : [-1..maxSt]
             END;

PROCEDURE Anuliraj(VAR p : KPolinom);

PROCEDURE Izracunaj(x : KompBroj; p : KPolinom; VAR rez :
KompBroj);

PROCEDURE Stampaj(p : KPolinom; d, h : CARDINAL);

PROCEDURE Saberi(p1, p2 : KPolinom; VAR zbir : KPolinom);

PROCEDURE Oduzmi(p1, p2 : KPolinom; VAR raz : KPolinom);

PROCEDURE Puta(p1, p2 : KPolinom; VAR proiz : KPolinom; VAR ok :
BOOLEAN);

PROCEDURE BrojPuta(c : KompBroj; p : KPolinom; VAR rez :
KPolinom);

PROCEDURE Deli(p1, p2 : KPolinom; VAR kol, ost : KPolinom; VAR ok
: BOOLEAN);

END PolinomK.
```

2.1.4 Kompleksni brojevi

$$z_1 = a_1 + b_1 \cdot i$$

$$z_2 = a_2 + b_2 \cdot i$$

$$Z = z_1 + z_2 = (a_1 + a_2) + (b_1 + b_2) \cdot i$$

$$R = z_1 - z_2 = (a_1 - a_2) + (b_1 - b_2) \cdot i$$

$$P = z_1 \cdot z_2 = (a_1 + b_1 \cdot i) \cdot (a_2 + b_2 \cdot i) = a_1 \cdot a_2 + a_1 \cdot b_2 \cdot i + a_2 \cdot b_1 \cdot i + b_1 \cdot b_2 \cdot i^2 = (a_1 \cdot a_2 - b_1 \cdot b_2) + (a_1 \cdot b_2 + a_2 \cdot b_1) \cdot i$$

$$K = \frac{z_1}{z_2} = \frac{a_1 + b_1 \cdot i}{a_2 + b_2 \cdot i} \cdot \frac{a_2 - b_2 \cdot i}{a_2 + b_2 \cdot i} = \frac{(a_1 \cdot a_2 + b_1 \cdot b_2) + (a_2 \cdot b_1 - a_1 \cdot b_2) \cdot i}{a_2^2 + b_2^2}$$

Ovim formulama su opisane procedure $SabKomp$, $OduKomp$, $PutaKomp$ i $DeliKomp$. Ostale procedure su veoma slične običnim, sa jedinom razlikom što se umesto sabiranja koristi sabiranje kompleksnih brojeva, oduzimanja koristi oduzimanje kompleksnih brojeva, itd. Takodje se prilikom proveravanja početnih uslova proveravaju i realni i imaginarni deo.

2.1.5 PolinomK.mod

```
IMPLEMENTATION MODULE PolinomK; (*Procedure za rad sa kompleksnim
brojevima*)

PROCEDURE SabKomp(k1, k2 : KompBroj) : KompBroj; VAR
  zbir : KompBroj;
BEGIN
  zbir.Re := k1.Re + k2.Re;
  zbir.Im := k1.Im + k2.Im;
  RETURN zbir
END SabKomp;

PROCEDURE OduKomp(k1, k2 : KompBroj) : KompBroj; VAR
  raz : KompBroj;
BEGIN
  raz.Re := k1.Re - k2.Re;
  raz.Im := k1.Im - k2.Im;
  RETURN raz
END OduKomp;

PROCEDURE PutaKomp(k1, k2 : KompBroj) : KompBroj; VAR
  proiz : KompBroj;
BEGIN
  proiz.Re := k1.Re * k2.Re - k1.Im * k2.Im;
  proiz.Im := k1.Re * k2.Im + k1.Im * k2.Re;
```

```
  RETURN proiz;
END PutaKomp;

PROCEDURE DeliKomp(k1, k2 : KompBroj) : KompBroj; VAR
  kol : KompBroj;
  imen : REAL;
BEGIN
  imen := k2.Re * k2.Re + k2.Im * k2.Im;
  kol.Re := (k1.Re * k2.Re + k1.Im * k2.Im) / imen;
  kol.Im := (k1.Im * k2.Re - k1.Re * k2.Im) / imen;
  RETURN kol;
END DeliKomp;

PROCEDURE StampajKomp(k : KompBroj; d, h : INTEGER); BEGIN
  WrStr('(');
  IF k.Re <> 0.0 THEN
    WrReal(k.Re, d, h);
  END;
  IF k.Im <> 0.0 THEN
    IF k.Im < 0.0 THEN
      WrStr(' - '); WrReal(ABS(k.Im), d, h); WrStr('i)');
    ELSE
      WrStr(' + '); WrReal(k.Im, d, h); WrStr('i)');
    END
  ELSE
    WrStr(')')
  END
END StampajKomp;

PROCEDURE Anuliraj(VAR p : KPolinom); VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    st := -1;
    FOR i := 0 TO maxSt DO
      k[i].Re := 0.0;
    k[i].Im := 0.0;
    END
  END
END Anuliraj;
```

```
PROCEDURE NadjiStepen(VAR p : KPolinom); CONST
  eps = 1.E-5;
BEGIN
  WITH p DO
    st := maxSt;
    WHILE (st > -1) AND (ABS(k[st].Re - 0.0) < eps) AND (ABS(k[st].Im - 0.0) < eps
      DEC(st)
    END
  END
END NadjiStepen;

PROCEDURE Izracunaj(x : KompBroj; p : KPolinom; VAR rez :
KompBroj); VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    IF st = -1 THEN
      rez.Re := 0.0;
    rez.Im := 0.0;
    ELSE
      rez := k[st];
      FOR i := st - 1 TO 0 BY -1 DO
        rez := SabKomp(PutaKomp(rez, x), k[i]);
      END
    END
  END
END Izracunaj;

PROCEDURE Stampaj(p : KPolinom; d, h : CARDINAL); VAR
  i : CARDINAL;
BEGIN
  FOR i := p.st TO 0 BY -1 DO
    StampajKomp(p.k[i], d, h);
    WrStr('x^');
    WrCard(i, 1);
  END
END Stampaj2;

TYPE
  znak = (plus, minus);
```

```
PROCEDURE Sab(p1, p2 : KPolinom; op : znak; VAR zbir : KPolinom);
VAR
  i : CARDINAL;
BEGIN
  Anuliraj(zbir);
  IF p1.st > p2.st THEN
    zbir.st := p1.st
  ELSE
    zbir.st := p2.st
  END;
  WITH zbir DO
    IF op = plus THEN
      FOR i := 0 TO st DO
        k[i] := SabKomp(p1.k[i], p2.k[i])
      END
    ELSE
      FOR i := 0 TO st DO
        k[i] := SabKomp(p1.k[i], p2.k[i])
      END
    END
  END;
  NadjiStepen(zbir)
END Sab;

PROCEDURE Saberi(p1, p2 : KPolinom; VAR zbir : KPolinom); BEGIN
  Sab(p1, p2, plus, zbir);
END Saberi;

PROCEDURE Oduzmi(p1, p2 : KPolinom; VAR raz : KPolinom); BEGIN
  Sab(p1, p2, minus, raz);
END Oduzmi;

PROCEDURE BrojPuta(c : KompBroj; p : KPolinom; VAR rez :
KPolinom); VAR
  i : CARDINAL;
BEGIN
  Anuliraj(rez);
  WITH p DO
    IF (st <> -1) AND (c.Re <> 0.0) AND (c.Im <> 0.0) THEN
      rez.st := st;
      FOR i := 0 TO st DO
```

```
        rez.k[i] := PutaKomp(c, k[i])
      END
    END
  END
END BrojPuta;

PROCEDURE Puta(p1, p2 : KPolinom; VAR proiz : KPolinom; VAR ok : BOOLEAN);
VAR
  i, j : CARDINAL;
BEGIN
  ok := TRUE;
  Anuliraj(proiz);
  IF (p1.st <> -1) AND (p2.st <> -1) THEN
    i := p1.st + p2.st;
    IF i > maxSt THEN
      ok := FALSE
    ELSE
      WITH proiz DO
        st := i;
        FOR i := 0 TO p1.st DO
          FOR j := 0 TO p2.st DO
            k[i + j] := SabKomp(k[i + j], PutaKomp(p1.k[i], p2.k[j]))
          END
        END
      END
    END
  END
END Puta;

PROCEDURE Deli(p1, p2 : KPolinom; VAR kol, ost : KPolinom; VAR ok
: BOOLEAN);
VAR
  i, j, m, l : CARDINAL;
BEGIN
  IF p2.st = -1 THEN
    ok := FALSE
  ELSE
    ok := TRUE;
    Anuliraj(kol);
    ost := p1;
    IF p1.st >= p2.st THEN
```

```
      kol.st := p1.st - p2.st;
      FOR j := p1.st TO p2.st BY -1 DO
        i := j - CARDINAL(p2.st);
        kol.k[i] := DeliKomp(ost.k[j], p2.k[p2.st]);
        FOR m := 0 TO p2.st DO
          l := m + i;
          ost.k[l] := OduKomp(ost.k[l], PutaKomp(kol.k[i], p2.k[m]))
        END
      END;
      NadjiStepen(ost)
    END
  END
END Deli;

END PolinomK.
```

2.1.6 PolinomR.def

```
DEFINITION MODULE PolinomR; FROM IO IMPORT WrStr, WrLn, WrCard,
WrReal, WrInt, RdReal, RdCard;

CONST
  maxSt = 100;
TYPE
  Razlomak = RECORD
               Bro, Ime : INTEGER;
             END;
  RPolinom = RECORD
               k : ARRAY[0..maxSt] OF Razlomak;
               st : [-1..maxSt]
             END;

PROCEDURE Anuliraj(VAR p : RPolinom);

PROCEDURE Izracunaj(x : Razlomak; p : RPolinom; VAR rez :
Razlomak);

PROCEDURE Stampaj(p : RPolinom; d : CARDINAL);
```

```
PROCEDURE Saberi(p1, p2 : RPolinom; VAR zbir : RPolinom);

PROCEDURE Oduzmi(p1, p2 : RPolinom; VAR raz : RPolinom);

