2. lekce

Algoritmus, cyklus

Miroslav Jílek

Algoritmus

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data
- výsledky musí být nalezeny po konečném počtu operací (kroků)

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data
- výsledky musí být nalezeny po konečném počtu operací (kroků)

Vlastnosti algoritmu

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data
- výsledky musí být nalezeny po konečném počtu operací (kroků)

Vlastnosti algoritmu

- *univerzálnost* – řeší skupinu podobných problémů, konkrétní problém je zadán pomocí vstupních dat

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data
- výsledky musí být nalezeny po konečném počtu operací (kroků)

Vlastnosti algoritmu

- *univerzálnost* – řeší skupinu podobných problémů, konkrétní problém je zadán pomocí vstupních dat
- *determinovanost* – každý krok algoritmu je jednoznačně (clear) definován, výsledek zpracování stejných dat musí dát stejný výsledek

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data
- výsledky musí být nalezeny po konečném počtu operací (kroků)

Vlastnosti algoritmu

- *univerzálnost* – řeší skupinu podobných problémů, konkrétní problém je zadán pomocí vstupních dat
- *determinovanost* – každý krok algoritmu je jednoznačně (clear) definován, výsledek zpracování stejných dat musí dát stejný výsledek
- *resultativnost* – pro každá přípustná vstupní data musí být výsledek

Algoritmus

- je konečná posloupnost operací, která dává řešení skupiny problémů
- je tvořen seznamem (list) přesně definovaných příkazů
- vrací očekávané výsledky pro jakákoli vstupní data
- výsledky musí být nalezeny po konečném počtu operací (kroků)

Vlastnosti algoritmu

- *univerzálnost* – řeší skupinu podobných problémů, konkrétní problém je zadán pomocí vstupních dat
- *determinovanost* – každý krok algoritmu je jednoznačně (clear) definován, výsledek zpracování stejných dat musí dát stejný výsledek
- *resultativnost* – pro každá přípustná vstupní data musí být výsledek
- *konečnost* – pro každá přípustná vstupní data bude po konečném počtu kroků výsledek

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Popis algoritmu

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Popis algoritmu

a) přirozeným jazykem (čeština, angličtina, ruština, …)

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Popis algoritmu

a) přirozeným jazykem (čeština, angličtina, ruština, …)

b) vývojový diagram (flowchart)

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Popis algoritmu

a) přirozeným jazykem (čeština, angličtina, ruština, …)

b) vývojový diagram (flowchart)

c) stuktogram (např.: Nassi-Shneidermann diagram)

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Popis algoritmu

a) přirozeným jazykem (čeština, angličtina, ruština, …)

b) vývojový diagram (flowchart)

c) stuktogram (např.: Nassi-Shneidermann diagram)

d) pseudokód

Vstup algoritmu

Data (informace), která popisují problém, který máme vyřešit.

Výstup algoritmu

Požadovaný (očekávaný) výsledek.

Efektivita algoritmu

a) z hlediska rychlosti zpracování

b) z hlediska paměti počítače

Popis algoritmu

a) přirozeným jazykem (čeština, angličtina, ruština, …)

b) vývojový diagram (flowchart)

c) stuktogram (např.: Nassi-Shneidermann diagram)

d) pseudokód

e) programovacím jazykem (C, C++, C#, java, javascript, pascal, python, visual basic, ….)

Příklady popisu algoritmu:

Úkol: z množiny čísel nalezněte největší číslo (maximum)

