

Štvrté cvičenie

Základné požiadavky na odovzdanie. Odovzdávate:

- JEDEN súbor obsahujúci celý zdrojový kód,
- v jazyku C (ANSI C podľa prednášok),
- s názvom a v štruktúre podľa zverejnených inštrukcií (MSTeams)
- KOMPILOVATELNÝ a spustiteľný (jediný zásah hodnotiteľa je v vo funkcii MAIN povoľovať a zakazovať volania jednotlivých úloh).
- Zadané nespĺňajúce tieto kritéria nebudú hodnotené a dostanú automaticky nulu
- Odovzdané zadané spĺňajúce vyššie uvedené predpoklady je počítané ako **účasť na cvičení**.

Ďalšie univerzálne požiadavky.

- Programy musia komunikovať. Ak program očakáva vstup, musí oznamovať aký vstup sa očakáva. Ak vypisuje výsledok, musí vypisovať zrozumiteľný oznam (napr. čo za hodnotu to vypisuje).
- Formátovanie zdrojového kódu by malo zodpovedať približne príkladom z prednášok. Odsadzovanie textov je základ. Príklad dobrého a zlého formátovania sú v prednáške číslo dva na konci.
- Tieto požiadavky sú platné až do konca semestra, pre zadania, písomky aj skúšku, všade kde sa odovzdáva ucelený program alebo funkcia.

Na riešenie žiadneho príkladu nie povolené použiť pole! Sme sa to neučili, ale hlavne to nie je POTREBNÉ.

Úloha prvá.

(1.5 bodu) Floydov trojuholník.

Napište program, ktorý do súboru **floyd.txt** vypíše Floydov trojuholník. Program z klávesnice načíta celé číslo **n** z intervalu $<1, 10>$. Floydov trojuholník má **n** riadkov, kde každý nasledujúci riadok má vždy o jeden prvok viac ako predchádzajúci riadok. Prvky trojuholníka tvorí **rastúca postupnosť čísel** začínajúc číslom 1. Každý riadok je ukončený znakom konca riadku a medzi jednotlivými číslami v riadku je vždy jedna medzera. Ak je na vstupe načítané iné číslo ako z definovaného intervalu, tak program **NEVYTvorí ŽIADNY SÚBOR** a na **OBRAZOVKU** vypíše správu: Nespravny vstup a skončí.

Príklad 1:

```
Zadajte pocet riadkov Floydovho trojuholnika od 1 do 10: 11↵
Nespravny vstup↵
```

Príklad 2:

```
Zadajte pocet riadkov Floydovho trojuholnika od 1 do 10: 6↵
```

Vznikne súbor **floyd.txt** a bude obsahovať 6 riadkov:

```
1↵
2 3↵
4 5 6↵
7 8 9 10↵
```

```
11 12 13 14 15↵
16 17 18 19 20 21↵
```

Kľúčom k programu je zostaviť dva cykly (riadky a stĺpce) a použiť jednu pomocnú premennú na priebežné počítanie.

Úloha druhá.

(1.5 bodu) Hviezdičky.

Napište program, ktorý načíta súbor **rozmary.txt**. V súbore sa predpokladajú dve čísla (nemusia byť – môže byť žiadne, jedno, dve alebo viac – použite povinne `fscanf()` a testujte číslo ktoré vracia – pozrite si prednášku – tam je podobný príklad – vám stačí ak funkcia načíta dve čísla).

Tieto dve CELÉ čísla sú ROZMERY v poradí RIADKY, STĹPCE.

Vytlačte na OBRAZOVKU obrátený trojuholník.

Príklad 1: predpokladajme vstup v súbore: 5 6

```
*****
*****
*****
*****
*****
```

Teda máme 5 riadkov po 6 hviezdičiek.

Príklad 2: predpokladajme vstup v súbore: 4 4

```
****
****
****
****
```

Úlohy na tréning:

1. Napište program, ktorý načíta číslo n . Ak je $n < 1$ alebo $n > 15$, program vypíše chybovú hlášku `Cislo nie je z daneho intervalu` a skončí. V opačnom prípade program vypíše n riadkov, kde každý bude obsahovať číslo riadku, dvojbodku, medzeru a čísla oddelené medzerou. V prvom riadku budú za dvojbodkou vypísané čísla od 1 po n . V každom nasledovnom riadku bude vždy o jedno číslo menej. Všetky čísla vypisujte na 2 miesta (zarovnávanie – šírka čísla).

