

Riešenie úloh

1. Táto časť kódu v tele cyklu:

```
sucet := pole[n-j+1] + pole[j];  
pole[j] := sucet - pole[j];  
pole[n-j+1] := sucet - pole[j];
```

vymení obsahy dvoch prvkov poľa s indexami j a $n-j+1$. Keďže while-cyklus sa vykoná len pre polovicu prvkov poľa, pôvodné pole s hodnotami (1, 2, 3, 4, 5, 6, 7) sa zmení na (7, 6, 5, 4, 3, 2, 1).

Teda program vypíše:

7 6 5 4 3 2 1

2. Úloha má niekoľko správnych riešení:

- a) [*2] [*2] [÷3+1] [÷3+1] [*2] [*2] [*2] [÷3+1]
- b) [*2] [÷3+1] [*2] [*2] [÷3+1] [*2] [*2] [÷3+1]
- c) [*2] [÷3+1] [*2] [÷3+1] [*2] [*2] [*2] [÷3+1]
- d) [÷3+1] [*2] [*2] [*2] [÷3+1] [*2] [*2] [÷3+1]
- e) [÷3+1] [*2] [*2] [÷3+1] [*2] [*2] [*2] [÷3+1]

Riešime postupným konštruovaním stromu: v koreni stromu je 5, strom sa rozvetvuje na dva smery: pre [*2] a [÷3+1]. V každom rozvetvení sa opäť rozvetvuje. Rozvetvujeme, kým sa nedosiahne číslo 9.

3. Postupne aplikujeme jedno z pravidiel podľa písmena pod kurzorom:

MAMKA
KAMKA
KMMKA
KMKKA
KMKAA
KMAAA
KKAAA
KKMAA
KKMMA
KKMMM

Výsledná oprava slova je **KKMMM**.

4. Program v Pascale:

```
program Pyramida;  
const n = 5;  
var i, j, cislo: integer;  
begin  
  cislo := n * (n+1) div 2;  
  for i := 1 to n do  
    begin  
      for j := 1 to i do  
        begin  
          write(cislo, ' ');  
          cislo := cislo - 1;  
        end;  
      writeln();  
    end;  
end.
```

alebo to isté v Pythone:

```
n = int(input('zadaj n: '))  
cislo = n * (n+1) // 2  
for i in range(n):  
  for j in range(i+1):  
    print(cislo, end=' ')  
    cislo -= 1  
  print()
```