

# Písemná přijímací zkouška z informatiky

2016/2017

univerzitní číslo:

Za každý vypracovaný úkol můžete získat až 25 bodů. Úkoly můžete řešit v libovolném pořadí. Na vypracování máte 60 minut. Hodně štěstí.

číslo otázky	získané body
1 (matematika)	
2 (algoritmus)	
3 (operační systém)	
4 (automaty)	
CELKEM	

## Zadání:

- Označme  $p_n$  ( $n \in \mathbb{N}$ ) počet palindromů nad abecedou  $A = \{a, b, c\}$ , ve kterých se každý znak abecedy vyskytuje ve stejném počtu.
  - Určete vztah pro výpočet  $p_n$ ,  $n \in \mathbb{N}$ .
  - Určete nejmenší přirozené  $n$  takové, že  $p_n$  je prvočíslo.  
(pozn.: Palindrom nad danou abecedou je slovo obsahující pouze znaky dané abecedy, které se čte stejně zepředu, jako zezadu.)
- V libovolném jazyce popište algoritmus, který spočítá  $n$ -tou mocninu čísla  $x$  bez použití funkce  $x^n$ . Na vstupu je základ  $x$  (celé číslo) a mocnina  $n$  (celé nezáporné číslo).  
(pozn.: Jazykem se myslí čeština, srozumitelný symbolický zápis či programovací jazyk. Nejvíce je ohodnocen korektní zápis v programovacím jazyce.)
- Vysvětlete pojem *uváznutí* (*deadlock*) a napište podmínky vzniku a metody prevence.
- Sestrojte deterministický konečný automat rozpoznávající jazyk  $L = \{w \in \{a, b\}^* \mid \text{obsahuje řetězec baba}\}$ .

### Příklady řešení:

ad 1)

a) Nejprve si uvědomíme, že palindrom je jednoznačně určen svou první polovinou znaků (druhou doplníme tak, aby se četl stejně zpředu jako zezadu). Jestliže navíc požadujeme, aby každý ze tří znaků abecedy byl zastoupen ve stejném počtu, musí být délka palindromu  $n$  dělitelná 6. První polovina palindromu tak bude obsahovat  $n/6$  znaků  $a$ ,  $n/6$  znaků  $b$  a  $n/6$  znaků  $c$  a to v libovolném pořadí, tedy jejich počet je roven počtu permutací s opakováním z  $n/6$  znaků  $a$ ,  $n/6$  znaků  $b$  a  $n/6$  znaků  $c$ .

Odtud  $p_n = \frac{(n/2)!}{(n/6)!(n/6)!(n/6)!}$  pro  $n$  dělitelné 6 a  $p_n = 0$  jinde (tj. pro  $n$ , které není dělitelné 6).

b) Vzhledem k výše uvedenému je zřejmé, že žádné takové  $n$  neexistuje (tj.  $\forall n \in \mathbb{N}$  není  $p_n$  prvočíslo).

ad 2)

```
program nmocnina;
var x, n, i, mocnina: integer;
    b:boolean;

begin
  write ('Zadej základ: ');
  readln (x);
  b:=true;
  repeat
    write ('Zadej mocninu: ');
    readln (n);
    if n<0 then b:=false else b:=true;
  until b;
  mocnina:=1;
  if n <> 0 then
    for i:=1 to n do
      mocnina:=mocnina*x;
  writeln ('Výsledek je: ', mocnina);
  readln;
end.
```

ad 3)

uváznutí = dva (a více) procesy (vzájemně) čekají na spuštění události, které může učinit pouze jeden z čekajících procesů (první událost může být dokončena až po dokončení druhé a zároveň druhá událost může být dokončena až po dokončení první)

podmínky (Musí platit všechny současně!):

- Prostředek lze přidělit právě jednomu procesu. Po jeho přidělení jej nemůže používat jiný – vzájemně se vyloučí.
- Proces, který má přidělen prostředek (drží je), si může požádat o další. O prostředky si může požádat postupně.
- Prostředek nemůže být procesu násilně odebrán.
- Dojde k cyckickému čekání procesů.

metody prevence (Principem je vždy porušení alespoň jedné z podmínek výše.):

- Virtualizace prostředku s výlučným přístupem
- Nutnost alokace všech prostředků na začátku
- a další

