

# Problém set 3 - C- polia

V tomto zadaní musíš vypracovať aspoň **tri** projekty z:

- Štatistika
- Cézarova šifra
- Bargraph
  - Vertikálny
  - \*\*Horizontálny +5 bodov navyše
- \*\*\*Míny +10 bodov navyše

## Štatistika

Zo stránky predmetu si stiahni *template* programu s názvom *statistika.c*. Template momentálne dokáže načítať ľubovoľný súbor, ktorý mu ponúkneš ako vstupný (*command line argument*). Tento súbor by mal však obsahovať len **kladné celé čísla**. Po spustení program načíta zvolený súbor do poľa s názvom (*unsigned int*) **arr**. Dĺžka tohto poľa je zaznamenaná v premennej **arr\_length**.

Časti programu sú nedopísané a zakomentované. Tvojou úlohou je dopísať tieto dve časti programu.

Špecifikácia programu:

- Dopíšete do programu *statistika.c* časť s názvom kumulatívna suma, ktorý do nového poľa s názvom *cumSum* uloží kumulatívnu sumu z hodnôt v poli *arr*

Príklad:

Nech je v poli *arr* nasledovné

*arr*[0] = 5; *arr*[1] = 6; *arr*[2] = 9; *arr*[3] = 10; atď.

Potom v poli *cumSum* má byť

*cumSum*[0] = 5; *cumSum*[1] = 11; *cumSum*[2] = 20; *cumSum*[3] = 30; atď

- Dopíšete do programu *statistika.c* časť s názvom priemer, ktorý spočíta priemer hodnôt v poli *arr*

Príklad:

Nech je v poli *arr* nasledovné

*arr*[0] = 1; *arr*[1] = 8; *arr*[2] = 4; *arr*[3] = 7; atď.

Potom priemer je  $(1 + 8 + 4 + 7) / 4$

Predpokladaný výstup programu je nasledovný, ak použijeme ako vstupný súbor *number.txt* (stiahnuteľný zo stránky predmetu). V tomto súbore sa v riadkoch nachádzajú hodnoty 5, 10, 15, 7, 9, 5. To znamená do premennej sa načítajú tieto hodnoty tak, že  $arr[0] = 5$ ,  $arr[1] = 10$  atď.

Výstup programu:

```
5      vypíše a do cumSum uloži: arr[0]
15     vypíše a do cumSum uloži: arr[1] + arr[0]
30     vypíše a do cumSum uloži: arr[2] + arr[1]
37     vypíše a do cumSum uloži: arr[3] + arr[2]
46     vypíše a do cumSum uloži: arr[4] + arr[3]
51     vypíše a do cumSum uloži: arr[5] + arr[4]
8.50   vypíše priemer hodnôt z poľa arr
```

## Caesarova šifra

Pravdepodobne Caesar (áno, že Caesar) zvykol „šifrovať“ dôverné správy presunutím každého písmena v liste o niekoľko miest. Napríklad by mohol písať A ako B, B ako C, C ako D,... a cyklicky Z ako A. A tak, aby niekto povedal AHOJ, tak Caesar by mohol napísať BIPK. Po prijatí takýchto správ od Caesara by ich príjemcovia museli „dešifrovať“ posunutím písmen v opačnom smere o rovnaký počet miest.

Utajenie tohto „kryptosystému“ sa spoliehalo iba na Caesara a na príjemcov, ktorí poznali tajomstvo, počet miest, o ktoré Caesar presunul svoje listy (napr. 1). Na moderné štandardy nie sú Cézarove šifry príliš bezpečné, ale hej, ak si snáď prvý, kto ich vynájde, je to celkom bezpečné!

Nešifrovaný text sa všeobecne nazýva nešifrovaný text. Zašifrovaný text sa všeobecne nazýva šifrovaný text. A použité tajomstvo sa nazýva kľúč.

