

# Programovanie (1) v C/C++ 2024/25

## Tréningový príklad na skúšku 1c

### Preteky

Toto je cvičný príklad pre prvý príklad na skúške. Odporúčame prečítať si zadanie až keď budete úlohu skutočne cvične riešiť. Predtým ako začnete programovať, si poriadne rozmyslite, aké dátové štruktúry (polia, matice, struct-y a pod.) chcete v programe použiť.

Vašou úlohou je spracovať údaje z pretekov. Zúčastnilo sa ich  $P$  pretekárov, ktorí sú očíslovaní číslami  $0, \dots, P - 1$ . Preteky prebiehajú nasledovne: V teréne je rozmiestnených  $N$  stanovišť, ktoré sú očíslované číslami  $0, \dots, N - 1$ . Pretekári ich hľadajú, pričom vždy keď stanovište nájdú, pošlú správu v tvare **kto kam**, ktorá oznamuje, že účastník s číslom **kto** objavil stanovište s číslom **kam**. Pretekári môžu stanovištia prechádzať v ľubovoľnom poradí. Môže sa stať, že pretekár omylom pošle správu o objavení určitého stanovišťa aj viackrát, pričom druhé a ďalšie objavenie stanovišťa tým istým pretekárom nemá vplyv na výsledky pretekov.

Cieľom je vypísať výsledky pretekov, ktoré zahŕňajú poradie úspešných pretekárov a zoznam neúspešných pretekárov. Účastníci, ktorí objavia všetky stanovištia sú úspešní, ostatní sú neúspešní. Poradie úspešných pretekárov sa určuje podľa toho, kedy objavili svoje posledné stanovište. Neúspešní účastníci nemajú určené poradie, ich zoznam vypíšete v poradí podľa čísla pretekárov.

Na prvom riadku vstupu sú celé čísla  $P$  a  $N$ . Potom nasleduje niekoľko riadkov správ v poradí, ako boli poslané účastníkmi. Každá správa pozostáva z dvojice čísel **kto kam** ktorá oznamuje, že účastník s číslom *kto* objavil stanovište s číslom *kam*. Môžete predpokladať, že každá správa je korektná a teda  $kto \in \{0, \dots, P - 1\}$  a  $kam \in \{0, \dots, N - 1\}$ .

Program vypíše na konzolu výsledky pretekov, presnejšie poradie úspešných pretekárov a zoznam neúspešných pretekárov. Presne dodržujte formát z príkladu nižšie. V prípade, že žiadni účastníci neboli neúspešní, by mal po dvojbodke v poslednom riadku hneď nasledovať znak nového riadku a za ním koniec výstupu.

**Obmedzenia:** V programe je zakázané používať polia konštantných veľkostí. Všetky polia **alokujte dynamicky** (príkazom **new**) alebo použite štruktúry, ktoré menia veľkosť podľa potreby (**vector**, **string** a podobne). Dynamicky alokovanú pamäť **odalokujte**.

#### Príklad vstupu:

```
4 5
0 0
0 1
0 0
2 1
0 2
0 4
0 3
2 0
2 2
1 2
2 1
2 4
2 3
```

#### Príklad výstupu:

```
1. skoncil 0
2. skoncil 2
Neuspesni ucastnici: 1 3
```

Upozornenie: čísla zo vstupu je treba čítať až do konca vstupu. Pri zadávaní vstupu ručne ho ukončíte v Linuxe stlačením Ctrl-D na začiatku riadku, vo Windows Ctrl-Z a Enter.