1 Naloge

Ta vaja od vas zahteva, da implementirate urejanje tabel v različnih situacijah in kontekstih. Implementirate lahko katerikoli algoritem za urejanje, razen, če naloga izrecno zahteva mehurčno.

- Implementirajte funkcijo bubbleSort(int tab[], int n, bool asc), ki prejme 1-dimenzionalno tabelo celih števil, dolžino tabele in parameter asc. Ta naj uredi vrednosti v naraščajočem vrstnem redu, če je parameter $asc = true$ in v padajočem, če je $asc = false$.
- Prejšnjo funkcijo popravite tako, da bo funkcija bubbleSort prejela tabelo znakov namesto celih števil.
- Implementirajte funkcijo bubbleSort2D1R(int m, int n, int tab[][n]), ki uredi prvo vrstico dvo-dimenzionalne tabele v naraščajočem vrstnem redu. Funkcijo pokličite kot bubbleSort2D1R(m, n, tab), kjer je m število vrstic, n število stolpcev in tab 2-dimenzionalna tabela.
- Implementirajte funkcijo bubbleSort2D1C(int m, int n, int tab[][n]), ki uredi prvi stolpec dvo-dimenzionalne tabele v padajočem vrstnem redu. Funkcijo pokličite kot bubbleSort2D1C(m, n, tab), kjer je m število vrstic, n število stolpcev in tab 2-dimenzionalna tabela.
- Implementirajte funkcijo bubbleSort2D1D(int m, int n, int tab[][n]), ki uredi levo diagonalo dvo-dimenzionalne tabele v padajočem vrstnem redu. Funkcijo pokličite kot bubbleSort2D1D(m, n, tab), kjer je m število vrstic, n število stolpcev in tab 2-dimenzionalna tabela.
- Implementirajte funkcijo bubbleSort2D(int m, int n, int tab[][n], bool asc), ki uredi stolpec dvo-dimenzionalne tabele v naraščajočem vrstnem redu, če $asc = true$ oz. padajočem, če $asc = false$. Funkcijo pokličite kot bubbleSort2D(m, n, tab, asc), kjer je m število vrstic, n število stolpcev, tab 2-dimenzionalna tabela in asc parameter, ki odreja, če je tabela padajoča ali naraščajoča. Ureja naj se najprej po stolpcih, nato po vrsticah, kjer je levo-zgornji element najmanjši.

2 Zahtevnejše naloge

- Implementirajte algoritem selection sort (`https://en.wikipedia.org/wiki/Selection_sort`
- Implementirajte algoritem insertion sort (`https://en.wikipedia.org/wiki/Insertion_sort`)
- Implementirajte bipartitni bubble sort. Iz večje tabele najprej ustvarite dve konceptualno manjši tabeli. Uredite najprej levi del in nato še desni del. Ta dela na koncu (pametno) združite v večjo tabelo. Ali lahko postopek ponovite s štirimi deli? Osmimi? Šestnajstimi? Ali mislite, da je takšen postopek hitrejši, kot navaden bubble sort?

2.1 Dodatne naloge

- `https://projecteuler.net/archives`
- `https://www.w3resource.com/c-programming-exercises/for-loop/index.php`
- `https://codeforwin.org/2015/06/for-do-while-loop-programming-exercises.html`