



Jak brouček sekal souček

aneb cykly a všechno co s tím souvisí

© Leonard Walletzký, ESF MU, 2003

Opakování

- Větvení algoritmu – if
- Rozhodněte, který ze zadaných vektorů je delší
 - funce *length(<vektor>)*
 - vrací velikost vektoru (počet prvků)

Program

```
v=input('1. vektor:');  
u=input('2. vektor:');  
if length(v)>length(u)  
    disp('v je delsi')  
else  
    disp('u je delsi')  
end
```

Opakování 2

- Rozhodněte, zda má matice více řádků, sloupců, nebo obojího stejně
 - funkce *size(<matice>)*
 - vrací **vektor** [ř s]
 - lze použít i indexy
 - `vel=size(A)`
 - `vel(1)`.....počet řádků
 - `vel(2)`.....počet sloupců

Řešení

```
A=input('Zadej matici:');  
vel=size(A);  
if vel(1)==vel(2)  
    disp('obojiho stejne')  
else  
    if vel(1)>vel(2)  
        disp('radku je vic')  
    else  
        disp('sloupcu je vic')  
    end  
end  
end
```

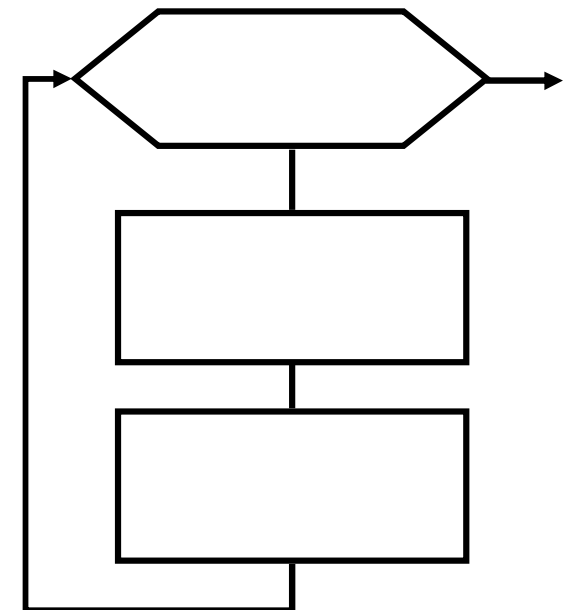
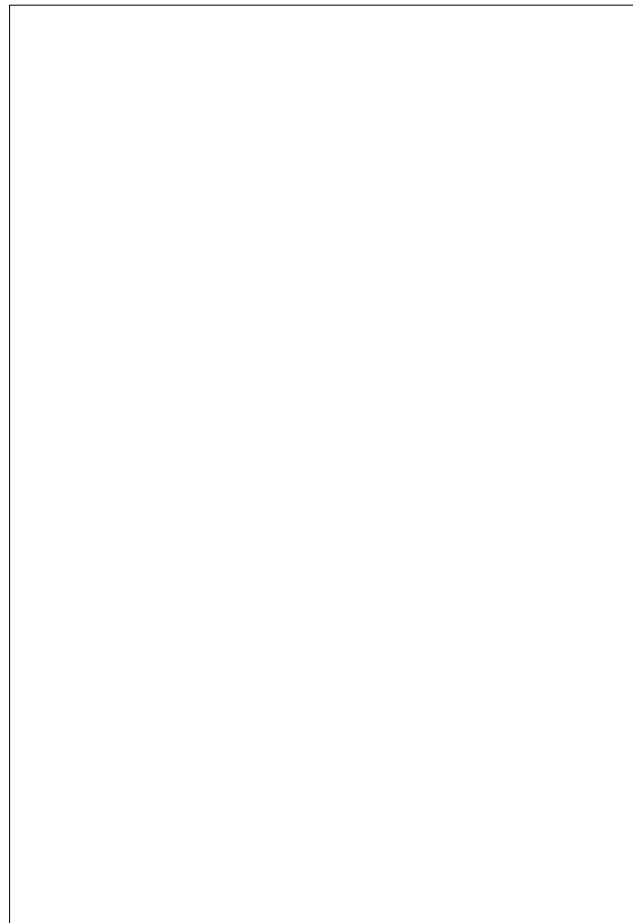
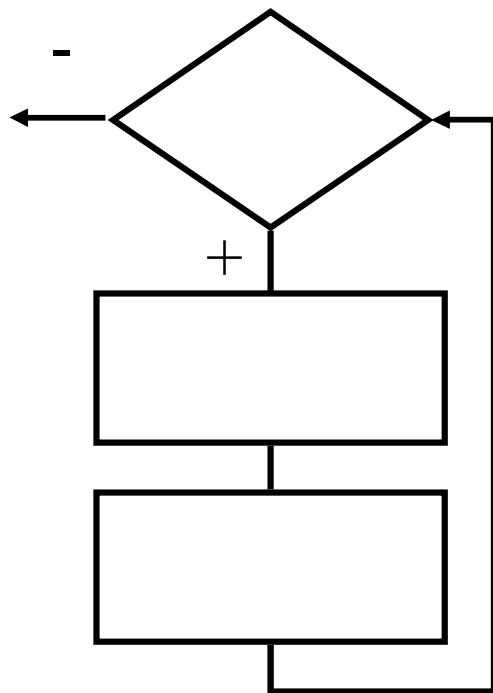
Cyklus