PROCEDURE Puta(p1, p2 : RPolinom; VAR proiz : RPolinom; VAR ok :
BOOLEAN);

PROCEDURE BrojPuta(c : Razlomak; p : RPolinom; VAR rez :
RPolinom);

PROCEDURE Deli(p1, p2 : RPolinom; VAR kol, ost : RPolinom; VAR ok
: BOOLEAN);

END PolinomR.
```

2.1.7 Racionalni brojevi

$$q_1 = \frac{a_1}{b_1}$$
$$q_2 = \frac{a_2}{b_2}$$

$$Z = q_1 + q_2 = \frac{a_1 \cdot b_2 + a_2 \cdot b_1}{b_1 \cdot b_2}$$
$$R = q_1 - q_2 = \frac{a_1 \cdot b_2 - a_2 \cdot b_1}{b_1 \cdot b_2}$$
$$P = q_1 \cdot q_2 = \frac{a_1 \cdot a_2}{b_1 \cdot b_2}$$
$$k = \frac{q_1}{q_2} = \frac{a_1 \cdot b_2}{b_1 \cdot a_2}$$

Funkcija NZD vraća najveći zajednički delilac, a funkcija Skrati vraća isti razlomak ali sa prostim činiocima.

Ovim formulama su opisane procedure *SabRaz*, *OduRaz*, *PutaRaz* i *DeliRaz*. Ostale procedure su veoma slične običnim, sa jedinom razlikom što se umesto sabiranja koristi sabiranje racionalnih brojeva, oduzimanja koristi oduzimanje racionalnih brojeva, itd. Takodje se prilikom proveravanja početnih uslova proveravaju i brojilac i imenilac deo.

2.1.8 PolinomR.mod

```
IMPLEMENTATION MODULE PolinomR; (* procedure za rad sa racionalnim
brojevima *)

PROCEDURE NZD(x, y : INTEGER) : INTEGER; VAR
  pom : INTEGER;
BEGIN
  pom := x MOD y;
  IF pom = 0 THEN
    RETURN y
  ELSE
    RETURN NZD(y, pom)
  END
END NZD;

PROCEDURE Skrati(p : Razlomak) : Razlomak; VAR
  r : Razlomak;
  pom : INTEGER;
BEGIN
  pom := NZD(p.Bro, p.Ime);
  r.Bro := p.Bro DIV pom;
  r.Ime := p.Ime DIV pom;
  RETURN r;
END Skrati;

PROCEDURE SabRaz(r1, r2 : Razlomak) : Razlomak; VAR
  zbir : Razlomak;
BEGIN
  zbir.Bro := r1.Bro * r2.Ime + r1.Ime * r2.Bro;
  zbir.Ime := r1.Ime * r2.Ime;
  RETURN Skrati(zbir);
END SabRaz;

PROCEDURE OduRaz(r1, r2 : Razlomak) : Razlomak; VAR
  raz : Razlomak;
BEGIN
  raz.Bro := r1.Bro * r2.Ime - r1.Ime * r2.Bro;
  raz.Ime := r1.Ime * r2.Ime;
  RETURN Skrati(raz);
```

```
END OduRaz;

PROCEDURE PutaRaz(r1, r2 : Razlomak) : Razlomak; VAR
  proiz : Razlomak;
BEGIN
  proiz.Bro := r1.Bro * r2.Bro;
  proiz.Ime := r1.Ime * r2.Ime;
  RETURN Skrati(proiz);
END PutaRaz;

PROCEDURE DeliRaz(r1, r2 : Razlomak) : Razlomak; VAR
  kol : Razlomak;
BEGIN
  kol.Bro := r1.Bro * r2.Ime;
  kol.Ime := r1.Ime * r2.Bro;
  RETURN Skrati(kol)
END DeliRaz;

PROCEDURE StampajRaz(r : Razlomak; d : INTEGER); BEGIN
  WrInt(r.Bro, d); WrStr('/'); WrInt(r.Ime, d);
END StampajRaz;

PROCEDURE Anuliraj(VAR p : RPolinom); VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    st := -1;
    FOR i := 0 TO maxSt DO
      k[i].Bro := 0;
      k[i].Ime := 0;
    END
  END
END Anuliraj;

PROCEDURE NadjiStepen(VAR p : RPolinom);
BEGIN
  WITH p DO
    st := maxSt;
    WHILE (st > -1) AND (k[st].Bro <> 0) AND (k[st].Ime <> 0) DO
      DEC(st)
    END
```

```
  END
END NadjiStepen;

PROCEDURE Izracunaj(x : Razlomak; p : RPolinom; VAR rez :
Razlomak); VAR
  i : CARDINAL;
BEGIN
  WITH p DO
    IF st = -1 THEN
      rez.Bro := 0;
      rez.Ime := 0;
    ELSE
      rez := k[st];
      FOR i := st - 1 TO 0 BY -1 DO
        rez := SabRaz(PutaRaz(rez, x), k[i]);
      END
    END
  END
END Izracunaj;

PROCEDURE Stampaj(p : RPolinom; d : CARDINAL); VAR
  i : CARDINAL;
BEGIN
  FOR i := p.st TO 0 BY -1 DO
    StampajRaz(p.k[i], d);
    WrStr('x^');
    WrCard(i, 1);
  END
END Stampaj2;

TYPE
  znak = (plus, minus);

PROCEDURE Sab(p1, p2 : RPolinom; op : znak; VAR zbir : RPolinom);
VAR
  i : CARDINAL;
BEGIN
  Anuliraj(zbir);
  IF p1.st > p2.st THEN
    zbir.st := p1.st
  ELSE
```

```
    zbir.st := p2.st
  END;
  WITH zbir DO
    IF op = plus THEN
      FOR i := 0 TO st DO
        k[i] := SabRaz(p1.k[i], p2.k[i])
      END
    ELSE
      FOR i := 0 TO st DO
        k[i] := SabRaz(p1.k[i], p2.k[i])
      END
    END
  END;
  NadjiStepen(zbir)
END Sab;

PROCEDURE Saberi(p1, p2 : RPolinom; VAR zbir : RPolinom); BEGIN
  Sab(p1, p2, plus, zbir);
END Saberi;

PROCEDURE Oduzmi(p1, p2 : RPolinom; VAR raz : RPolinom); BEGIN
  Sab(p1, p2, minus, raz);
END Oduzmi;

PROCEDURE BrojPuta(c : Razlomak; p : RPolinom; VAR rez :
RPolinom); VAR
  i : CARDINAL;
BEGIN
  Anuliraj(rez);
  WITH p DO
    IF (st <> -1) AND (c.Bro <> 0) AND (c.Ime <> 0) THEN
      rez.st := st;
      FOR i := 0 TO st DO
        rez.k[i] := PutaRaz(c, k[i])
      END
    END
  END
END BrojPuta;

PROCEDURE Puta(p1, p2 : RPolinom; VAR proiz : RPolinom; VAR ok :
BOOLEAN); VAR
```

```
  i, j : CARDINAL;
BEGIN
  ok := TRUE;
  Anuliraj(proiz);
  IF (p1.st <> -1) AND (p2.st <> -1) THEN
    i := p1.st + p2.st;
    IF i > maxSt THEN
      ok := FALSE
    ELSE
      WITH proiz DO
        st := i;
        FOR i := 0 TO p1.st DO
          FOR j := 0 TO p2.st DO
            k[i + j] := SabRaz(k[i + j], PutaRaz(p1.k[i], p2.k[j]))
          END
        END
      END
    END
  END
END Puta;

PROCEDURE Deli(p1, p2 : RPolinom; VAR kol, ost : RPolinom; VAR ok
: BOOLEAN); VAR
  i, j, m, l : CARDINAL;
BEGIN
  IF p2.st = -1 THEN
    ok := FALSE
  ELSE
    ok := TRUE;
    Anuliraj(kol);
    ost := p1;
    IF p1.st >= p2.st THEN
      kol.st := p1.st - p2.st;
      FOR j := p1.st TO p2.st BY -1 DO
        i := j - CARDINAL(p2.st);
        kol.k[i] := DeliRaz(ost.k[j], p2.k[p2.st]);
        FOR m := 0 TO p2.st DO
          l := m + i;
          ost.k[l] := OduRaz(ost.k[l], PutaRaz(kol.k[i], p2.k[m]))
        END
      END;
```

```
      NadjiStepen(ost)
    END
  END
END Deli;

END PolinomR.
```

2.1.9 ProcPoli.def

```
DEFINITION MODULE ProcPoli; FROM PolinomN IMPORT Polinom,
Anuliraj, Izracunaj, Saberi, Oduzmi, BrojPuta, Puta, Deli; CONST
  maxSt = 100;
TYPE
  Nula = RECORD
           nula : REAL;
           br : CARDINAL;
         END;
  NizNula = ARRAY[0..maxSt] OF Nula;
  NizSume = ARRAY[0..maxSt] OF REAL;
  NizPolinoma = ARRAY[0..maxSt - 1] OF Polinom;

PROCEDURE Izvod(p : Polinom; VAR izv : Polinom);

PROCEDURE pODg(p1, p2 : Polinom; VAR rez : Polinom);

PROCEDURE Visestrukost(p : Polinom; VAR q : NizNula);

PROCEDURE SumaPP(p : Polinom; VAR q : NizSume);

PROCEDURE SumaPC(p : Polinom; a, b : REAL; VAR q : NizSume);

PROCEDURE SumaPO(p : Polinom; VAR q : NizSume);

PROCEDURE PodeliP(p : Polinom; VAR a : NizPolinoma; br :
CARDINAL);

