

Úloha 5.1: Napíšte program na výpočet obsahu a obvodu obdĺžnika. Vytvorte funkciu nacitaj, ktorá načíta a pomocou argumentov vráti dve reálne čísla. Ďalej vytvorte funkciu vypocitaj, ktorá ako argumenty dostane dĺžky strán obdĺžnika a prostredníctvom argumentov vráti obsah a obvod obdĺžnika. V hlavnom programe volajte funkciu nacitaj na načítanie rozmerov obdĺžnika a funkciu vypocitaj na výpočet obsahu a obvodu obdĺžnika. Program vypíše obsah a obvod obdĺžnika zaokrúhlený na tri desatinné miesta.

Ukážka vstupu:

3.5 4.75

Výstup pre ukážkový vstup:

Obsah: 16.625

Obvod: 16.500

Úloha 5.2: Napíšte funkciu **maximum**, ktorá dostane na vstupe adresy na dve celé čísla a vráti adresu na celé číslo, ktoré je väčšie. Program načíta zo štandardného vstupu dve celé čísla. Zavolá funkciu maximum, odovzdá jej adresy premenných, kde sú načítané hodnoty uložené a vypíše väčšie číslo. Pozor funkcia maximum vráti na výstupe adresu na premennú, kde je uložené väčšie číslo.

Ukážka vstupu:

5 8

Výstup pre ukážkový vstup:

Cislo 8 je vacsie.

Úloha 5.3: Napíšte program, ktorý zo štandardného vstupu načíta reálne čísla a, b, c. Pomocou funkcie kvadraticka_rovnica určí korene kvadratickej rovnice $ax^2 + bx + c = 0$ (ako riešiť kvadratickú rovnicu s jednou neznámou nájdete napr. na https://sk.wikipedia.org/wiki/Kvadratick%C3%A1_rovnica). Funkcia kvadraticka_rovnica vráti počet koreňov a ich hodnoty. Program návratové hodnoty vypíše na konzolu. Môžete predpokladať, že koeficient a je rôzny od 0 (čo ak by sa rovnal 0, ako to ovplyvní program?). Odmocninu počítajte pomocou funkcie sqrt z knižnice math.

Ukážka vstupu:

1 5 6

Ukážka vstupu:

2 -4 2

Ukážka výstupu:

Pocet korenov kvadratickej rovnice je: 2
Prvy koren ma hodnotu: -2.00000
Druhy koren ma hodnotu: -3.00000

Ukážka výstupu

Pocet korenov kvadratickej rovnice je: 1
Prvy koren ma hodnotu: 1.00000
Druhy koren ma hodnotu: 0.00000

Úloha 5.4: Napíšte procedúru `vymen_ukazovatele`, ktorá dostane na vstupe adresy dvoch ukazovateľov na celočíselné premenné (`a` a `b` s hodnotami 10 a 14) a následne vymení ich hodnoty t.j. adresy premenných, na ktoré ukazujú. Program na konzolu vypíše stav (adresy na ktoré ukazujú ukazovatele a ich hodnoty) pred a po zavolaní procedúry `vymen_ukazovatele`.

Ukážka výstupu:

```
p_a: 000000000064FE2C 10
p_b: 000000000064FE28 14
p_a: 000000000064FE28 14
p_b: 000000000064FE2C 10
```

Úloha 5.5: Napíšte funkciu `int delitele(int x[], int pocetx, int y[], int k)`, ktorá z poľa `x` prekopíruje do poľa `y` všetky delitele čísla `k`, v poradí v akom sa nachádzajú v poli `x` a vráti počet prvkov poľa `y`. Argument `pocetx` určuje počet prvkov poľa `x` (alokované staticky). Môžete predpokladať, že argument `y`, bude mať dostatočnú veľkosť pre všetky delitele čísla `k`, ktoré sú v poli `x`.

Ukážka volania:

```
x = {4, 7, 10, 2, 3, 9, 6, 5, 8, 12}
pocetx = 10
pocety = delitele(x, pocetx, y, 24); // volanie funkcie
pocety: 6 // vypis vysledku
y: {4, 2, 3, 6, 8, 12}
```

Úloha 5.6: Program na začiatku načíta rozmery poľa `m` (nezáporné celé číslo), vo funkcii `alokuj_1D_pole(...)` sa dynamicky alokuje pole, následne sa celé pole naplní číslami zadaných z klávesnice. Program následne načítané pole vykreslí do konzoly prostredníctvom funkcie `vypis_1D_pole(...)`. Následne vo funkcii `main()` sa vypíše maximálna a minimálna hodnota z daného poľa, pričom príslušné pozície, kde sa dané hodnoty nachádzajú sa nájdu prostredníctvom funkcie `maximum_poz(...)` a `minimum_poz(...)`. Pred skončením programu uvoľníte pamäť prostredníctvom funkcie `uvolni()`.

Ukážka vstupu:

```
5
3
1
2
3
4
```

Výstup pre ukážkový vstup:

```
3,1,2,3,4,
```