Zadajte počet riadkov od 1 do 15: 4↵

Ukázkový výstup:

```
1:  1  2  3  4↵
2:  1  2  3↵
3:  1  2↵
4:  1↵
```

2. Napište program, ktorý načíta číslo n . Ak je $n < 1$ alebo $n > 15$, program vypíše chybovú hlášku `Cislo nie je z daneho intervalu` a skončí. V opačnom prípade program vypíše n riadkov, kde každý bude obsahovať číslo riadku, dvojbodku, medzeru a čísla oddelené medzerou. V prvom riadku budú za

dvojbodkou vypísané čísla od n po 1. V každom nasledovnom riadku bude vždy o jedno číslo menej. Všetky čísla vypisujte na 2 miesta.

Zadajte počet riadkov od 1 do 15: 4↵

Ukážkový výstup:

```
1:  4  3  2  1↵
2:  3  2  1↵
3:  2  1↵
4:  1↵
```

3. Napíšte program, ktorý zo štandardného vstupu načíta celé číslo n a vypíše všetky také čísla k ($k=1,2,\dots,n$), pre ktoré platí, že súčet $1+2+\dots+k$ je deliteľný číslom 3. Pomôcka: ľahšie riešenie používa dva cykly. Lepšie riešenie sa dá urobiť na jeden cyklus. Napríklad pre $n=5$ sa vypíšu čísla 2,3,5.
Lebo: pre $k=2$ je súčet $1+2$ deliteľný 3. pre $k=3$ je súčet $1+2+3$ deliteľný 3, pre $k=4$ súčet $1+2+3+4$ nie je deliteľný 3 a pre $k=5$ je súčet $1+2+3+4+5$ deliteľný 3.

4. Napíšte program, ktorý načíta číslo n . Ak je $n < 1$, $n > 15$ alebo je n párne číslo, program vypíše chybu `Zly vstup` a skončí. Ak bude program pokračovať, zo znakov `*` a ` ` (medzera) nakreslí rovnoramenný trojuholník s výškou n .

Zadajte neparne číslo od 1 do 15: 5↵

Ukážkový výstup:

```
*----↵
**----↵
***--↵
****-↵
*****↵
****-↵
***--↵
**----↵
*----↵
```

5. Napíšte program, ktorý načíta číslo n . Ak je $n < 1$, $n > 15$ alebo je n párne číslo, program vypíše chybu `Zly vstup` a skončí. Ak bude program pokračovať, zo znakov `*` a ` ` nakreslí hviezdu o veľkosti $n \times n$.

Zadajte neparne číslo od 1 do 15: 9↵

Ukážkový výstup:

```
*---*---*↵
-*---*---↵
--*-*-*-↵
---*---↵
*****↵
---*---↵
--*-*-*-↵
-*---*---↵
*---*---*↵
```

6. Napíšte program, ktorý načíta dve celé čísla n , v oddelé medzerou. Ak je $n < 1$, $n > 15$, n je párne číslo, alebo v nie je z intervalu $<1, 5>$, program vypíše chybu Zlý vstup a skončí. Ak bude program pokračovať, zo znakov '*' a '-' nakreslí v obrázkov hviezdy pod seba o veľkosti $n \times n$.

Ukázkový vstup: 5 2 ↵

Ukázkový výstup:

```
*---*---*↵
-*---*---*↵
--*-*-*---*↵
---***---*↵
*****---*↵
---***---*↵
--*-*-*---*↵
-*---*---*↵
*---*---*↵
*---*---*↵
-*---*---*↵
--*-*-*---*↵
---***---*↵
*****---*↵
---***---*↵
--*-*-*---*↵
-*---*---*↵
*---*---*↵
```

7. Napíšte program, ktorý načíta dve celé čísla n , s oddelených medzerou. Ak je $n < 1$, $n > 15$, n je párne číslo, alebo s nie je z intervalu $<1, 5>$, program vypíše chybu Zlý vstup a skončí. Ak bude program pokračovať, zo znakov '*' a '-' nakreslí s obrázkov hviezdy vedľa seba o veľkosti $n \times n$.

Ukázkový vstup:

9 2 ↵

Ukázkový výstup:

```
*---*---*---*---*---*↵
-*---*---*---*---*---*↵
--*-*-*---*---*---*---*↵
---***---*---*---*---*↵
*****---*---*---*---*↵
---***---*---*---*---*↵
--*-*-*---*---*---*---*↵
-*---*---*---*---*---*↵
*---*---*---*---*---*↵
```