END ProcPoli.
```

2.1.10 Pomoćne procedure

Procedura $Izvod$ pravi polinom koji je jednak izvodu polaznog polinoma:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$
$$P(x)' = n a_n x^{n-1} + (n-1) a_{n-1} x^{n-2} + \ldots + 2 a_2 x + a_1$$

Procedura $pODg$ pravi kompoziciju dve funkcije, odnosno dva polinoma:

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$
$$G(x) = b_m x^m + b_{m-1} x^{m-1} + \ldots + b_1 x + b_0$$

$$\begin{aligned} P(G(x)) = {} & a_n (b_m x^m + b_{m-1} x^{m-1} + \ldots + b_1 x + b_0)^n + \\ & + a_{n-1} (b_m x^m + b_{m-1} x^{m-1} + \ldots + b_1 x + b_0)^{n-1} + \ldots + \\ & + a_1 (b_m x^m + b_{m-1} x^{m-1} + \ldots + b_1 x + b_0) + a_0 \end{aligned} \tag{1}$$

Procedura $Nule$ pravi niz racionalnih nula datog polinoma. Prvo pronadje delilac slobodnog člana, a zatim vodećeg. Njihov količnik ubaci u polinom umesto x i proveri da li se kao rezultat dobija 0. Isto to ponovi sa negativnim količnikom.

Funkcija $Deljiv$ vraća TRUE ako su dva polinoma deljiva, a FALSE ako nisu. U oba slučaja vraća njihov količnik.

Procedura $Visestrukost$ odredjuje nule i njihovo višestruko pojavljivanje. Prvo odredi nule pomoću opisane procedure, a zatim odredjuje broj pojavljivanja tako što deli polinom sa $(x - rn)$, gde je rn svaka od racionalnih nula, dokle god je to moguće. Kao rezultat dobijamo niz nula i broja njihovih pojavljivanja u datom polinomu.

Procedura $InitP$ inicijalizuje promenljivi polinom xi, odnosno polinom koji zavisi od brojača i. U konkretnom slučaju, pravili smo polinom $(x + i)$.

Procedura $NapraviP$ pravi polinom xi na i, odnosno u našem slucaju $(x+i)^i$. Ova procedura se menja ako je stepen u zavisnosti od i.

Procedura $SumaPP$ rešava problem:

$$P(x) = \sum_{i=0}^{st} a_i (x + i)^i$$

U parametrima poziva procedure $InitP$ se definiše osnova. U opštem slučaju ona je $ax + b$, ali zamenom $a = 1$ i $b = i$ rešavamo ovaj konkretan problem. Ova procedura se koristi samo u slucajevima kada je osnova promenljiva, odnosno zavisi od brojača i.

Ako se stepen polinoma u osnovi poveća, ovoj proceduri su potrebni mali dodaci. Ako je stepen 2, onda će se tražiti dva niza a_i i b_i gde će se jedan nalaziti u $kol.k[0]$, a drugi u $kol.k[1]$. Za veće stepene, važe analogna proširenja. Takodje, poveća ce se broj parametara u proceduri $InitP$ za jedan, jer je sada opšti oblik osnove $ax^2 + bx + c$.

Procedura *InitC* pravi polinom koji zavisi samo od konstantnih faktora, u ovom slučaju to su a i b, koji su unapred zadati. Ova procedura pravi $(ax+b)^st$ odnosno pravi polinom koji je istog stepena kao polinom p od kojeg se pravi suma. Da je osnova bila 2, stepen bi bio $stDIV2$, itd.

Procedura *SumaPC* rešava problem:

$$P(x) = \sum_{i=0}^{st} a_i(ax+b)^i$$

Ova procedura ne podleže nikakve promene. Sve se menja u proceduri *InitC* ukoliko su osnova i stepen različiti od navedenih. Jedino ako se stepen polinoma u osnovi poveća, onda su potrebni mali dodaci, koji su isti kao u proceduri *SumaPP*

Procedura *SumaPO* rešava problem:

$$P(x) = \sum_{i=0}^{st} a_i(x+i)^i + a_i^2$$

Ova procedura računa a_i tako što izjednači vodeće članove, a zatim oduzimanjem dobija novi polinom koji ima manji stepen, a koeficijente računa na isti način.

Procedura *PodeliP* rešava problem:

$$P(x) = A_1(x^k) + xA_2(x^k) + \ldots + x^{k-1}A_k(x^k)$$

gde je k neki zadati broj, a A_i nizovi polinoma koje treba izračunati. Ukoliko je neko A_i pomnoženo još sa nekim brojem K, onda se na kraju procedure taj ceo polinom podeli sa tim brojem, odnosno napiše se:

```
BrojPuta(1.0 / K, A[i], A[i])
```

2.1.11 ProcPoli.mod

```
IMPLEMENTATION MODULE ProcPoli; (* pomocne procedure za resavanje
zadataka *)

PROCEDURE Izvod(p : Polinom; VAR izv : Polinom); VAR
  i : CARDINAL;
BEGIN
  Anuliraj(izv);
  IF p.st <> -1 THEN
    WITH izv DO
      st := p.st - 1;
```

```
      FOR i := p.st TO 1 BY -1 DO
        k[i - 1] := FLOAT(i) * p.k[i]
      END
    END
  END
END Izvod;


PROCEDURE pODg(p1, p2 : Polinom; VAR rez : Polinom); VAR
  ok : BOOLEAN;
  i : CARDINAL;
  sab, stp : Polinom;

BEGIN
  Anuliraj(rez);
  WITH rez DO
    st := 0; k[0] := p1.k[0]
  END;
  Anuliraj(stp);
  WITH stp DO
    st := 0; k[0] := 1.0
  END;

  FOR i := 1 TO p1.st DO
    Puta(stp, p2, stp, ok);
    BrojPuta(p1.k[i], stp, sab);
    Saberi(rez, sab, rez)
  END
END pODg;

PROCEDURE Nule(p : Polinom; VAR q : NizNula); VAR
  a, b, br : CARDINAL;
  rn, rez : REAL;
BEGIN
  br := 0; a := 1;
  WHILE a <= TRUNC(p.k[0]) DO
    IF TRUNC(p.k[0]) MOD a = 0 THEN
      b := 1;
      WHILE b <= TRUNC(p.k[p.st]) DO
        IF TRUNC(p.k[p.st]) MOD b = 0 THEN
          rn := FLOAT(a) / FLOAT(b);
```

```
          Izracunaj(rn, p, rez);
          IF rez = 0.0 THEN
            INC(br);
            q[br].nula := rn;
          END;
          Izracunaj(-rn, p, rez);
          IF rez = 0.0 THEN
            INC(br);
            q[br].nula := -rn;
          END;
        END;
        INC(b);
      END
    END;
    INC(a);
  END;
  q[0].br := br;
END Nule;

PROCEDURE Deljiv(p1, p2 : Polinom; VAR kol : Polinom) : BOOLEAN;
VAR
  ok : BOOLEAN;
  ost : Polinom;
BEGIN
  Deli(p1, p2, kol, ost, ok);
  IF ost.st = -1 THEN
    RETURN TRUE
  ELSE
    RETURN FALSE
  END
END Deljiv;

PROCEDURE Visestrukost(p : Polinom; VAR q : NizNula); VAR
  i : CARDINAL;
  pnula, kol : Polinom;
BEGIN
  Nule(p, q);
  FOR i := 1 TO q[0].br DO
    q[i].br := 0;
    Anuliraj(pnula);
    WITH pnula DO
```

```
      pnula.st := 1; pnula.k[0] := -q[i].nula; pnula.k[1] := 1.0;
    END;
    LOOP
      IF Deljiv(p, pnula, kol) THEN
        INC(q[i].br);
        p := kol
      ELSE
        EXIT;
      END
    END
  END
END Visestrukost;

PROCEDURE InitP(VAR xi : Polinom; a, b : REAL);
BEGIN
  Anuliraj(xi);
  WITH xi DO
    st := 1; k[st] := a; k[0] := b;
  END;
END InitP;

PROCEDURE NapraviP(VAR stp : Polinom; xi : Polinom; i : CARDINAL);
VAR
  j : CARDINAL;
  ok : BOOLEAN;
BEGIN
  Anuliraj(stp);
  WITH stp DO
    st := 0; k[0] := 1.0;
  END;
  FOR j := 1 TO i DO
    Puta(stp, xi, stp, ok)
  END;
END NapraviP;

PROCEDURE SumaPP(p : Polinom; VAR q : NizSume); VAR
  xi, ost, kol, stp : Polinom;
  i : CARDINAL;
  ok : BOOLEAN;
  a, b : REAL;
BEGIN
```

```
  ost := p; a := 1.0; b := 1.0;
  FOR i := p.st TO 0 BY -1 DO
    InitP(xi, 1.0, FLOAT(i));
    NapraviP(stp, xi, i);
    Deli(ost, stp, kol, ost, ok);
    q[i] := kol.k[0]
  END
END SumaPP;

PROCEDURE InitC(p : Polinom; a, b : REAL; VAR xc, stp : Polinom);
VAR
  i : CARDINAL;
  ok : BOOLEAN;
BEGIN
  Anuliraj(xc);
  WITH xc DO
    xc.st := 1; xc.k[0] := b; xc.k[1] := a;
  END;
  Anuliraj(stp);
  WITH stp DO
    stp.st := 0; stp.k[0] := 1.0;
  END;
  FOR i := 1 TO p.st DO
    Puta(stp, xc, stp, ok)
  END
END InitC;

PROCEDURE SumaPC(p : Polinom; a, b : REAL; VAR q : NizSume); VAR
  xc, ost, kol, stp, pom : Polinom;
  i : CARDINAL;
  ok : BOOLEAN;
BEGIN
  InitC(p, a, b, xc, stp);
  ost := p;
  FOR i := p.st TO 0 BY -1 DO
    Deli(ost, stp, kol, ost, ok);
    Deli(stp, xc, stp, pom, ok);
    q[i] := kol.k[0];
  END
END SumaPC;
```

```
PROCEDURE SumaPO(p : Polinom; VAR q : NizSume); VAR
  xi, ost, stp : Polinom;
  i : CARDINAL;
BEGIN
  InitP(xi);
  ost := p;
  FOR i := p.st TO 0 BY -1 DO
    q[i] := ost.k[ost.st];
    NapraviP(stp, xi, i);
    BrojPuta(q[i], stp, stp);
    stp.k[0] := stp.k[0] + q[i] * q[i];
    Oduzmi(ost, stp, ost);
  END
END SumaPO;