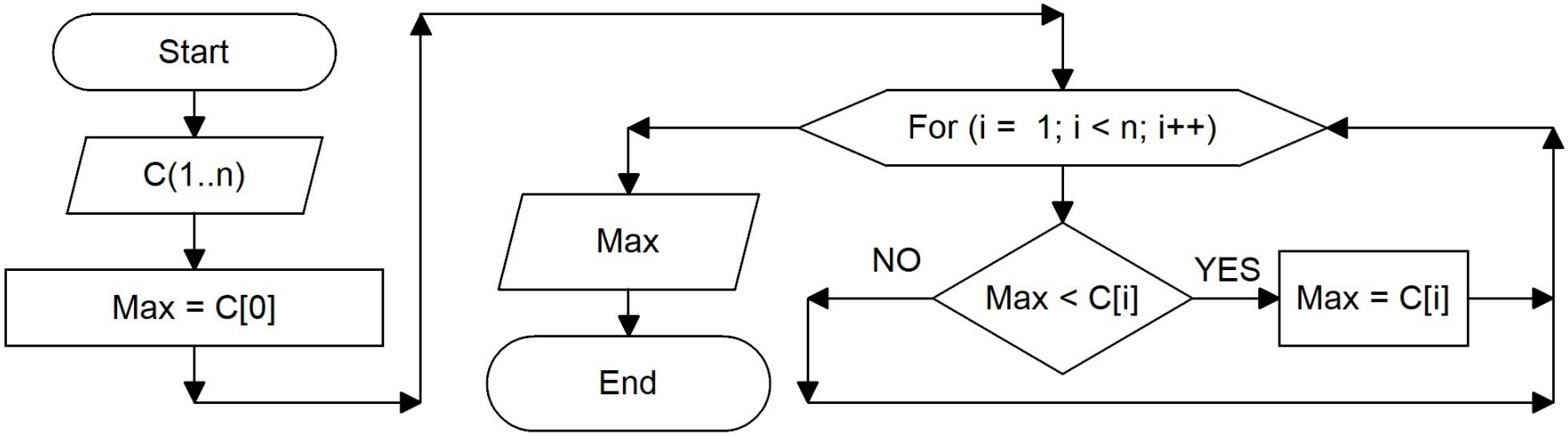
a) přirozený jazyk

Maximum je první číslo.

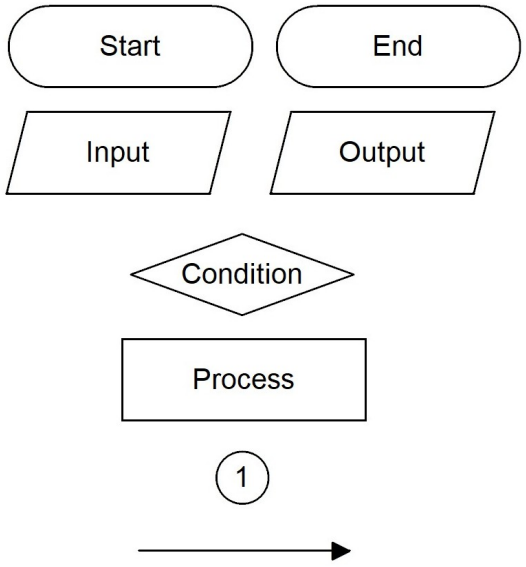
Postupně, pro všechna další čísla z množiny čísel, budeme zjišťovat, jestli dané číslo je větší než aktuální maximum. Pokud ano, pak nové maximum bude toto číslo.

Výsledkem bude maximum po testování posledního čísla množiny.

b) vývojový diagram



Symboly použité ve vývojovém diagramu:



