

5. Algoritmy pre výpočet hodnôt funkcie

- pre korektný výpočet sa používa rozvoj do radov
- nasledujúci člen je vyjadrený dosadením predchádzajúceho člena do rekurentného vzorca
- ukončenie rozvoja je dané presnosťou E

1. Zostavte algoritmus pre výpočet $F(x)$ podľa vzťahu:

$$F(x) = 1 + \frac{x \cdot 3}{x^3} + \frac{x \cdot 5}{x^5} + \frac{x \cdot 7}{x^7} + \dots + \frac{x \cdot n}{x^n}$$

Uvažujte taký počet členov rozvoja radu, aby rozdiel medzi poslednými dvomi členmi bol menej ako 0,001.

Vstupné premenné: x

Pomocné premenné: a, y, pom, pres

Výstupná premenná: sa

Analýza riešenia:

Pre zvyšovanie násobiteľov a mocniteľov čísla x využijeme pomocnú premennú y, ktorej hodnotu budeme zvyšovať o 2. Pri hodnote premennej $y=1$ je funkčná hodnota rovná 1, teda hodnotu prvého člena radu a nastavíme na začiatku na 1. Následne v cykle zvyšujeme hodnotu y o 2 a počítame hodnotu ďalších členov radu. Výpočet ukončíme vtedy, ak rozdiel dvoch po sebe nasledujúcich členov je menej ako 0,001. Na zistenie rozdielu nám slúži pomocná premenná pres, ktorá je rozdielom premennej pom, do ktorej sme uložili hodnotu predchádzajúceho člena a premennej a, čo je vlastne aktuálne vypočítaný člen radu.

Slovný popis algoritmu:

1. krok: načítať hodnotu x
2. krok: nastaviť hodnoty pomocných premenných: $y=1$, $a=1$
3. krok: výstupnej premennej sa priradiť hodnotu prvého člena radu: $sa=a$
4. krok: priradiť nasledujúcim premenným dané hodnoty:

$$pom = a$$

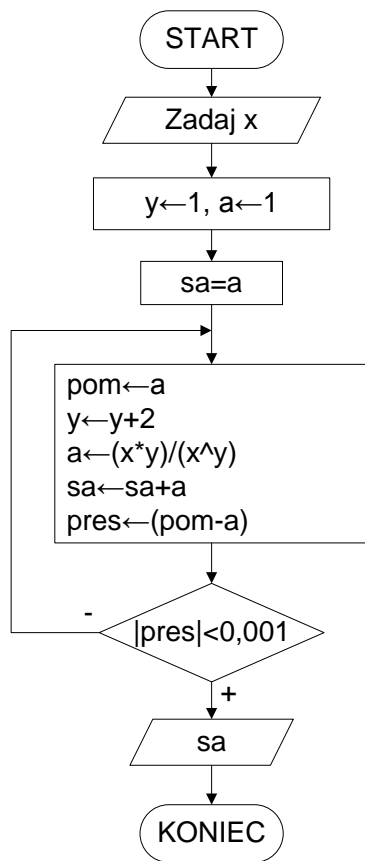
$$y = y + 2$$

$$a = \frac{x \cdot y}{x^y}$$

$$sa = sa + a$$

$$pres = pom - a$$

5. krok: ak $|pres| < 0,001$ pokračuj na krok 6, inak opakuj krok 4
6. krok: vypísať hodnotu výstupnej premennej sa



```

int main(void)
{
    printf("Vypocet hodnoty funkcie\n\n");
    float y=1,sa,a=1,x,pom,pres;
    printf("Zadaj x: ");
    scanf("%f",&x);

    sa=a;
    while(1){
        pom=a;
        y=y+2;
        a=(x*y)/pow(x,y);
        sa=sa+a;
        pres=pom-a;
        if(fabs(pres)<0.001)
            break;
    }
    printf("Hodnota danej funkcie pre x=%0.2f je %0.5f.\n\n",x,sa);

    system("PAUSE");
    return 0;
}
  
```

2. Navrhните algoritmus pre výpočet druhej odmocniny $X = \sqrt{a}$ pričom $a > 0$. Pre výpočet použite Newtonov iteračný vzorec: $x_{i+1} = \frac{1}{2}(x_i + a/x_i)$, kde $i = 0, 1, 2 \dots n$. V prípade zadania čísla $a \leq 0$, program vypíše chybu.