PROCEDURE InitNiz(VAR a : NizPolinoma; p : Polinom; br :
CARDINAL); VAR
  i : CARDINAL;
BEGIN
  FOR i := 0 TO br - 1 DO
    Anuliraj(a[i]);
    a[i].st := CARDINAL(p.st) DIV br;
  END
END InitNiz;

PROCEDURE PodeliP(p : Polinom; VAR a : NizPolinoma; br :
CARDINAL); VAR
  i : CARDINAL;
BEGIN
  InitNiz(a, p, br);
  FOR i := 0 TO p.st DO
    a[i MOD br].k[i DIV br] := p.k[i]
  END
END PodeliP;

END ProcPoli.
```

2.2 Zadaci

2.2.1 1. Zadatak - Januar 1998.

Text

Napisati program koji izračunava koeficijente polinoma $a(x)$ i $b(x)$ ako važi:

$$f_17(x) = f_9(x)a(x) + b(x)$$

i stepen polinoma $b(x)$ je manji od stepena polinoma $f_9(x)$. Niz polinoma f_n je definisan na sledeći način:

$$f_n(x) = x^2 f_{n-1}(x) + x(f'_{n-1}(x) - f''_{n-2}(x)) + x f^2_{n-2}(x), \qquad n > 1$$

$$f_0(x) = 4, \qquad f_1(x) = 1 - x$$

Program

```
MODULE Poli0198; FROM PolinomN IMPORT Anuliraj, Oduzmi, Saberi,
                     Puta, Deli, Stampaj;
FROM PoliProc IMPORT Izvod;

VAR
  i : CARDINAL;
  px, pxna2, pom1, pom2, pom3, a, b, f, f1, f2, f9 : polinom;
  ok : BOOLEAN;
  BEGIN
    Anuliraj(px);
    px.st := 1; px.k[1] := 1.0;
    Anuliraj(pxna2);
    pxna2.st := 2; pxna2.k[2] := 1.0;
    Anuliraj(f1); Anuliraj(f2);
    f2.st := 0; f2.k[0] := 4.0;
    f1.st := 1; f1.k[0] := 1.0; f1.k[1] := -1.0;
    FOR i := 2 TO 17 DO
      Izvod(f1, pom1);
      Izvod(f2, pom3);
      Izvod(pom3, pom2);
```

```
    Oduzmi(pom1, pom2, pom3);
    Puta(px, pom3, ok, pom1);
    Puta(pxna2, f1, ok, pom2);
    Saberi(pom1, pom2, pom3);
    Puta(f2, f2, ok, pom1);
    Puta(px, pom1, ok, pom2);
    Saberi(pom3, pom2, f);
    f2 := f1; f1 := f;
    IF i = 9 THEN
      f9 := f
    END;
  END;
  Deli(f, f9, a, b, ok);

  Stampaj(a, 5, 8);
  Stampaj(b, 5, 8);

END Poli0198.
```

Objašnjenje

Jedan jednostavan zadatak za zagrevanje. Polinom koji je rekurentno definisan, kao prvi deo zadatka, a jedno obično deljenje kao drugi deo.

2.2.2 2. Zadatak - April 1997.

Text

Niz polinoma je definisan na sledeći način:

$$p_1(x) = 2x - 7$$

$$p_n(x) = p'_{n-1}(x)(2x^2 + 3x - 17) + 13p_{n-1}(x) - 18, \qquad n > 1$$

Naći i prikazati polinome $q_1(x)$ i $q_2(x)$ za koje važi jednačina:

$$(x^3 - 1)p_10(x) = (x^4 + x^3 + x^2 + x)q_1(x) + q_2(x)$$

gde je stepen polinoma $q_2(x)$ manji od 4.

Program

```
MODULE Poli0497; FROM PolinomN IMPORT Polinom, Anuliraj, BrojPuta,
                      Puta, Saberi, Deli, Stampaj;
FROM PoliProc IMPORT Izvod;

CONST
  maxN = 10;
VAR
  p1, p, stx2, stx3, stx4, izv, sab1, sab2, rez, q1, q2 : Polinom;
  i : CARDINAL;
  ok : BOOLEAN;
BEGIN
  Anuliraj(p1);
  WITH p1 DO
    st := 1; k[0] := -7.0; k[1] := 2.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[0] := -17.0; k[1] := 3.0; k[2] := 2.0;
  END;
  Anuliraj(stx3);
  WITH stx3 DO
    st := 3; k[0] := -1.0; k[3] := 1.0;
  END;
  Anuliraj(stx4);
  WITH stx4 DO
    st := 4; k[1] := 1.0; k[2] := 1.0; k[3] := 1.0; k[4] := 1.0;
  END;
  FOR i := 2 TO maxN DO
    Izvod(p1, izv);
    Puta(izv, stx2, sab1, ok);
    BrojPuta(13.0, p1, sab2);
    Saberi(sab1, sab2, p);
    p.k[0] := p.k[0] - 18.0;
    p1 := p;
  END;
  Puta(stx3, p, rez, ok);
  Deli(rez, stx4, q1, q2, ok);
  Stampaj(q1, 5, 8);
```

```
  Stampaj(q2, 5, 8);
END Poli0497.
```

Objašnjenje

Zadatak je veoma sličan prethodnom. U drugom delu moramo napraviti polinome koji su deljenik i delilac, dok smo tamo već imali.

2.2.3 3. Zadatak - Maj 1998.

Text

Napisati program koji učitava realne brojeve a i b i izračunava sve polinome $p_i(x)$, $0 <= i <= 50$, ako se p_i definiše na sledeći način:

$$p_i(x) = \frac{x^3}{3} + a, \qquad \text{ako je } i = 0$$

$$p_i(x) = 2x - 6b, \qquad \text{ako je } i = 1$$

$$p_i(x) = p_{i-2}(x)(ax + 3) - p_{i-1}(b)(x^3 + 2b + 3), \qquad \text{ako je } i \text{ neparno}$$

$$p_i(x) = p_{i-1}(i)((ap_{i-1}(x))MODp_{i-2}(x)), \qquad \text{ako je } i \text{ parno}$$

Oznakom $p\ MOD\ q$ je predstavljena operacija ostatak po modulu pri deljenju polinoma p polinomom q.

Program

```
MODULE Poli0598; FROM IO IMPORT WrStr, WrLn, RdReal, RdCard; FROM
PolinomN IMPORT Polinom, Anuliraj, Oduzmi, Izracunaj,
                     BrojPuta, Deli, Stampaj;
FROM PoliProc IMPORT pODg;

VAR
  p0, p1, p, stx1, stx3, cin, kol, ost, sab1, sab2 : Polinom;
  a, b, rez : REAL;
  n, i : CARDINAL;
  ok : BOOLEAN;
BEGIN
```

```
  WrStr('Unesite a: ');
  a := RdReal(); WrLn;
  WrStr('Unesite b: ');
  b := RdReal(); WrLn;
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;
  Anuliraj(p0);
  WITH p0 DO
    st := 3; k[0] := a; k[3] := 1.0 / 3.0;
  END;
  Anuliraj(p1);
  WITH p1 DO
    st := 1; k[0] := -6.0 * b; k[1] := 2.0;
  END;
  Anuliraj(stx1);
  WITH stx1 DO
    st := 1; k[0] := 3.0; k[1] := a;
  END;
  Anuliraj(stx3);
  WITH stx3 DO
    st := 3; k[0] := 3.0; k[1] := 2.0 * b; k[3] := 1.0;
  END;
  FOR i := 2 TO n DO
    IF i MOD 2 = 0 THEN
      Izracunaj(FLOAT(i), p1, rez);
      BrojPuta(a, p1, cin);
      Deli(cin, p0, kol, ost, ok);
      BrojPuta(rez, ost, p);
    ELSE
      pODg(p0, stx1, sab1);
      Izracunaj(b, p1, rez);
      BrojPuta(rez, stx3, sab2);
      Oduzmi(sab1, sab2, p);
    END;
    p0 := p1; p1 := p;
  END;
  Stampaj(p, 5, 8);
END Poli0598.
```

Objašnjenje

Ovde je potrebno razgraničiti slučajeve kada je brojač paran i neparan, pa onda odredjene naredbe izvršavati.

Takodje, $p\ MOD\ q$ operacijom kod polinoma nam je u stvari potreban ostatak korišćenjem procedure $Deli$.

2.2.4 4. Zadatak - Avgust 2000.

Text

Prvih trideset članova niza polinoma p se definiše na sledeći način:

$p_1(x) = x$
$p_2(x) = x^2 - 1$
$p_i(x) = (3x + 2)p_{i-1}(x) - 2xp_{i-2}(x), \qquad i = 3, \quad 4, \quad \ldots, \quad 30$

Napisati program koji učitava broj $k(1 <= k <= 30)$ i koji pronalazi i ispisuje koeficijente $a_0, a_1, \ldots, a_{k-1}$ i polinom $q(x)$ za koje važi relacija:

$$p_{30}(x) = q(x)(x - k)^k + \sum_{i=0}^{k-1} a_i(x - i)^i$$

Program

```
MODULE Poli0800; FROM IO IMPORT WrStr, WrLn, RdCard; FROM PolinomN
IMPORT Polinom, Anuliraj, Oduzmi,
                     Puta, Deli, Stampaj;
FROM PoliProc IMPORT NizSume, SumaPP;

CONST
  maxN = 30;
VAR
  b : NizSume;
  p1, p2, p, stx1, stx2, stxj, pol, q, r, sab1, sab2 : Polinom;
```