Nešifrovaný text	A	H	O	J
Kľúč	1	1	1	1
Šifrovaný text	B	I	P	K

Formálnejšie, Caesarov algoritmus (t. J. šifra) šifruje správy „otáčaním“ každého písmena o pozície  $k$ . Viac formálne, ak  $p$  je nejaký obyčajný text (t.j. nešifrovaná správa),  $p_i$  je  $i$ -tý znak v  $p$ , a  $k$  je tajný kľúč (t.j. nezáporné celé číslo), potom každé písmeno,  $c_i$ , v šifrovom texte,  $c$ , sa vypočíta ako

$$c_i = (p_i + k) \% 26$$

Kde  $\% 26$  tu znamená „zvyšok pri delení 26“. Tento vzorec pravdepodobne spôsobí, že šifra je zložitejšia ako je, ale je to skutočne iba výstižný spôsob presného vyjadrenia algoritmu.

Na účely diskusie uvažujte o A (alebo a) ako 0, B (alebo b) ako 1,..., H (alebo h) ako 7, I (alebo i) ako 8,... a Z (alebo z) ako 25. Predpokladajme, že Caesar chce povedať niekomu dôverne, že tentoraz používa kľúč,  $k$ , z 3. A tak jeho holý text,  $p$ , je „Cau“, v tomto prípade je jeho prvý znak,  $p_0$ , C (teda 2) a druhý znak jeho textu,  $p_1$ , je „a“ (teda 0) - atď. Prvým znakom jeho šifrového textu,  $c_0$ , je teda „F“ (teda  $2 + 3 = 5$ ), a druhým znakom jeho šifrového textu,  $c_1$ , je teda „d“ (teda  $0 + 3 = 3$ ). Vidíš prečo?

Zo stránky predmetu si stiahni *template* programu s názvom *cezar.c*

Template momentálne načíta premennú ***unsigned int kluc***, a pole znakov ***text*** - teda nešifrovaný text, tvojou úlohou je vypísať do konzoly šifrovaný text.

Časti programu sú nedopísané a zakomentované. Tvojou úlohou je dopísať tieto časti programu.