Vstupné premenné: a, E

Pomocné premenné: x, pres

Výstupné premenné: od

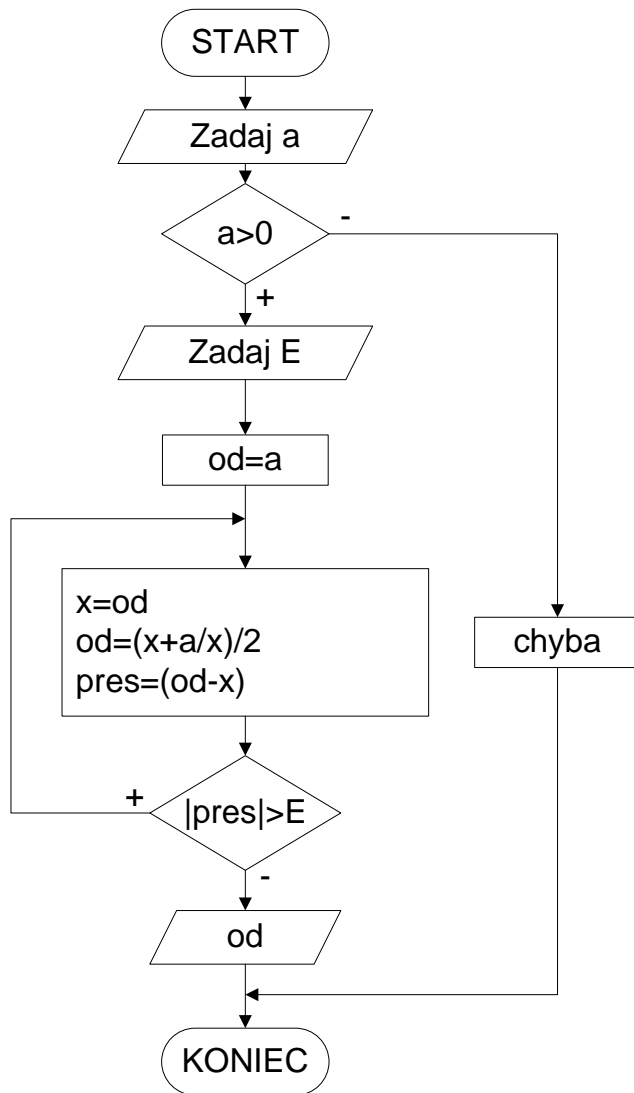
Analýza riešenia:

Prvou vstupnou premennou je číslo a, z ktorého sa bude počítat' odmocnina. Použijeme podmienku či je zadané číslo väčšie ako 0, pretože číslo umocnené na druhú nemôže mať záporný výsledok. Pri nesplnení podmienky program vypíše chybu a ukončí sa, ak je podmienka splnená, používateľ bude vyzvaný k zadaniu presnosti výpočtu.

Premennej od priradíme najprv hodnotu premennej a a následne spustíme cyklus, v ktorom je hodnota premennej od priradená premennej x, čo nám slúži na výpočet presnosti v premennej pres. Cyklus sa opakuje dovtedy, kým rozdiel dvoch po sebe nasledujúcich členov nie je menší ako zadaná presnosť.

Slovný popis algoritmu:

1. krok: zadaj číslo a
2. krok: ak je $a > 0$ pokračuj krokom 3, inak vypíš chybu a ukončí program
3. krok: zadaj presnosť E
4. krok: výstupnej premennej od prirad' číslo a
5. krok: nasledujúcim premenným prirad' tieto hodnoty:
$$x = od$$
$$od = \frac{1}{2}\left(x + \frac{a}{x}\right)$$
$$pres = od - x$$
6. krok: ak je $|pres| > E$, opakuj krok 5, inak prejdi na krok 7
7. krok: vypíš hodnotu druhej odmocniny od



```

int main(void)
{
    printf("Vypocet 2. odmocniny zo zadaneho
    cisla\n\n");
    float a,E,od,pres,x;

    printf("Zadaj a: ");
    scanf("%f",&a);

    if(a>0)
    {
        printf("Zadaj E: ");
        scanf("%f",&E);
        od=a;
        do
        {
            x=od;
            od=(x+a/x)/2;
            pres=(od-x);
        }
        while(fabs(pres)>E);

        printf("\nDruha odmocnina z cisla %0.2f je
        %0.2f.\n\n",a,od);
    }
    else printf("Chybne cislo!\n\n");

    system("PAUSE");
    return 0;
}
  
```