```
  j, i : CARDINAL;
  ok : BOOLEAN;
BEGIN
  Anuliraj(p1);
  WITH p1 DO
    st := 1; k[1] := 1.0;
  END;
  WITH p2 DO
    st := 2; k[0] := -1.0; k[2] := 1.0;
  END;
  Anuliraj(stx1);
  WITH stx1 DO
    st := 1; k[0] := 2.0; k[1] := 3.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 1; k[1] := 2.0;
  END;
  FOR i := 3 TO maxN DO
    Puta(stx1, p2, sab1, ok);
    Puta(stx2, p1, sab2, ok);
    Oduzmi(sab1, sab2, p);
    p1 := p2; p2 := p;
  END;
  WrStr('Unesite j: ');
  j := RdCard();
  Anuliraj(stxj);
  WITH stxj DO
    st := 1; k[0] := -FLOAT(j); k[1] := 1.0;
  END;
  Anuliraj(pol);
  WITH pol DO
    st := 0; k[0] := 1.0;
  END;
  FOR i := 1 TO j DO
    Puta(pol, stxj, pol, ok);
  END;
  Deli(p, pol, q, r, ok);
  SumaPP(r, b);
  Stampaj(q, 5, 8);
END Poli0800.
```

Objašnjenje

Kada izračunamo $p_30(x)$ i napravimo $(x-k)^k$, korišćenjem procedure *Deli* dobijamo $q(x)$ i $r(x)$. Ovo drugo nam je upravo potrebna suma. Za to imamo već napravljenu proceduru *SumaPP*, jer je osnova promenljiva, koju je potrebno samo malo izmeniti, jer je osnova drugačija od one za koju je procedura napisana.

$$ax + b = x - i$$

$$a = 1 \wedge b = -i$$

2.2.5 5. Zadatak - Januar 2000

Text

Niz polinoma P je dat na sledeći način:

$P_1(x) = x^2 - 3x + 7,$
$P_2(x) = 4x^3 + 2,$
$P_k(x) = (x - 3x^2 + 4)P'_{k-2}(x) - P_{k-1}(k-1)(2x-3)P_{k-1}(x), \qquad k > 2.$

Napisati program koji učitava broj $n(1 <= n <= 20)$, izračunava polinom P_n i pronalazi i štampa koeficijente $a_i(0 <= i <= st)$ za koje važi:

$P_n(x) = \sum_{i=0}^{st} a_i(x+1)^{st-i}$, gde je st stepen polinoma $P_n(x)$.

Program

```
MODULE Poli0100;
FROM IO IMPORT WrStr, WrLn, RdInt;
FROM PolinomN IMPORT Polinom, Anuliraj, Puta, Izracunaj,
                     BrojPuta, Oduzmi;
FROM PoliProc IMPORT NizSume, Izvod, SumaPC;

VAR
```

```
  p1, p2, p, izv, stx1, stx2, sab1, sab2 : Polinom;
  rez1, rez2, a, b : REAL;
  k, n : CARDINAL;
  ok : BOOLEAN;
  q : NizSume;

BEGIN
  WrStr('Unesite n: ');
  n := RdInt();
  Anuliraj(p1);
  WITH p1 DO
    st := 2; k[0] := 7.0; k[1] := -3.0; k[2] := 1.0;
  END;
  Anuliraj(p2);
  WITH p2 DO
    st := 3; k[0] := 2.0; k[3] := 4.0;
  END;
  Anuliraj(stx1);
  WITH stx1 DO
   st := 1; k[0] := 3.0; k[1] := 2.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
   st := 2; k[0] := 4.0; k[1] := 1.0; k[2] := -3.0;
  END;
  FOR k := 3 TO n DO
    Izvod(p1, izv);
    Puta(stx2, izv, sab1, ok);
    Izracunaj(FLOAT(k), p2, rez1);
    Izracunaj(FLOAT(k - 1), p2, rez2);
    Puta(stx1, p2, sab2, ok);
    BrojPuta(rez1 * rez2, sab2, sab2);
    Oduzmi(sab1, sab2, p);
    p1 := p2; p2 := p;
  END;

  a := 1.0; b := 1.0;
  SumaPC(p, a, b, q);
END Poli0100.
```

Objašnjenje

Prvi deo je tipski skoro uvek ovakav, a u drugom delu se ovom prilikom računa suma. Pošto je osnova konstantna, koristićemo $SumaPC$ proceduru.

$$ax + b = x + 1$$

$$a = 1 \wedge b = 1$$

Ostalo je još samo stepen da sredimo. Pošto je stepen $st - i$, u našoj proceduri umesto reda:

```
FOR i := p.st TO 0 BY - 1 DO
```

treba da stoji:

```
FOR i := 0 TO p.st BY - 1 DO
```

2.2.6 6. Zadatak - Januar 2001.

Text

Niz polinoma se definiše na sledeći način:

$P_n(x) = (x - 3x^2)P_{n-1}(x) + P_{n-1}(1)P_{n-2}(x)(x - 3)$
$P_1(x) = 1 + x^2$
$P_2(x) = 4 + 5x + x^3$

Za uneto n, naći polinom $Q(x)$ i koeficijente a_i za koje važi jednakost:

$$P_n(x) = (P_{n\ DIV\ 8}(x))^8 Q(x) + \sum_{i=0}^{st} a_i(x + i)^i$$

Program

```
MODULE Poli0101;
FROM IO IMPORT WrStr, WrLn, RdCard;
FROM PolinomN IMPORT Polinom, Anuliraj, Izracunaj, BrojPuta,
                     Puta, Saberi, Deli, Stampaj;
```

```
FROM PoliProc IMPORT NizSume, Izvod, SumaPP;

VAR
  b : NizSume;
  p1, p2, p8, p, stx1, stx2, sab1, sab2, izv, q, r : Polinom;
  i, n, h : CARDINAL;
  rez : REAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite n: ');
  n := RdCard();
  Anuliraj(p1);
  WITH p1 DO
    st := 2; k[0] := 1.0; k[2] := 1.0;
  END;
  Anuliraj(p2);
  WITH p2 DO
    st := 3; k[0] := 4.0; k[1] := 5.0; k[3] := 1.0;
  END;
  Anuliraj(stx1);
  WITH stx1 DO
    st := 1; k[0] := -3.0; k[1] := 1.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[1] := 1.0; k[2] := -3.0;
  END;
  h := n DIV 8;
  FOR i := 3 TO n DO
    Izvod(p2, izv);
    Puta(stx2, izv, sab1, ok);
    Puta(p1, stx1, sab2, ok);
    Izracunaj(1.0, p2, rez);
    BrojPuta(rez, sab2, sab2);
    Saberi(sab1, sab2, p);
    p1 := p2; p2 := p;
    IF i = h THEN
      p8 := p;
    END;
  END;
```

```
    Puta(p8, p8, p8, ok);
    Puta(p8, p8, p8, ok);
    Puta(p8, p8, p8, ok);
    Deli(p, p8, q, r, ok);
    Stampaj(q, 5, 8);
    SumaPP(r, b);
  END Poli0101.
```

Objašnjenje

Pošto znamo koliko je n, onda znamo i koliko je $nDIV8$. Dok iterativno računamo P_i, kada naidjemo na taj broj, jednostavno zapamtimo taj polinom. Stepenujemo ga na 8, pa kada podelimo P_n sa ovim polinom, kao količnik dobijamo $Q(x)$, a kao ostatak $R(x)$, koji jednostavno poznatom procedurom pretvaramo u zadatu sumu.

2.2.7 7. Zadatak - Februar 2001.

Text

Niz polinoma p_n se definiše na sledeći način:

$p_1(x) = x^2 - 7x + 12$
$p_2(x) = x^3 + 1$
$p_k(x) = p_{k-1}(4)p'_{k-1}(x) + p_{k-2}(x)(x^2 - 8x + 3)$

Za dato k, naći polinome $A(x)$, $B(x)$ i $C(x)$ tako da je zadovoljena sledeća relacija:

$$p_k(x) = A(x^3) + xB(x^3) + x^2C(x^3)$$

Program

```
MODULE Poli0201; FROM IO IMPORT WrStr, WrLn, RdCard; FROM PolinomN
IMPORT Polinom, Anuliraj, Puta, BrojPuta,
                     Izracunaj, Saberi, Stampaj;
FROM PoliProc IMPORT NizPolinoma, Izvod, PodeliP;
```

```
VAR
  q : NizPolinoma;
  p1, p2, p, stx2, sab1, sab2, izv : Polinom;
  rez : REAL;
  k, i : CARDINAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite k: ');
  k := RdCard();
  Anuliraj(p1);
  WITH p1 DO
    st := 2; k[0] := 12.0; k[1] := -7.0; k[2] := 1.0;
  END;
  Anuliraj(p2);
  WITH p2 DO
    st := 3; k[0] := 1.0; k[3] := 1.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[0] := 3.0; k[1] := -8.0; k[2] := 1.0;
  END;
  FOR i := 3 TO k DO
    Izvod(p2, izv);
    Izracunaj(4.0, p2, rez);
    BrojPuta(rez, izv, sab1);
    Puta(p1, stx2, sab2, ok);
    Saberi(sab1, sab2, p);
    p1 := p2; p2 := p;
  END;
  PodeliP(p, q, 3);
  Stampaj(q[0], 5, 8);
  Stampaj(q[1], 5, 8);
  Stampaj(q[2], 5, 8);
END Poli0201.
```

Objašnjenje

Za drugi deo zadatka je potrebna samo procedura *PodeliP* koju smo ranije detaljno opisali.

2.2.8 8. Zadatak - Novembar 2000.

Text

Nizovi polinoma T_n i Q_n se definišu na sledeći način:

$T_1 = x + 2$
$T_n = 2xQ_n - (6 - x)T_{n-1}, \qquad n > 1$
$Q_1 = 3x - 2$
$Q_n = (2x^2 + 5)T'_{n-1} + (x - 1)Q_{n-1}, \qquad n > 1$

Napisati program koji za uneto n veće od 1 pronalazi i ispisuje polinome $A(x)$ i $B(x)$ za koje važi jednačina:

$T_n = nA(x^2) + (n - 1)B(x^2)$

Program

```
MODULE Poli1100; FROM IO IMPORT WrStr, WrLn, RdInt; FROM PolinomN
IMPORT Polinom, Anuliraj, Puta, Saberi,
                     Oduzmi, BrojPuta, Stampaj;
FROM PoliProc IMPORT NizPolinoma, Izvod, PodeliP;