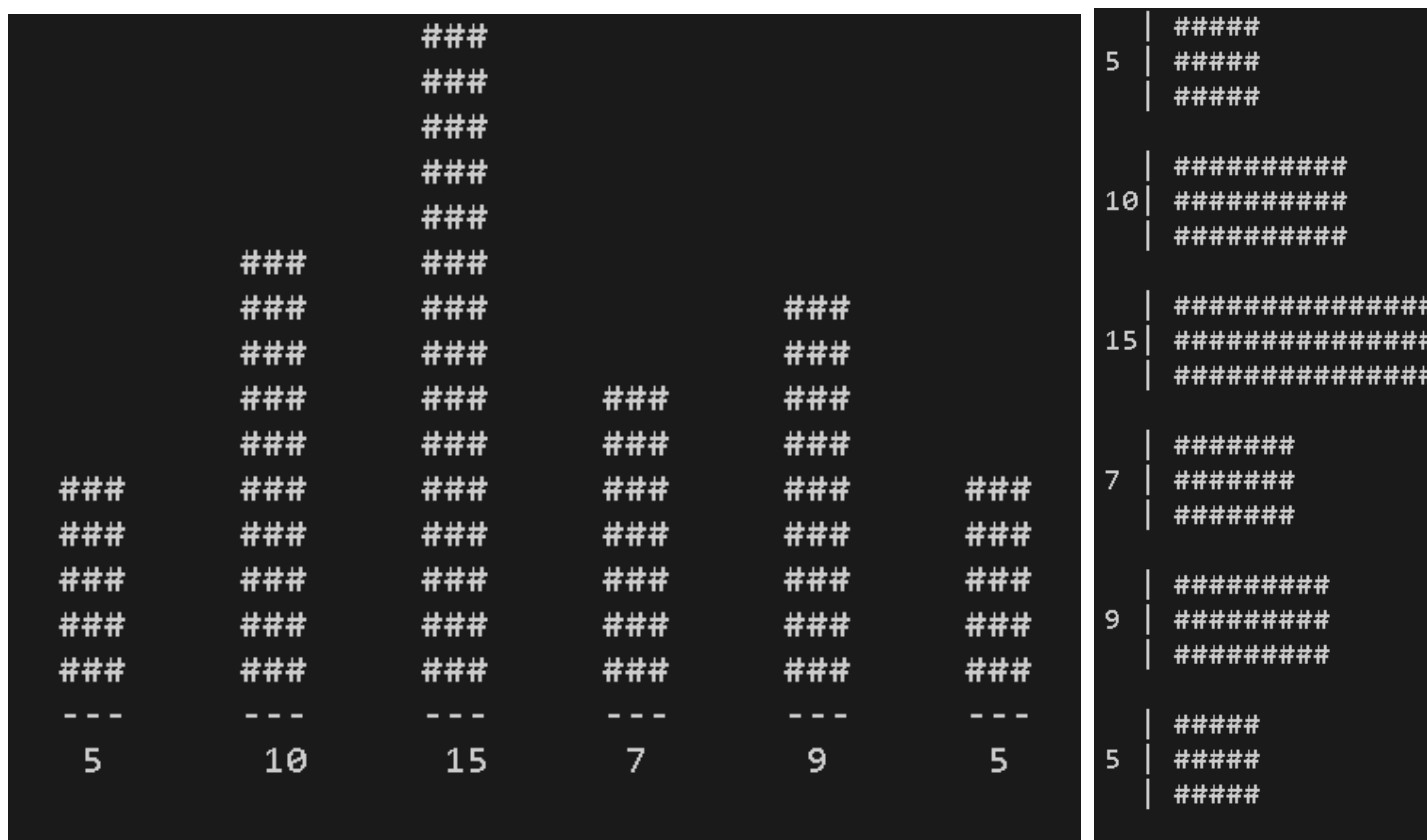
- Vypíš na obrazovku šifrovaný text
- Zašifruj len písmená bez diakritiky
- Nešifruj písmená z diakritikou, čísla a iné znaky
- Program musí rozlišovať malé a veľké písmená t.j. vstup „ahoj“ nie je ten istý ako „Ahoj“ alebo „aHoJ“
- Šifru nemusíš ukladať do novej premennej, stačí vypísať znaky na obrazovku

Predpokladaný výstup programu

Kluc: 1 Nesifrovany text: AbbA Sifrovany text: BccB	Kluc: 12 Nesifrovany text: FOObAr Sifrovany text: RAAnmD
Kluc: 65 Nesifrovany text: Tomas1 Sifrovany text: Gbznf1	Kluc: 128 Nesifrovany text: ahoj ako sa mas Sifrovany text: yfmh yim qy kyq

## Bargraph

Bargraf (po slovensky stĺpcový diagram, graf) je forma zobrazovania kategórií do stĺpcov. Môže byť vykreslený horizontálne alebo vertikálne.



Obrázok 1 Vertikálny bargraph vľavo a horizontálny vpravo ako výstup z programov pri použití vstupného súboru *number.txt*

Tvojou úlohou je implementovať jednu z možností tohto stĺpcového grafu. Chcem prízvukovať, že implementovanie vertikálneho grafu je algoritmicky o trochu ťažšie ako horizontálneho a preto je aj ohodnotení väčším počtom bodov.

