

### 3. Algoritmy s opakovaním

- použitím cyklu si môžeme ušetriť písanie dlhého zdrojového kódu pri programe, v ktorom na vstupe zadávame viac hodnôt do vstupnej premennej
- umožňujú riešenie aj zložitejších úloh opakovaním vybraných krokov
- cyklus môžeme ukončiť rôznymi spôsobmi, napr. zadaním čísla, ktoré uvedieme v podmienke alebo dosiahnutím zadaného počtu vstupov
- poznáme niekoľko typov cyklov:

**cyklus s podmienkou na začiatku** – po splnení podmienky sa vykoná telo cyklu, pri nesplnení podmienky cyklus končí, pričom sa telo nemusí vykonať ani raz

**cyklus s podmienkou na konci** – telo cyklu sa vykoná minimálne raz, po vykonaní cyklu podmienka rozhodne či sa cyklus zopakuje alebo nie

**cyklus s daným počtom opakovaní** – na začiatku sa určí podmienka dokedy má cyklus bežať zadaním tzv. riadiacej premennej  $i$

1. Vytvorte algoritmus, ktorý vypočíta súčin zadaných čísel. Zadávanie čísel bude ukončené číslom 1, aby výsledok nebol skreslený. Použite cyklus WHILE s podmienkou na konci.

Vstupné premenné:  $x$

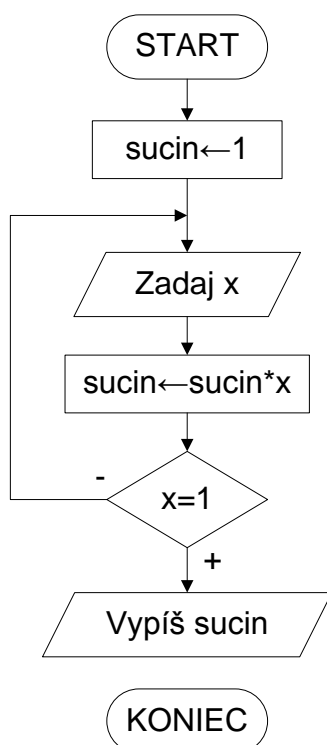
Výstupná premenná:  $sucin$

Analýza riešenia:

V prvom rade nastavíme počiatočnú hodnotu premennej  $sucin$  na 1. Program vyzve užívateľa k zadaniu čísla a po zadaní vypočíta nový súčin po vynásobení aktuálnej hodnoty súčinu zadaným číslom. Program opätovne vyzýva užívateľa k zadaniu ďalšieho čísla, až kým zadané číslo nie je rovné 1.

Slovný popis algoritmu:

1. krok:  $sucin \leftarrow 1$
2. krok: zadaj  $x$
3. krok:  $sucin = sucin * x$
4. krok: ak  $x = 1$  pokračuj krokom 5 inak opakuj krok 2
5. krok: vypíš  $sucin$



```
int main(void)
{
    int x,sucin=1;

    printf("Sucin zadanych cisel \n\n");

    printf("Zadajte cisla x (ukoncite cislom 1): \n");
    while(1){
        scanf("%d",&x);
        sucin=sucin*x;
        if(x==1)
            break;
    }

    printf("Sucinom zadanych cisel je cislo: %d \n\n",sucin);

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

## 2. Vytvorte algoritmus na výpočet n-tej mocniny zadaného čísla x.

Vstupné premenné: x, n

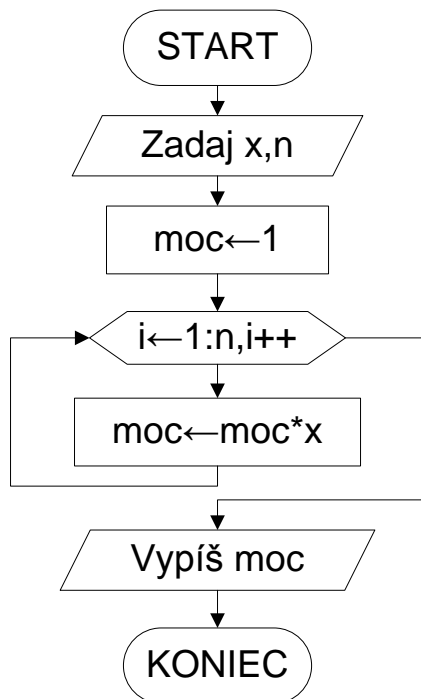
Výstupné premenné: moc

Analýza riešenia:

Užívateľ na vstupe zadá dvojicu čísel, prvé číslo, ktoré chce umocniť (x) a následne mocniteľa (n). Hodnotu výstupnej premennej nastavíme na začiatku na 1. Cyklus FOR beží od čísla 1 po n a aktuálnu hodnotu premennej moc násobí číslom x, kým nie je cyklus ukončený. Nakoniec vypíšeme premennú moc na monitor.

Slovný popis algoritmu:

1. krok: zadať vstupné premenné: x, n
2. krok: nastaviť hodnotu výstupnej premennej: moc=1
3. krok: ak je  $i < n$  pokračovať krokom 4, inak prejsť na krok 6
4. krok:  $pom = pom * x$
5. krok:  $i++$ , vráť sa na krok 3
6. krok: vypísať moc



```
int main(void)
{
    int x,i,n,moc=1;

    printf("Vypocet n-tej mocniny cisla x \n\n");

    printf("Zadaj cislo x: ");
    scanf("%d",&x);

    printf("Zadaj mocnitela n: ");
    scanf("%d",&n);

    for(i=1;i<=n;i++){
        moc=moc*x;
    }

    printf("\nCislo %d umocnene na %d je %d. \n\n",x,n,moc);

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