VAR
  a : NizPolinoma;
  t1, t, q1, q, stx1, stx2, stx3, stx4, sab1, sab2, izv : Polinom;
  i, n : CARDINAL;
  ok : BOOLEAN;
BEGIN
  WrStr('Unesite n: ');
  n := RdInt();
  Anuliraj(t1);
  WITH t1 DO
    st := 1; k[0] := 2.0; k[1] := 1.0;
  END;
  Anuliraj(q1);
  WITH q1 DO
    st := 1; k[0] := -2.0; k[1] := 3.0;
  END;
```

```
  Anuliraj(stx1);
  WITH stx1 DO
    st := 1; k[1] := 2.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 1; k[0] := 6.0; k[1] := -1.0;
  END;
  Anuliraj(stx3);
  WITH stx3 DO
    st := 2; k[0] := 5.0; k[2] := 2.0;
  END;
  Anuliraj(stx4);
  WITH stx4 DO
    st := 1; k[0] := -1.0; k[1] := 1.0;
  END;
  FOR i := 2 TO n DO
    Izvod(t1, izv);
    Puta(stx3, izv, sab1, ok);
    Puta(stx4, q1, sab2, ok);
    Saberi(sab1, sab2, q);
    Izvod(q, izv);
    Puta(stx1, izv, sab1, ok);
    Puta(stx2, t1, sab2, ok);
    Oduzmi(sab1, sab2, t);
    q1 := q;
    t1 := t;
  END;
  PodeliP(t, a, 2);
  BrojPuta(1.0 / FLOAT(n), a[0], a[0]);
  BrojPuta(1.0 / FLOAT(n - 1), a[1], a[1]);
  Stampaj(a[0], 5, 8);
  Stampaj(a[1], 5, 8);
END Poli1100.
```

Objašnjenje

U ovom zadatku imamo dva medjusobno povezana polinoma, napisana pomoću rekurentnih veza. Potrebno je primetiti da nije svejedno koji će se prvi računati, jer jedan zavisi od trenutnog opšteg polinoma, a drugi od opštih polinoma koji su već u prethodnom koraku izračunati. Zato se mora prvo drugi Q_n polinom izračunati, da bi se mogao tačno i prvi T_n.

U drugom delu zadatka potrebno je primeniti već opisanu proceduru *PodeliP*, ali takodje jedne koeficijente polinoma podeliti sa n, a druge sa $n-1$.

2.2.9 9. Zadatak - Jun 1998.

Text

Napisati program koji učitava polinom $p(x)$ i prirodan broja k (manji od 50), i pronalazi i ispisuje polinom $g_k(x)$, ako se niz polinoma g definiše na sledeći način:

$g_1(x) = p(1) + x$
$g_i(x) = g_{i-1}(p(i)) + x^i, \qquad i > 1$

Program

```
MODULE Poli0698; FROM IO IMPORT WrStr, WrLn, RdInt; FROM PolinomN
IMPORT Polinom, Ucitaj, Anuliraj, Izracunaj;

VAR
  p, g, g1 : Polinom;
  rez : REAL;
  i, k : CARDINAL;
BEGIN
  Ucitaj(p);
  WrStr('Unesite k: ');
  k := RdInt();
  Anuliraj(g1);
  WITH g1 DO
    st := 1; Izracunaj(1.0, p, k[0]); k[1] := 1.0;
  END;
  FOR i := 2 TO k DO
    Anuliraj(g);
    Izracunaj(FLOAT(i), p, rez);
    WITH g DO
      Izracunaj(rez, g1, k[0]);
      k[i] := 1.0;
```

```
    END;
    g1 := g;
  END;
END Poli0698.
```

Objašnjenje

Veoma kratak i lep zadatak. Potrebno je pravilno shvatiti rekurentnu vezu, odnosno da se g_k sastoji od nekog broja $g_{k-1}(p(i))$ i x^k.

2.2.10 10. Zadatak - Jul 2000.

Text

Prvih pedeset članova niza polinoma p se definiše na sledeći naǐn:

$p_1(x) = x - 3$
$p_2(x) = x^2 + 1$
$p_i(x) = (3x^2 + 2)p'_{i-1}(x) - p_{i-2}(10.5)p_{i-2}(x), \qquad i = 3,\ 4,\ \ldots,\ 50$

Prvih pedeset članova niza polinoma q se definiše na sledeći način:

$q_i(x) = (2x^3 - 3x^4 + x)p_i(3.14)p_{51-i}(x), \qquad i = 1,\ 2,\ 3,\ \ldots,\ 50$

Napisati program koji štampa one i samo one članove niza polinoma q, za koje važi da im je niz koeficijenata palindrom. Niz koeficijenata nekog polinoma $r(x) = a_k x^k + \ldots + a_1 x + a_0$ (a_k je različito od 0) je palindrom, ako važi da je $a_i = a_{k-i}$ za $i = 0,\ 1,\ 2,\ \ldots,\ (k\ DIV\ 2)$.

Program

```
MODULE Poli0700; FROM IO IMPORT WrStr, WrLn, RdCard; FROM PolinomN
IMPORT Polinom, Anuliraj, Izracunaj,
                     BrojPuta, Puta, Oduzmi, Stampaj;
FROM PoliProc IMPORT Izvod;

CONST
```

```
  n = 50;
TYPE
  NizPolinoma = ARRAY[1..n] OF Polinom;

PROCEDURE Palindrom(q : Polinom) : BOOLEAN; VAR
  i : CARDINAL;
BEGIN
  FOR i := 0 TO q.st DIV 2 DO
    IF q.k[i] <> q.k[CARDINAL(q.st) - i] THEN
      RETURN FALSE
    END
  END;
  RETURN TRUE;
END Palindrom;

VAR
  p, p1, p2, sab1, sab2, stx2, stx4, izv, q : Polinom;
  rez : REAL;
  i : CARDINAL;
  ok : BOOLEAN;

BEGIN
  Anuliraj(p1);
  WITH p1 DO
    st := 1; k[0] := -3.0; k[1] := 1.0;
  END;
  Anuliraj(p2);
  WITH p2 DO
    st := 2; k[0] := 1.0; k[2] := 1.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[0] := 2.0; k[2] := 3.0;
  END;
  Anuliraj(stx4);
  WITH stx4 DO
    st := 4; k[1] := 1.0; k[3] := 2.0; k[4] := -3.0;
  END;

  p := p1;
  FOR i := 1 TO 2 DO
```

```
    Izracunaj(3.14, p[i], rez);
    Puta(stx4, p, sab1, ok);
    BrojPuta(rez, sab1, q);
    IF Palindrom(q) THEN
      WrInt(51 - i ); WrLn;
      Stampaj(q, 5, 8)
    END;
    p := p2;
  END;

  FOR i := 3 TO n DO
    Izvod(p2, izv);
    Puta(stx2, izv, sab1, ok);
    Izracunaj(10.5, p1, rez);
    BrojPuta(rez, p1, sab2);
    Oduzmi(sab1, sab2, p);
    p1 := p2; p2 := p;

    Izracunaj(3.14, p[i], rez);
    Puta(stx4, p, sab1, ok);
    BrojPuta(rez, sab1, q);
    IF Palindrom(q) THEN
      WrInt(51 - i); WrLn;
      Stampaj(q, 5, 8)
    END;
  END;

END Poli0700.
```

Objašnjenje

Nizove polinoma p i q računamo istovremeno, sa razlikom što dok računamo p_i računamo i q_{51-i}. Za poslednja dva člana niza q potrebno je na početku izračunati.

Napisana je funkcija *Palindrom*, koja za dati polinom vraća da li je on palindrom ili ne. Jednostavnom proverom za svako q_k proverimo i ispišemo ako jeste.

2.2.11 11. Zadatak - Septembar 2001.

Text

Niz polinoma P se definiše na sledeći način:

$P_0(x) = 3$
$P_1(x) = x - 4$
$P_i(x) = P_{i-1}(x^2) - P_{i-1}(4)P_{i-2}(x)$

Za zadatih prvih n polinoma, pronaći da li postoje i, j i k tako da važi:

$P_k^2(x) - P_i(x^2) = P_j(5)P_j(x)$

Program

```
MODULE Poli0901; FROM IO IMPORT WrStr, WrLn, RdCard; FROM PolinomN
IMPORT Polinom, Anuliraj, Puta, Oduzmi,
                     BrojPuta, Izracunaj, Stampaj;
FROM PoliProc IMPORT pODg;

CONST
  maxN = 30;
TYPE
  NizPolinoma = ARRAY[1..maxN] OF Polinom;

PROCEDURE Naci(VAR p : NizPolinoma; n : CARDINAL;
               VAR i1, j1, k1 : CARDINAL) : BOOLEAN;
VAR
  stx2, sab1, sab2, rez1, rez2 : Polinom;
  broj : REAL;
  i, j, k : CARDINAL;
  ok : BOOLEAN;
BEGIN
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[2] := 1.0;
  END;
```

```
  FOR i := 1 TO n - 2 DO
    FOR j := i + 1 TO n - 1 DO
      FOR k := j + 1 TO n DO
        Puta(p[k], p[k], sab1, ok);
        pODg(p[i], stx2, sab2);
        Oduzmi(sab1, sab2, rez1);
        Izracunaj(5.0, p[j], broj);
        BrojPuta(broj, p[j], rez2);
        IF rez1 = rez2 THEN
          i1 := i; j1 := j; k1 := k;
          RETURN TRUE
        END
      END
    END
  END;
  RETURN FALSE;
END Naci;

VAR
  n, i, j, l : CARDINAL;
  p : NizPolinoma;
  stx2, sab1, sab2 : Polinom;
  rez : REAL;

BEGIN
  WrStr('Unesite n: ');
  n := RdCard();
  Anuliraj(p[0]);
  WITH p[0] DO
    st := 0; k[0] := 3.0;
  END;
  Anuliraj(p[1]);
  WITH p[1] DO
    st := 1; k[0] := -4.0; k[1] := 1.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[2] := 1.0;
  END;
  FOR i := 1 TO n DO
    pODg(p[i - 1], stx2, sab1);
```