- Umožňuje vrátet se v algoritmu zpět

S podmínkou na začátku

S podmínkou na konci

Se známým počtem průchodů



Cyklus s podmínkou na začátku

- Klíčové slovo *while*
- syntaxe:
while <podmínka>
 <blok příkazů>
end
- po skončení cyklu (po nesplnění podmínky) se skočí na **první** řádek za *end*

Příklad

- Funkce pro výpočet faktoriálu čísla n

```
n=input('Zadej cislo:');
```

```
f=1;
```

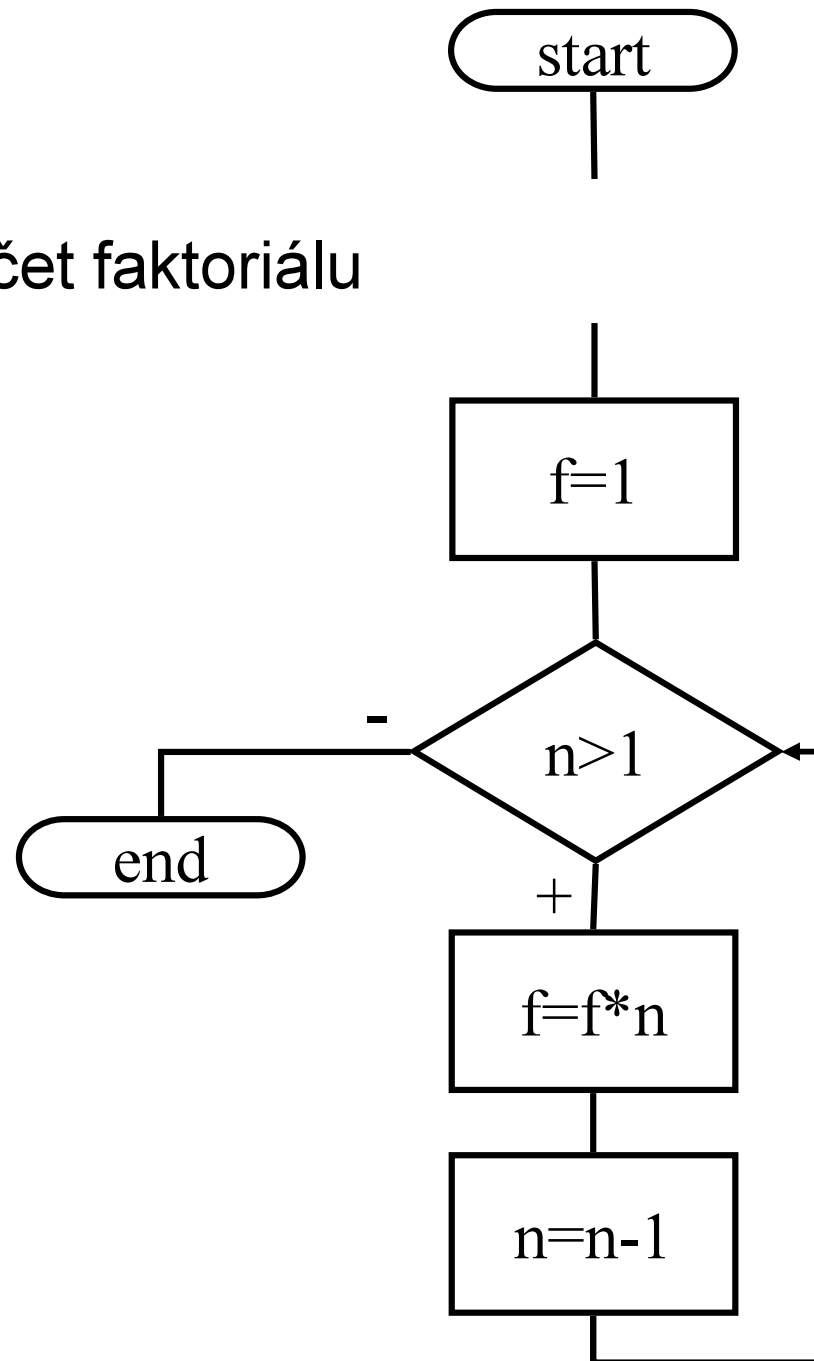
```
while n>1
```

```
    f=f*n;
```

```
    n=n-1;
```