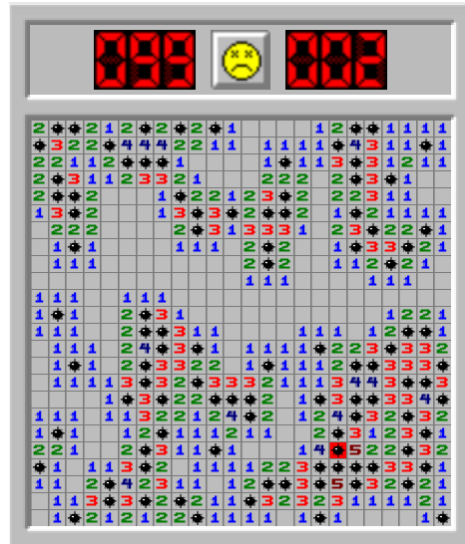
Zo stránky predmetu si stiahni *template* programu s názvom *bargraph.c*. Niektoré časti sú predpísané, niektoré zakomentované. Momentálne program načíta hodnoty zo súboru, ktorý mu zadáš ako vstupný. Tento súbor by mal však obsahovať len **kladné celé čísla**. Po spustení program načíta zvolený súbor do poľa s názvom (*unsigned int*) **arr**. Dĺžka tohto poľa je zaznamenaná v premennej **arr\_length**.

Tvojou úlohou je vytvoriť stĺpcový graf z hodnôt v poli **arr**. Sám sa rozhodni, ktorý z grafov chceš implementovať. Výstup z programu pri použití súboru *number.txt* vidíš vyššie.

### \*\*\*Míny

Tento program je ohodnotený + 10 bodov navyše

Kto by nepoznal klasickú hru míny? V hre si odkrýval políčka a možno si natrafil na mínu, vtedy hra skončila. Keď si ale odkryl políčko bez míny, hra ti ukázala počet mín v okolí tohto políčka. Tvojou úlohou je (zatiaľ) implementovať vygenerovanie hracieho poľa pre hru míny. Toto zahŕňa náhodné uloženie mín v poli a pre každé políčko spočítanie počtu mín v okolí tohto políčka.



Implementuj túto časť hry (zvyšok neskôr) univerzálne t.j. pre akékoľvek rozmery hracej plochy či už je to plocha 5x5, 50x50 či 100x100 ale aj 50x10 alebo 100x500.

1. Na začiatku sa spýtaj užívateľa na rozmery hracej plochy. Podmienka je zadanie hodnoty väčšej ako nula. Ak zadá inú, spýtaj sa ho znova.

Príklad vstupu:

Zadaj dĺžku: 10  
Zadaj výšku: 10

Zadaj dĺžku: 0  
Zadaj dĺžku: 5  
Zadaj výšku: -5  
Zadaj výšku: 10

2. Spýtaj sa užívateľa na počet mín. Samozrejme podmienka počtu mín musí spĺňať nejaké kritériá. Počet mín musí byť väčší ako 0 ale zároveň menší ako veľkosť hracej plochy. Ak je veľkosť hracej plochy 10x10 (t.j. 100 políčok), dovoľme užívateľovi max. 60% mín tzn. V tomto prípade 60 mín. Ak je veľkosť 5x5 (25 políčok) dovoľme užívateľovi max. 15 mín. Ak zadá viac, dovoľme maximálny počet.

Príklad vstupu:

Zadaj dĺžku: 5  
Zadaj výšku: 10  
Zadaj (max. 30) počet mín: 18  
Pocitam s 18 minami.

Zadaj dĺžku: 3  
Zadaj výšku: 2  
Zadaj (max. 3) počet mín: 5  
Pocitam s 3 minami.

3. Vygeneruj dvojrozmerné pole, nazvime ho *grid*, ktoré bude mať *minimálne* rozmery dĺžka x výška t.j. definícia, ak sa rozhodnete pre tieto rozmery je `int grid[vyska][dlzka]`. *Odporúčam toto pole inicializovať na samé 0.*

4. Na náhodné políčka vygeneruj míny.

Možno by si mal ešte raz preštudovať program *rand.c* pre generovanie náhodných čísiel.