```
    Izracunaj(4.0, p[i - 1], rez);
    BrojPuta(rez, p[i - 2], sab2);
    Oduzmi(sab1, sab2, p[i]);
  END;

  IF Naci(p, n, i, j, l) THEN
    Stampaj(p[i], 5, 8);
    Stampaj(p[j], 5, 8);
    Stampaj(p[l], 5, 8);
  ELSE
    WrStr('Ne postoje i, j, l tako da vazi dati uslov')
  END

END Poli0901.
```

Objašnjenje

U proceduri *Naci* je opisan postupak kako se proverava da li postoje tri elementa iz nekog skupa koja zadovoljavaju neku osobinu.

2.2.12 12. Zadatak - Septembar 2000.

Text

Dat je skup S od $n(1 < n < 20)$ polinoma. Napisati program koji radi sledeće:

· sa tastature učitava polinome iz skupa S,

· ispituje da li postoje dva polinoma $a(x)$ i $b(x)$ iz skupa S za koje važi relacija:

$$2a(b(x)) = a(x) + b(x) + 2a(x)b(x)$$

ako dva takva polinoma postoje treba ih ispisati na ekranu.

· proverava i ispisuje da li za svaka tri polinoma $a(x)$, $b(x)$ i $c(x)$ iz skupa S važi nejednakost:

$$a(x^3) + b(x^2) + c(x) \neq (x^3 + x^2 + x)(a(x) + 4b(x) - 3x^3c(x))$$

Program

```
MODULE Poli0900; FROM IO IMPORT WrStr, WrLn, RdCard; FROM PolinomN
IMPORT Polinom, Anuliraj, Saberi, Oduzmi,
                     BrojPuta, Puta, Ucitaj, Stampaj;
FROM PoliProc IMPORT pODg;

CONST
  maxN = 20;
TYPE
  NizPolinoma = ARRAY[1..maxN] OF Polinom;

PROCEDURE Naci1(VAR s : NizPolinoma; n : CARDINAL); VAR
  i, j : CARDINAL;
  rez1, sab1, sab2, rez2 : Polinom;
  ok : BOOLEAN;
BEGIN
  FOR i := 1 TO n - 1 DO
    FOR j := i + 1 TO n DO
      pODg(s[i], s[j], rez1);
      BrojPuta(2.0, rez1, rez1);
      Saberi(s[i], s[j], sab1);
      Puta(s[i], s[j], sab2, ok);
      BrojPuta(2.0, sab2, sab2);
      Saberi(sab1, sab2, rez2);
      IF rez1 = rez2 THEN
        Stampaj(s[i], 5, 8);
        Stampaj(s[j], 5, 8);
      END
    END
  END
END Naci1;

PROCEDURE Naci2(VAR s : NizPolinoma; n : CARDINAL); VAR
  i, j, k : CARDINAL;
  stx1, stx2, stx3, stx4, sab1, sab2, rez1, rez2 : Polinom;
  ok : BOOLEAN;
BEGIN
  Anuliraj(stx1);
  WITH stx1 DO
```

```
    st := 3; k[1] := 1.0; k[2] := 1.0; k[3] := 1.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 3; k[3] := 3.0;
  END;
  Anuliraj(stx3);
  WITH stx3 DO
    st := 3; k[3] := 1.0;
  END;
  Anuliraj(stx4);
  WITH stx4 DO
    st := 2; k[2] := 1.0;
  END;
  FOR i := 1 TO n - 2 DO
    FOR j := i + 1 TO n - 1 DO
      FOR k := j + 1 TO n DO
        pODg(s[i], stx3, sab1);
        pODg(s[j], stx4, sab2);
        Saberi(sab1, sab2, sab1);
        Saberi(sab1, s[k], rez1);
        BrojPuta(4.0, s[j], sab1);
        Saberi(s[i], sab1, sab2);
        Puta(stx2, s[k], sab1, ok);
        Oduzmi(sab2, sab1, sab1);
        Puta(stx1, sab1, rez2, ok);
        IF rez1 <> rez2 THEN
          Stampaj(s[i], 5, 8);
          Stampaj(s[j], 5, 8);
          Stampaj(s[k], 5, 8);
        END
      END
    END
  END
END Naci2;


VAR
  s : NizPolinoma;
  n, i : CARDINAL;
```

```
BEGIN
  WrStr('Unesite n: ');
  n := RdCard();
  FOR i := 1 TO n DO
    Ucitaj(s[i]);
  END;
  Naci1(s, n);
  Naci2(s, n);
END Poli0900.
```

Objašnjenje

Ovde su opisane procedure koje proveravaju svaka dva, odnosno svaka tri elementa iz nekog skupa.

2.2.13 13. Zadatak - Oktobar II 2001.

Text

Prvih trideset članova niza polinoma P se definiše na sledeći način:

$$P_0(x) = x$$
$$P_1(x) = x - 4$$
$$P_2(x) = x^3 - 7$$
$$P_k(x) = P_{k-1}(x^3)P_{k-2}(x+2) + P_{k-3}(x)P_{k-1}(x), \qquad k = 3, \ldots,\ 30$$

Za unapred zadato $n < 30$ ispitati za svakih 5 uzastopnih koeficijenata polinoma $P_n(x)$ da li važi sledeće:

$$p.k[i]^2 + p.k[i+1]^2 = p.k[i+2]^2 = p.k[i+3]^2 + p.k[i+4]^2$$

Ukoliko je polinom stepena manjeg od 5, podrazumeva se da traženi uzastopni niz koeficijenata ne postoji.

Program

```
MODULE Poli1001; FROM IO IMPORT WrStr, WrLn, WrCard, RdCard; FROM
PolinomN IMPORT Polinom, Anuliraj, Puta, Saberi; FROM PoliProc
IMPORT pODg;

PROCEDURE Jednakost(p : Polinom; i : CARDINAL) : BOOLEAN; VAR
  rez1, rez2, rez3 : REAL;
BEGIN
  rez1 := p.k[i] * p.k[i] + p.k[i + 1] * p.k[i + 1];
  rez2 := p.k[i + 2] * p.k[i + 2];
  rez3 := p.k[i + 3] * p.k[i + 3] + p.k[i + 4] * p.k[i + 4];
  IF rez1 <> rez2 THEN
    RETURN FALSE;
  END;
  IF rez2 <> rez3 THEN
    RETURN FALSE
  END;
  RETURN TRUE;
END Jednakost;

VAR
  p0, p1, p2, p, cin1, cin2, sab1, sab2, stx3, stx1 : Polinom;
  n, i : CARDINAL;
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;
  Anuliraj(p0);
  WITH p0 DO
    st := 3; k[3] := 1.0;
  END;
  Anuliraj(p1);
  WITH p1 DO
    st := 1; k[0] := -4.0; k[1] := 1.0;
  END;
  Anuliraj(p2);
  WITH p2 DO
    st := 3; k[0] := -7.0; k[3] := 1.0;
```

```
    END;
    Anuliraj(stx3);
    WITH stx3 DO
      st := 3; k[3] := 1.0;
    END;
    Anuliraj(stx1);
    WITH stx1 DO
      st := 1; k[0] := 3.0; k[1] := 1.0;
    END;
    FOR i := 3 TO n DO
      pODg(p2, stx3, cin1);
      pODg(p1, stx1, cin2);
      Puta(cin1, cin2, sab1, ok);
      Puta(p0, p2, sab2, ok);
      Saberi(sab1, sab2, p);
      p0 := p1; p1 := p2; p2 := p;
    END;
    FOR i := 0 TO p.st - 4 DO
      IF Jednakost(p, i) THEN
        WrCard(i, 1); WrLn;
      END
    END
  END Poli1001.
```

Objašnjenje

Procedura *Jednakost* proverava traženu ekvivalenciju tri elementa.

2.2.14 14. Zadatak - Jun 2001.

Text

Dat je polinom:

$$P(x) = P_n x^n + P_{n-1} x^{n-1} + \ldots + P_1 x + P_0$$

(pored velikog P se nalazi indeks) stepena ne veceg od 6.

Koeficijente polinoma učitati iz fajla. Zna se da polinom ima tri različite višestruke nule. Prva nula je delioc slobodnog člana polinoma $P(x)$, druga je delioc koeficijenta uz najveći stepen polinoma, a treća je jednaka zbiru delioca slobodnog člana i delioca koeficijenta uz najveći stepen. Naći višestrukost nula polinoma i ispisati polinom u obliku

$$P(x) = (x - x_1)^{i_1}(x - x_2)^{i_2}(x - x_3)^{i_3}(B_k x^k + B_{k-1}x^{k-1} + \ldots + B_1 x + B_0)$$

Program

```
MODULE Poli0601; FROM IO IMPORT WrStr, WrReal, WrCard; FROM
PolinomN IMPORT Polinom(*, Ucitaj*), Anuliraj,
                     Izracunaj, Deli, Stampaj;

TYPE
  Nula = RECORD
           nula : REAL;
           br : CARDINAL;
         END;
  NizNula = ARRAY[1..3] OF Nula;


PROCEDURE Deljiv(p1, p2 : Polinom; VAR kol : Polinom) : BOOLEAN;
VAR
  ok : BOOLEAN;
  ost : Polinom;
BEGIN
  Deli(p1, p2, kol, ost, ok);
  IF ost.st = -1 THEN
    RETURN TRUE
  ELSE
    RETURN FALSE
  END
END Deljiv;

PROCEDURE VisestrukostS(p : Polinom; VAR b : Polinom;
                                    VAR q : NizNula);
VAR
  i : CARDINAL;
  pnula, kol : Polinom;
BEGIN
  FOR i := 1 TO 3 DO
```

```
    q[i].br := 0;
    Anuliraj(pnula);
    WITH pnula DO
      pnula.st := 1; pnula.k[0] := -q[i].nula; pnula.k[1] := 1.0;
    END;
    LOOP
      IF Deljiv(p, pnula, kol) THEN
        INC(q[i].br);
        p := kol
      ELSE
        EXIT;
      END
    END
  END;
  b := p;
END Visestrukost;