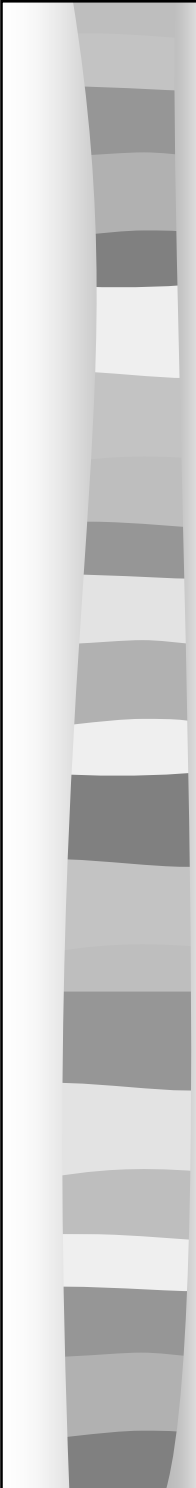
```
end
```

```
disp(f)
```



Trasovací tabulka

Počet průchodů	n	f
0	4	1
1	3	4
2	2	12
3	1	24



Cyklus se známým počtem průchodů

- Klíčové slovo *for*
- syntaxe:
for <proměnná>=<interval>
 <blok příkazů>
end
- přiřazuje postupně do <proměnné> hodnoty z <intervalu>, po jejich vyčerpání skočí na první příkaz za *end*

Modifikace výpočtu faktoriálu

- Výpočet faktoriálu pomocí cyklu *for*

```
n=input('Zadej cislo:');
```

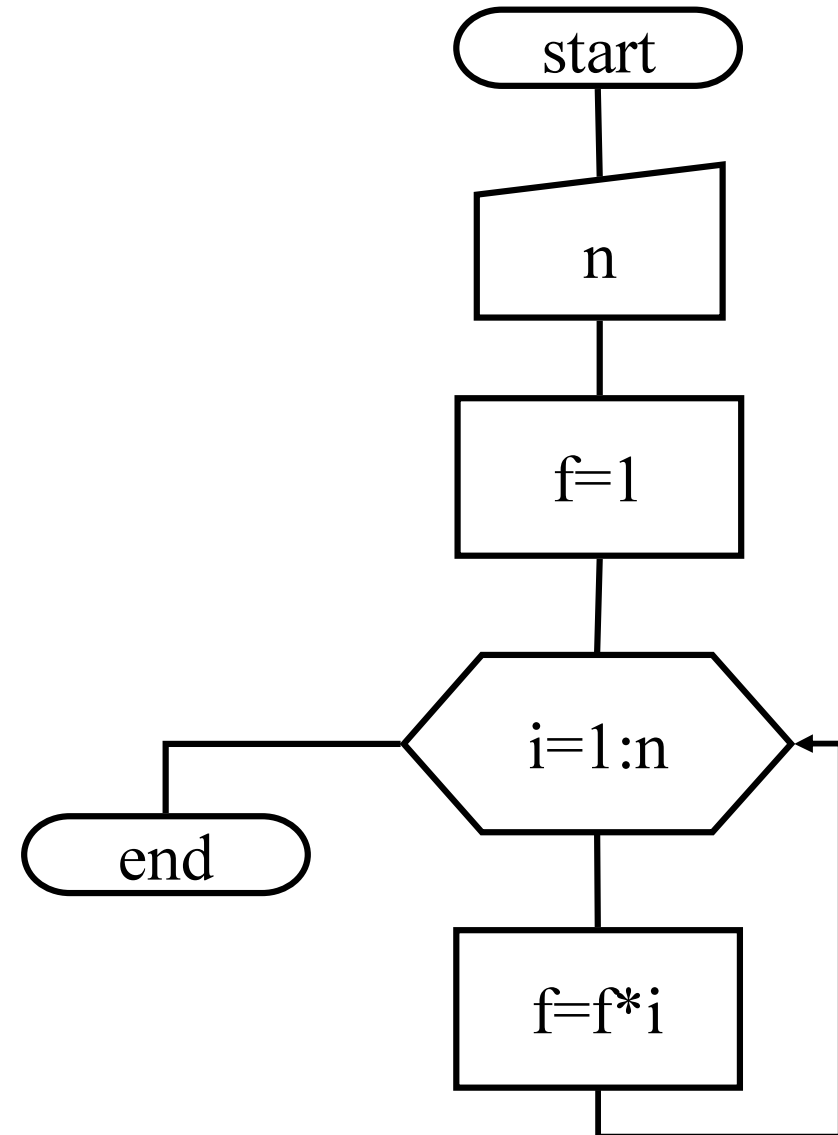
```
f=1;
```

```
for i=1:n
```

```
    f=f*i;
```

```
end
```

```
disp(f)
```



Trasovací tabulka

Počet průchodů	i	f
0	-	1
1	1	1
2	2	2
3	3	6
4	4	24

Příklady

- Napište program, který sečte prvky vektoru

```
v=input('Zadej vektor');
```

```
vel=length(v);
```

```
s=0;
```

```
for i=1:vel
```

```
    s=s+v(i);
```

```
end
```

```
disp(s)
```

Příklad

- Napište program, který najde první nulový prvek ve vektoru.
- Nejprve předpokládejte, že tam nulový prvek vždy je
- Potom přepište program tak, aby počítal i s vektorem bez nulového prvku

Řešení

```
v=input('Zadej vektor:');
```

```
i=1;
```

```
while v(i)~=0
```

```
    i=i+1;
```

```
end
```

```
disp(i)
```