začátek nebo konec algoritmu

vstup nebo výstup

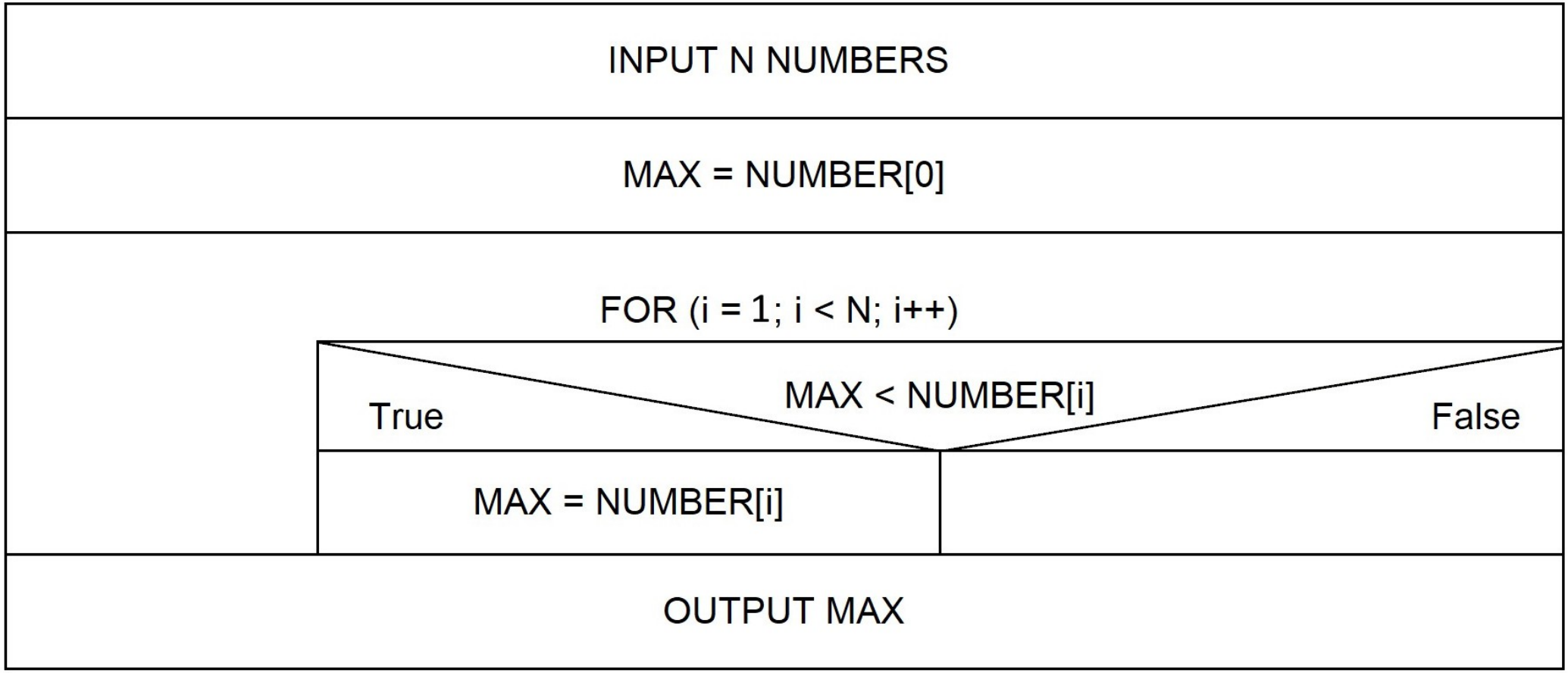
rozvětvení algoritmu na základě vyhodnocení podmínky

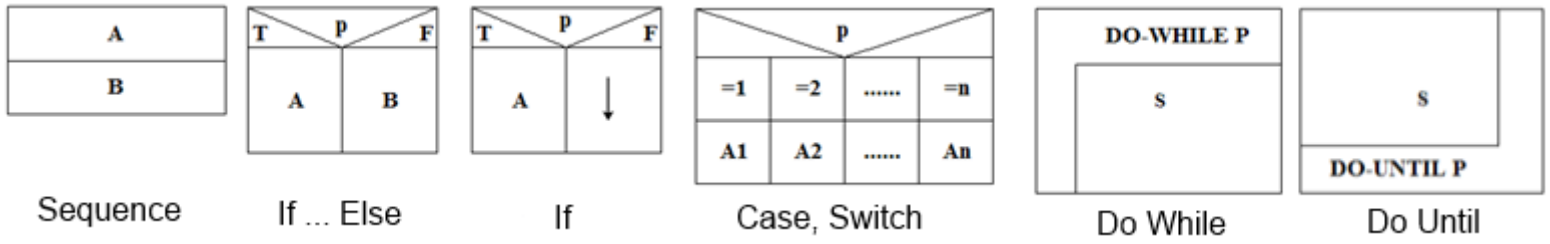
vnitřní proces algoritmu, přižazení hodnoty

napojovací znak

propojovací šipka, definuje posloupnost příkazů algoritmu

c) struktogram (Nassi-Shneidermann diagram)

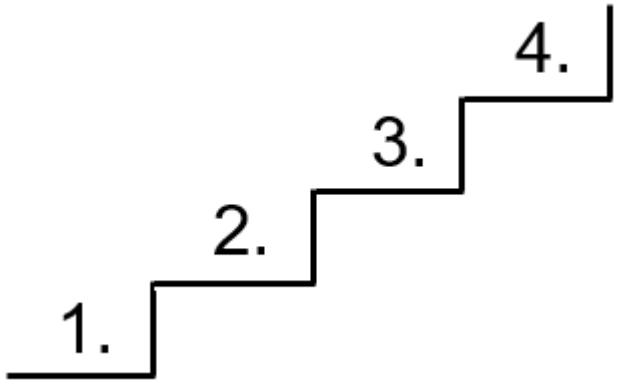




d) pseudokód

Nacti pole N čísel
Max = první číslo v poli
Pro čísla od druhého do N-tého proveď:
    Jestli Číslo > Max, potom Max = Číslo
Zobraz maximum

Používají se příkazy v pořadí, v jakém budou v algoritmu provedeny!