Nezabudni, že ak sa na tomto políčku, ktoré si náhodne vygeneroval mína už nachádza, neber toto políčko v úvahu. Na hracej ploche musí byť vždy presne toľko mín, koľko užívateľ zadal. Ak si vygeneroval dve náhodné čísla `int stlpec` a `int riadok`, potom do `grid[riadok][stlpec]` priradi hodnotu -1. Míny teda budú reprezentované v poli číslom -1.

5. Pre všetky ostatné polia spočítaj počet mín v okolí tohto poľa. Všimni,



Ak sa políčko dotýka jednej míny svieti na ňom číslo 1 (prvé tri hodnoty v prvom riadku). Uložíme vtedy do konkrétneho miesta v poli `grid` číslo 1.

Ak sa políčko nedotýka, žiadnej míny, políčko je prázdne. Uložíme vtedy do konkrétneho

miesta v poli `grid` číslo 0.

T.j do konkrétneho miesta v poli `grid` uložíme číslo reprezentujúce počet mín, ktorých sa toto políčko dotýka (vo všetkých smerom).

6. Na obrazovku vypíšete obsah poľa `grid`. Namiesto míny však vypíšete symbol #.

Napiš celý program *miny.c*

Možné výstupy

```
dĺžka: 60
výška: 20
Zadaj (max. 720) počet mín: 300
Pocítam s 300 minami

# 4 # 2 3 # 2 0 # 2 # 2 2 # 2 # 2 # 2 1 1 0 0 0 1 # 4 # 3 # 1 1 # # # 2 1 1 1 0 1 # 4 # 3 1 2 # 4 # 3 # 3 1 1 0 0 0 2 #
# 4 # 4 # 2 0 3 # 5 # 2 1 3 2 4 3 4 # 2 1 0 1 1 2 3 # 5 3 3 3 4 5 # 2 2 # 2 0 2 3 # # # 2 2 # 2 3 # 4 # 3 1 1 0 2 #
4 5 6 # 5 2 1 0 2 # 3 2 2 1 1 # 2 # 5 # 3 2 3 # 3 2 3 # 4 # # 2 1 1 2 # 2 0 1 # 3 4 # 2 1 2 2 1 1 1 4 # 5 # 3 2 4 3
# # # # 3 1 0 1 1 1 2 # 3 2 2 2 3 # 5 # # 3 # # 3 1 2 4 # 5 # 3 1 0 0 1 1 1 0 1 1 1 2 3 3 2 1 1 0 0 1 3 # 6 # 4 # # #
4 6 # 6 # # 1 1 2 1 3 # 4 # 2 1 3 # 3 3 4 # 3 2 # 1 3 # 4 1 2 1 1 0 0 0 0 1 1 1 1 2 # 4 # 3 1 0 2 # 4 # 4 3 4 3
# 4 # 4 2 2 3 # # 1 2 # 4 2 3 # 3 # 2 1 # 2 1 1 1 1 2 # 2 0 2 # 2 1 2 2 1 2 # 3 2 # 5 # 6 # # 2 0 2 # 3 3 4 # 1 1 #
2 3 4 # 3 1 1 # # 3 1 1 2 3 # 2 2 # 3 # 1 1 1 0 0 0 0 1 2 2 1 2 # 3 2 # # 2 3 # # 3 2 3 # # 4 # 2 0 1 1 1 1 # 3 2 2 1
1 2 # 3 # 2 3 4 3 1 0 1 2 # 3 2 1 2 3 3 2 2 1 2 1 1 0 0 0 1 # 2 2 3 # 3 2 3 # 2 2 3 # 1 1 3 3 4 3 4 3 2 2 1 2 1 2 # 1 0