PROCEDURE Upisi(a, b, c : INTEGER; VAR q : NizNula); BEGIN
  q[1].nula := FLOAT(a);
  q[2].nula := FLOAT(b);
  q[3].nula := FLOAT(c);
END Upisi;

PROCEDURE NadjiS(a : INTEGER; p : Polinom; VAR q : NizNula); VAR
  b, c : INTEGER;
  rez : REAL;
BEGIN
  b := 1;
  WHILE b <= INTEGER(p.k[p.st]) DO
    IF INTEGER(p.k[p.st]) MOD b = 0 THEN
      Izracunaj(FLOAT(b), p, rez);
      IF rez = 0.0 THEN
        c := a + b;
        Upisi(a, b, c, q);
      END;
      Izracunaj(-FLOAT(b), p, rez);
      IF rez = 0.0 THEN
        c := a - b;
        Upisi(a, b, c, q);
      END;
    END;
```

```
    INC(b);
  END
END Nadji;

PROCEDURE NuleS(p : Polinom; VAR q : NizNula); VAR
  a : INTEGER;
  rez : REAL;
BEGIN
  a := 1;
  WHILE a <= INTEGER(p.k[0]) DO
    IF INTEGER(p.k[0]) MOD a = 0 THEN
      Izracunaj(FLOAT(a), p, rez);
      IF rez = 0.0 THEN
        Nadji(a, p, q)
      END;
      Izracunaj(-FLOAT(a), p, rez);
      IF rez = 0.0 THEN
        Nadji(-a, p, q)
      END
    END;
    INC(a)
  END
END Nule;

PROCEDURE Ispisi(p, b : Polinom; q : NizNula); BEGIN
  Stampaj(p, 5, 8);
  WrStr(' = (x - '); WrReal(q[1].nula, 5, 8); WrStr(')^'); WrCard(q[1].br, 1);
  WrStr(' (x - '); WrReal(q[2].nula, 5, 8); WrStr(')^'); WrCard(q[2].br, 1);
  WrStr(' (x - '); WrReal(q[3].nula, 5, 8); WrStr(')^'); WrCard(q[3].br, 1);
  WrStr(' ('); Stampaj(b, 5, 8); WrStr(')');
END Ispisi;

VAR
  p, b : Polinom;
  q : NizNula;
BEGIN
  Ucitaj(p);
  NuleS(p, q);
  VisestrukostS(p, b, q);
  Ispisi(p, b, q);
END Poli0601.
```

Objašnjenje

Procedure korišćene u ovom zadatku su mala modifikacija objašnjenih procedura potrebnih za rad procedure $Visestrukost$. Pošto se ovde na specifičan način traže nule, ceo program je ispisan. Objasnjenja su veoma slična.

2.2.15 15. Zadatak - April 2002.

Text

Napisati program koji učitava niz od n tačaka u ravni koje su zadate svojim koordinatama (x_i, y_i), $x_i, y_i \in \mathbf{Z}$, $\quad i = 1..n$, i formira interpolacioni polinom koji prolazi kroz te tačke. Interpolacioni polinom definiše se sa:

$$I(x) = \sum_{i=1}^{n} y_i \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j}$$

Program zatim treba da formira polinom:

$$J(x) = \left(\prod_{i=1}^{k} (x - z_i)^{Vi} \right)'$$

pri čemu su z_i sve celobrojne nule polinoma $I(x)$ čija je višestrukost prost broj, a Vi je višestrukost i-te nule. Na kraju odštampati polinom $J(x)$.

Program

```
MODULE Poli0402;
FROM IO IMPORT WrStr, WrLn, RdInt, RdCard;
FROM PolinomN IMPORT Polinom, Anuliraj, Puta,
                     BrojPuta, Saberi, Stampaj;
FROM PoliProc IMPORT NizNula, Visestrukost, Izvod;

CONST
  maxN = 1000;
TYPE
  tacka = RECORD
```

```
          x, y : INTEGER;
        END;
  NizTacaka = ARRAY[1..maxN] OF tacka;

PROCEDURE UcitajTacke(VAR n : CARDINAL; a : NizTacaka);
VAR
  i : CARDINAL;
BEGIN
  WrStr('Unesite n: ');
  n := RdCard();
  FOR i := 1 TO n DO
    WITH a[i] DO
      WrStr('Unesite x kordinatu tacke: ');
      x := RdInt();
      WrStr('Unesite y kordinatu tacke: ');
      y := RdInt();
    END;
  END
END UcitajTacke;

PROCEDURE Prost(broj : CARDINAL) : BOOLEAN;
VAR
  i : CARDINAL;
BEGIN
  FOR i := 2 TO broj DIV 2 DO
    IF broj MOD i = 0 THEN
      RETURN FALSE;
    END
  END;
  RETURN TRUE
END Prost;

VAR
  q : NizNula;
  a : NizTacaka;
  n, i, j : CARDINAL;
  p, sab, stxj, I, J, stx : Polinom;
  proiz : INTEGER;
  ok : BOOLEAN;

BEGIN
```

```
  UcitajTacke(n, a);
  FOR i := 1 TO n DO
    Anuliraj(p);
    WITH p DO
      st := 0; k[0] := 1.0;
    END;
    proiz := 1;
    FOR j := 1 TO n DO
      IF i <> j THEN
        Anuliraj(stxj);
        WITH stxj DO
          st := 1; k[0] := -FLOAT(a[j].x); k[1] := 1.0;
        END;
        Puta(p, stxj, p, ok);
        proiz := proiz * (a[i].x - a[j].x);
      END
    END;
    BrojPuta(FLOAT(a[i].y) / FLOAT(proiz), p, sab);
    Saberi(I, sab, I);
  END;
  Visestrukost(I, q);
  Anuliraj(p);
  WITH p DO
    st := 0; k[0] := 1.0;
  END;
  FOR i := 1 TO q[0].br DO
    IF Prost(q[i].br) THEN
      Anuliraj(stx);
      WITH stx DO
        st := 1; k[0] := -q[i].nula; k[1] := 1.0;
      END;
      Puta(p, stx, p, ok);
    END;
  END;
  Izvod(p, J);
  Stampaj(J, 5, 8);
END Poli0402.
```

Objašnjenje

Za računanje interpolacionog polinoma potrebne su nam pomoćne promenljive:

$$P_i(x) = \prod_{\substack{j=1 \\ j<>i}}^{n} (x - x_j)$$

$$proiz_i = \prod_{\substack{j=1 \\ j<>i}}^{n} (x_i - x_j)$$

$$I(x) = \sum_{i=1}^{n} \frac{y_i}{proiz_i} P_i(x)$$

Zatim, nadjemo višestruke nule polinoma $I(x)$.

Pomoću funkcije *Prost* odredjujemo koje su od tih nula pojavljuju prost broj puta i njih množimo. Izvod od dobijenog polinoma nam predstavlja rešenje $J(x)$.

2.2.16 16. Zadatak - April 1999.

Text

Napisati program koji odredjuje i štampa polinom $p_n(x)$ za koji važi sledeća relacija:

$$p_n(x) = \sum_{k=1}^{n} \left((-1)^{n-k} \frac{n}{k} x^k \right) - p_{n-1}(x) \sum_{k=1}^{n} (x^2 - x + 3^k)(x-2)^{2k-1}$$

gde je $p_1(x) = 3x^2 - 2$, a n zadat prirodan broj manji od 20.

Program

```
MODULE Poli0499;
FROM IO IMPORT WrStr, WrLn, RdCard;
FROM PolinomN IMPORT Polinom, Anuliraj, Puta,
                     Oduzmi, Stampaj;

VAR
  stx1, stx2, p1, p, pom, sab1, sab2, sab3 : Polinom;
  n, i : CARDINAL;
  zn : REAL;
```

```
  ok : BOOLEAN;

BEGIN
  WrStr('Unesite n: ');
  n := RdCard(); WrLn;
  Anuliraj(p1);
  WITH p1 DO
    st := 2; k[0] := -2.0; k[2] := 3.0;
  END;
  Anuliraj(stx2);
  WITH stx2 DO
    st := 2; k[0] := 3.0; k[1] := -1.0; k[2] := 1.0;
  END;
  Anuliraj(stx1);
  WITH stx1 DO
    st := 1; k[0] := -2.0; k[1] := 1.0;
  END;
  Puta(stx1, stx2, sab2, ok);
  Puta(stx1, stx1, pom, ok);
  IF (n - 1) MOD 2 = 0 THEN
    zn := 1.0
  ELSE
    zn := -1.0
  END;
  Anuliraj(sab1);
  WITH sab1 DO
    st := 1; k[1] := zn * FLOAT(n);
  END;
  FOR i := 2 TO n DO
    zn := -zn;
    WITH sab1 DO
      st := i; k[i] := zn * FLOAT(n) / FLOAT(i);
    END;
    stx2.k[0] := stx2.k[0] * 3.0;
    Puta(stx1, pom, stx1, ok);
    Puta(stx1, stx2, sab3, ok);
    Saberi(sab2, sab3, sab2);
    Puta(p1, sab2, sab3, ok);
    Oduzmi(sab1, sab3, p);
    p1 := p;
  END;
```

```
  Stampaj(p, 5, 8);
END Poli0499.
```

Objašnjenje

Ovaj zadatak je veoma specifičan. Kao i uvek, postoji rekurentna veza za traženi niz polinoma, ali ovom prilikom prvi put se srećemo sa sumom. Način rešavanja ovakvih zadataka je veoma sličan načinu kako smo do sada naučili da rešavamo sume - rekurentne veze.

$$sab1_k = \sum_{k=1}^{n} (-1)^{n-k} \frac{n}{k} x^k$$

$$sab1_k = sab1_{k-1} + (-1)^{n-k} \frac{n}{k} x^k$$

$$sab1_1 = (-1)^{n-1} nx$$

$$pom_k = (x-2)^{2k-1}$$

$$pom_k = pom_{k-1}(x-2)^2$$

$$pom_1 = x - 2$$

$$br_k = 3^k$$

$$br_k = br_{k-1} \cdot 3$$

$$br_1 = 3$$