e) programovací jazyk *(příklad pro C)*

```
int x, max, pole[]={5, 6, 17, 0, 6, -4};
max = pole[0];
for (x=1; x<=5; x++)
{
      if (max < pole[x]) max = pole[x];
}
printf (“Max = %d\n”,max);
```

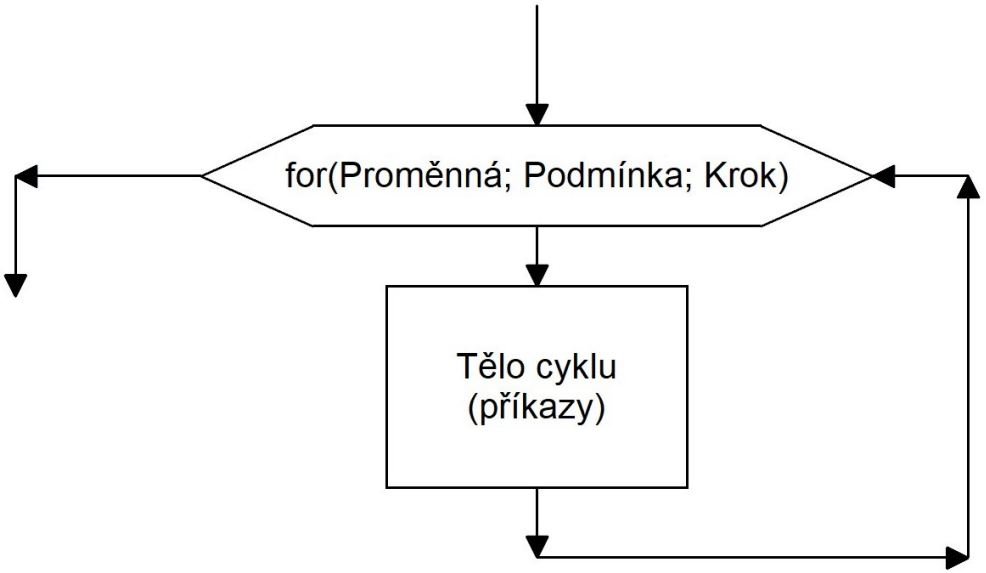
Cykly a podmínky

Cyklus je část algoritmu, ve které se opakuje definovaná činnost – příkazy.

Cyklus For

Řídící proměnná, *(... její hodnota za cyklem!)*
Sestupný, vzestupný
Definovaný počet cyklů, smyčka (loop) cyklu

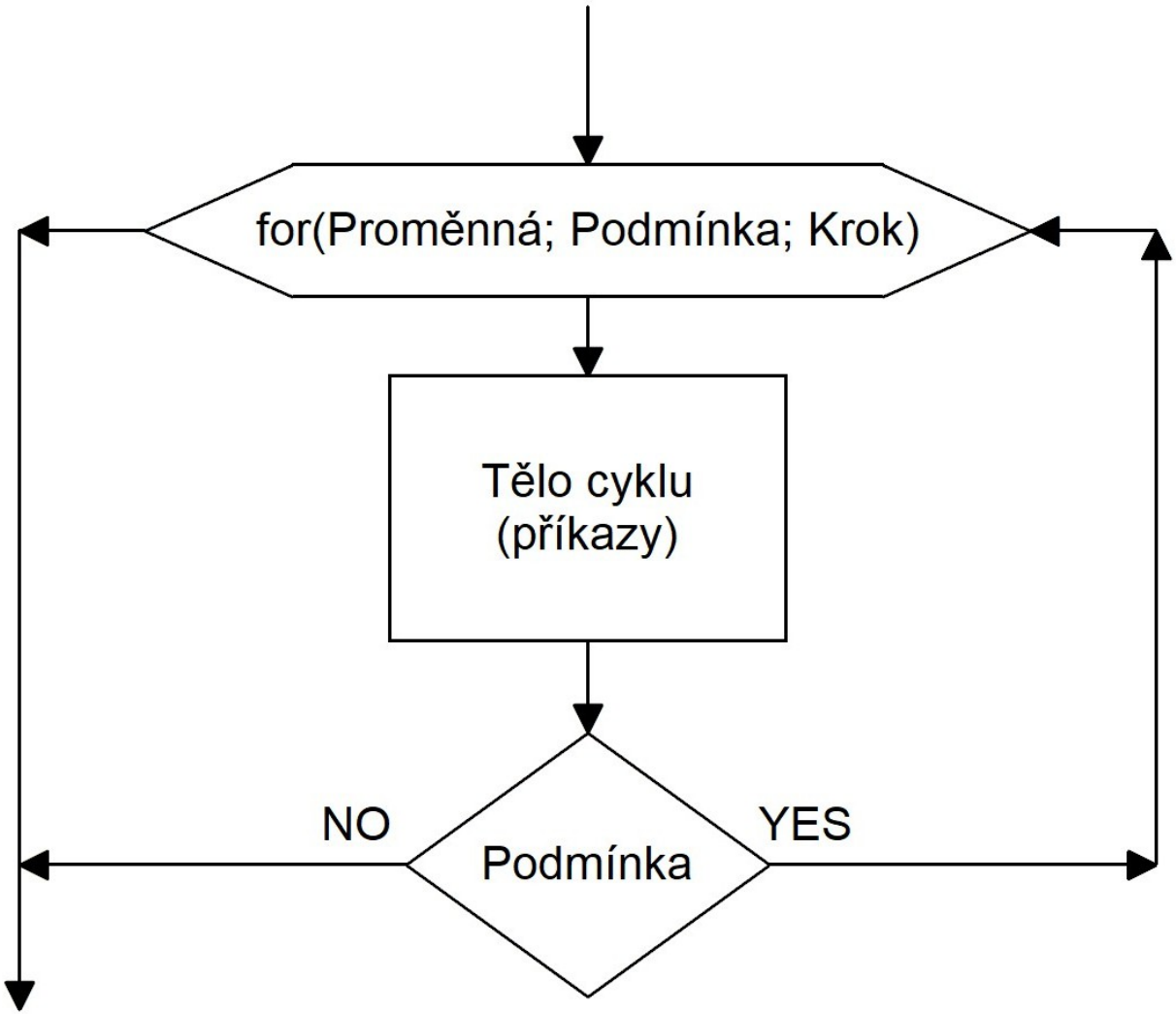
Základní varianta cyklu For:



Parametry cyklu For:

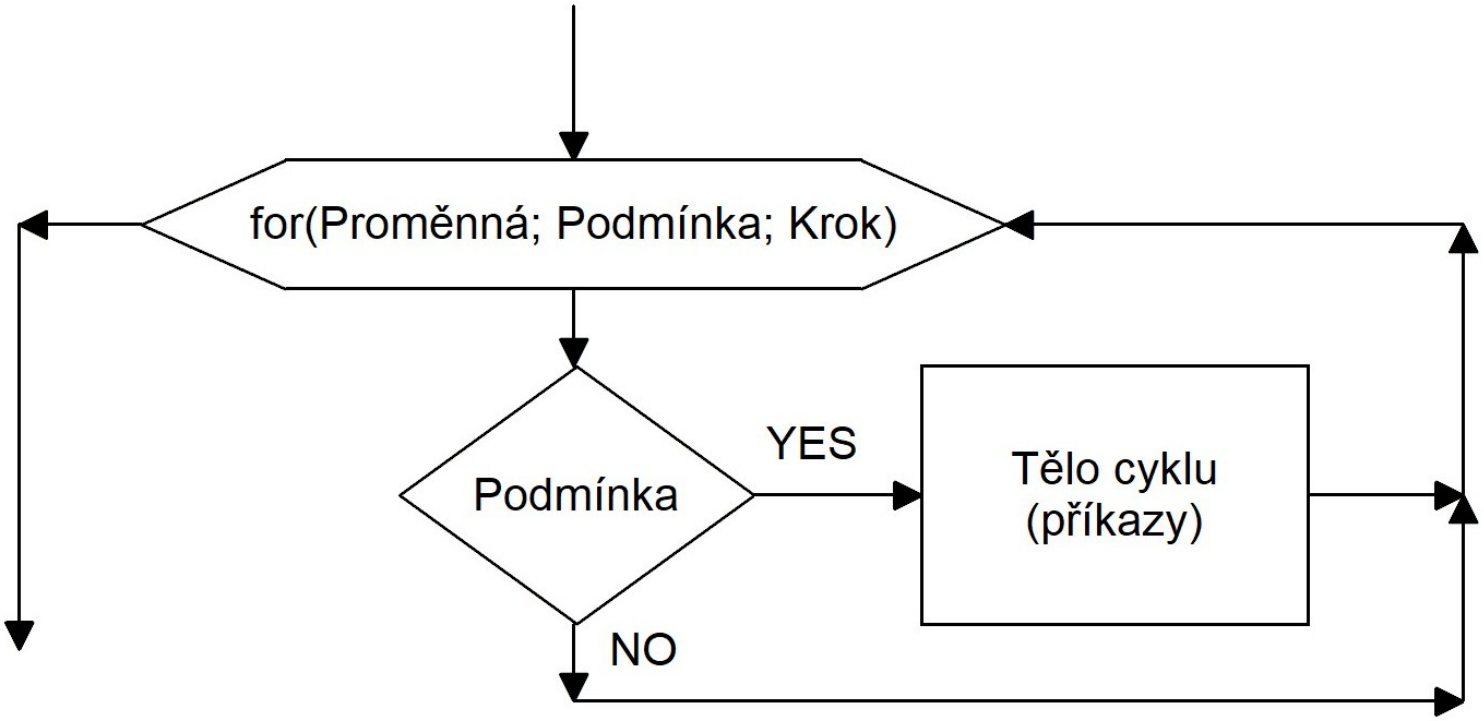
Proměnná - řídící proměnná cyklu a její počáteční hodnota
Podmínka - podmínka, která definuje konec cyklu
Step - velikost přírůstku hodnoty řídící proměnné cyklu