2 # 2 2 1 3 # # 3 1 1 1 # 3 # 1 1 2 # 3 # 3 # 2 # 2 1 1 1 3 3 # 2 3 # 2 0 1 1 1 0 1 2 3 2 2 # 3 # # # # 3 # 1 0 1 1 1 0
# 3 1 1 1 4 # # 3 # 1 2 2 4 2 1 # 3 4 # 3 1 2 2 # 1 1 # 2 # 3 # 2 1 1 0 0 0 1 2 2 2 # # 3 4 # 5 5 # 4 # 3 2 1 0 0 0
# 2 1 2 # 4 # 4 2 1 1 1 # 3 # 3 2 3 # 2 1 1 0 0 1 1 1 1 3 2 3 1 1 0 1 1 2 1 2 # # 3 3 4 # 4 # 4 # 3 4 2 3 # 1 0 0 0
1 1 1 # 2 3 # 3 1 1 3 3 5 # 3 # 3 2 2 0 0 0 1 1 2 1 1 1 2 # 2 1 1 0 1 # 4 # 3 2 3 # 2 3 # 4 3 # 3 # 3 # 3 3 4 2 1 0 0
0 1 2 3 2 2 2 # 1 1 # 3 # # 3 4 4 4 # 2 1 2 1 2 # 2 # 1 1 # 4 5 # 3 1 1 2 # # 3 1 2 2 3 # 3 # 3 3 4 3 # 3 3 # # 1 1 1
0 1 # 2 # 2 3 3 3 2 3 # 3 2 2 # # # 2 3 # 3 # 2 1 2 1 1 2 3 # # # 2 1 2 2 3 # 1 1 # 4 3 3 1 2 # # 2 2 # 2 3 4 3 1 1 #
1 3 4 4 2 2 # # 2 # 3 2 2 1 2 3 3 2 1 2 # 3 1 1 0 1 1 2 2 # 4 # # 3 2 # 1 0 1 2 2 2 2 # # 2 1 2 3 3 1 2 3 3 2 # 2 1 1 1
1 # # # 2 2 3 3 3 3 # 1 # 1 0 0 0 1 2 2 1 1 1 3 # 3 # 2 2 2 1 1 2 2 1 1 2 # 2 3 3 4 # 1 1 # 2 1 2 # # 2 2 # 1 0 0
1 2 3 2 # # 1 2 # 3 # 2 2 2 3 2 1 1 1 2 # 3 3 # 3 # 4 2 3 1 1 0 0 0 1 # 1 1 # 4 # 5 # 3 1 2 2 2 2 # 2 2 2 2 2 2 2
0 0 1 1 2 1 1 2 # 4 2 2 1 # 3 # # 1 1 # 2 2 # # 4 5 3 3 # 2 # 1 1 1 1 2 2 2 1 1 3 # # # 3 2 3 # 1 1 2 2 1 1 2 # 1 1 #
2 2 2 # 1 0 0 1 1 3 # 2 2 2 4 # 3 2 2 2 1 2 3 4 # # # 3 2 3 2 3 2 # 2 2 # 1 0 1 2 3 4 4 5 # # 2 1 0 1 # 1 2 # 3 2 2 3 2
# # 2 1 1 0 0 0 2 # 2 1 # 2 1 1 1 # 1 0 1 # 2 2 3 3 # 1 1 # 2 # 2 2 # 2 1 0 1 # 1 1 # # # 3 1 0 0 1 1 1 2 # 2 1 # 1 0
```

```
dĺžka: 10
výška: 10
Zadaj (max. 60) počet mín: 5
Pocítam s 5 minami

0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 0 0 0 1 1
# 1 1 # 1 0 0 0 1 #
1 1 1 1 1 0 0 0 1 1
0 0 0 0 1 1 1 0 0 0
1 1 1 0 0 1 # 1 0 0
1 # 1 0 0 1 1 1 0 0
1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

```
dĺžka: 10
výška: 1
Zadaj (max. 6) počet mín: 10
Pocítam s 6 minami

# # # # 1 1 # 2 # 1
```

```
dĺžka: 5
výška: 5
Zadaj (max. 15) počet mín: 15
Pocítam s 15 minami

# 1 2 # 3
3 4 4 # #
# # # 6 #
# # # 3
# # 5 # 2
```

