

## Prijímacie pohovory z informatiky

1. Nasledovný program pracuje s jednorozmerným  $n$ -prvkovým poľom celých čísel:

```

for i := 1 to n do
  pole[i] := i;
j := n;
while n-j+1 < j do
  begin
    sucet := pole[n-j+1] + pole[j];
    pole[j] := sucet - pole[j];
    pole[n-j+1] := sucet - pole[j];
    j := j - 1;
  end;
for i := 1 to n do
  write(pole[i], ' ');

```

Zistite, čo sa vypíše pre  $n=7$ .

2. V múzeu výpočtovej techniky sme objavili historickú celočíselnú kalkulačku, ktorá až na dve tlačidlá vôbec nefunguje. Jedno tlačidlo má označenie  $[*2]$  a po jeho zatlačení sa číslo na displeji vynásobí 2. Druhé tlačidlo  $[\div 3+1]$  vydolí číslo na displeji tromi a pripočíta 1 - delenie je celočíselné, t.j. zvyšok po delení sa zanedbá. Na začiatku je na displeji číslo 5. Postupným stláčaním tlačidiel môžeme na displeji dosiahnuť rôzne výsledky. Zistite, ako dosiahneme číslo 9 na čo najmenší počet stláčaní tlačidiel.

3. Písmenkový robot dokáže opravovať napísaný text. Pracuje podľa takto definovaných pravidiel: ak je jeho kurzor nad písmenom  $X$ , prerobí ho na písmeno  $Y$  a posunie kurzor o jeden znak vpravo. Takéto pravidlo by sme zapísali  $(X, Y, \rightarrow)$ . Zrejme zápis pravidla  $(X, Z, \leftarrow)$  bude označovať, že ak je kurzor nad písmenom  $X$ , prepíše ho na písmeno  $Z$  a kurzor presunie o jeden znak vľavo (pred práve opravené písmeno). Písmenkovejmu robotu môžeme pripraviť ľubovoľne veľa takýchto pravidiel a on potom opravuje zadané texty, kým má na písmeno pod kurzorom nejaké pravidlo. Inak spokojne skončí s tým, že už urobil všetko, čo bolo treba.

Zadali sme takéto tri pravidlá:  $(A, M, \rightarrow)$ ,  $(M, K, \rightarrow)$ ,  $(K, A, \leftarrow)$  a nechali sme mu opraviť slovo **MAMKA**. Zistite, ako ho tento robot opravil, t.j. ako sa toto slovo zmenilo po skončení práce robota.

4. Napíšte program, ktorý vypíše takúto  $n$ -poschodovú pyramídu: v prvom riadku je jedno číslo, v druhom dve, atď. v poslednom riadku je  $n$  čísel. Posledný riadok pritom obsahuje čísla od 1 do  $n$  v opačnom poradí. Predposledný riadok obsahuje  $n-1$  nasledovných čísel (opäť v opačnom poradí). Každý predchádzajúci riadok obsahuje stále väčšie a väčšie čísla. Napr. pre  $n=3$  číselná pyramída má tvar

```

6
5 4
3 2 1

```

Pre  $n=4$ :

```

10
9 8
7 6 5
4 3 2 1

```

Váš program by mal pre ľubovoľné zadané  $n$  postupne vypísať  $n$  riadkov takýchto čísel. Môžete použiť ľubovoľný programovací jazyk, ktorý ste používali na strednej škole.