Varianta cyklu For s předčasným ukončením cyklu (break):



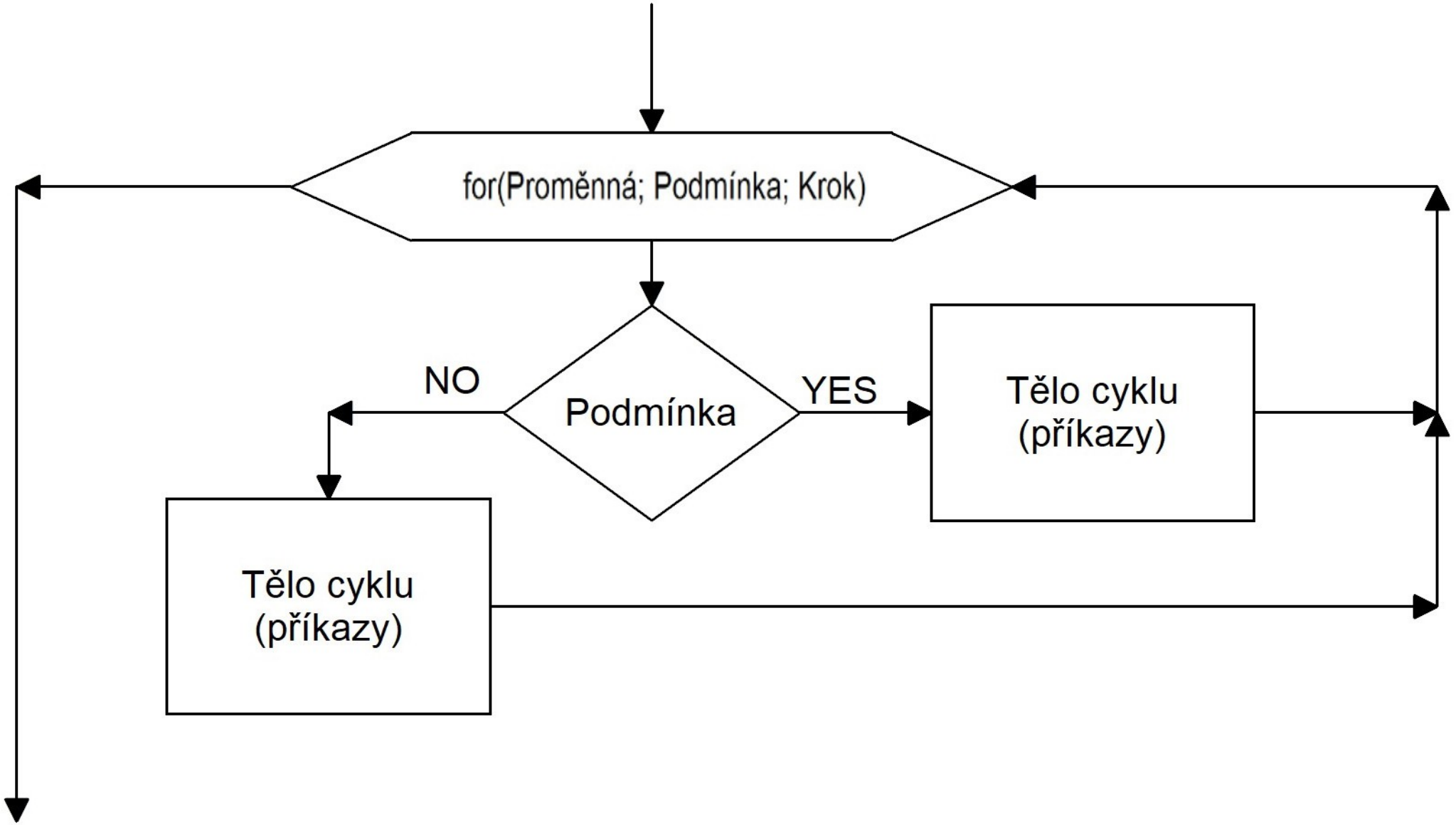
***Break** předčasně ukončuje cyklus For – dříve, než je ukončen vyhodnocení podmínky!*

Varianta cyklu For s ukončením smyčky cyklu (continue):



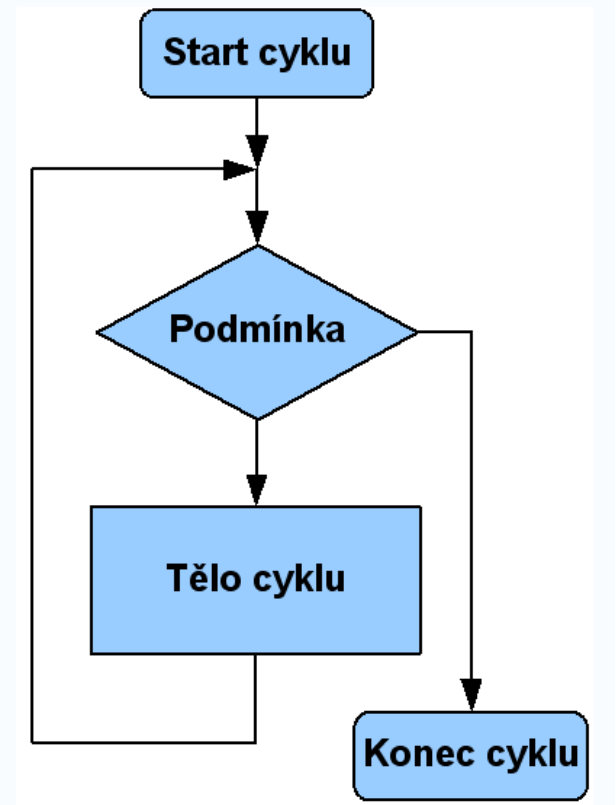
Continue *ignoruje všechny následující příkazy v aktuální smyčce cyklu a pokračuje další smyčkou!*

Varianta cyklu For se dvěma oddělenými těly cyklu:



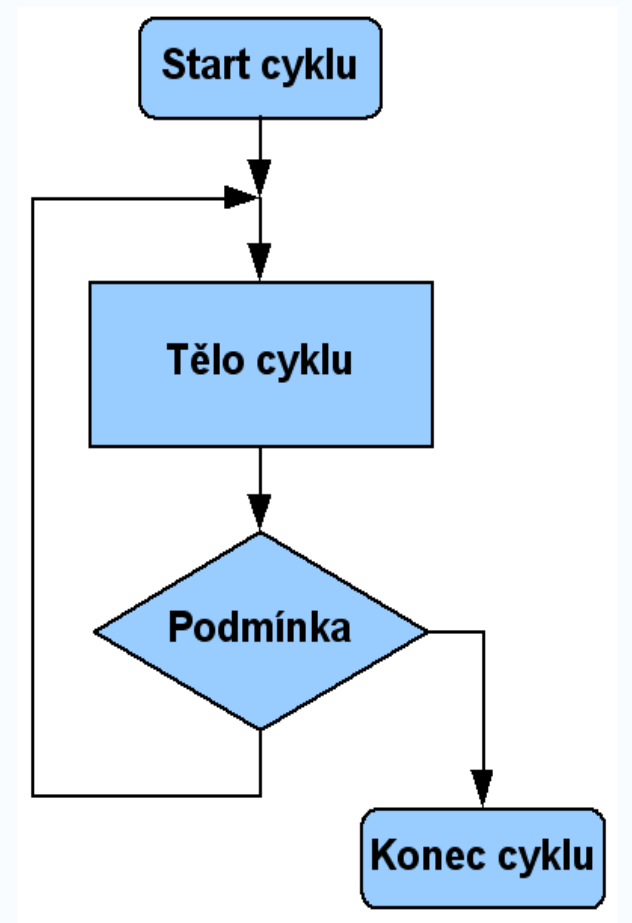
Cyklus While

Podmínka, která rozhodne, jestli bude provedena (další) smyčka (loop), je na začátku cyklu.



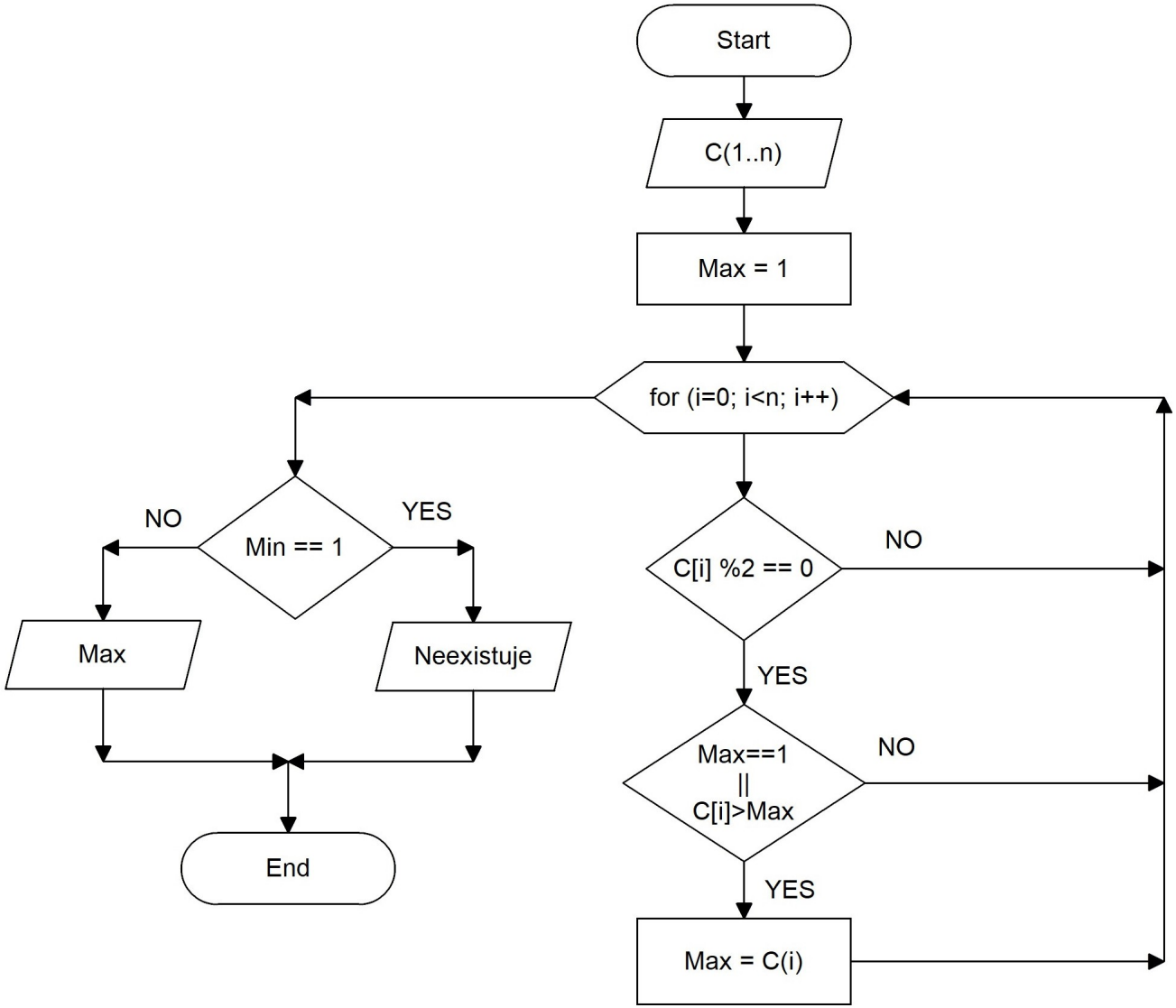
Cyklus Repeat

Podmínka, která rozhodne, jestli bude provedena další smyčka (loop), je na začátku cyklu.



Úkol: Vytvořte vývojový diagram pro nalezení největšího sudého čísla

Úkol: Vytvořte vývojový diagram pro nalezení největšího sudého čísla



Domácí úkol:

Vytvořte vývojový diagram algoritmu pro výpočet průměrného čísla z pole čísel.

Konec...

Vaše dotazy? ….