



## Programiranje II

### LV 5

1. Napisati generičku funkciju *Obrni*(*Tip \*pocetak, Tip \*iza\_kraja*) koja prima dva pokazivača, a koja obrne dio niza (ili vektora) između tih pokazivača. Podrazumijevajte da *pocetak* pokazuje na element koji je prije elementa na koji pokazuje pokazivač *iza\_kraja*.
2. Napisati funkciju *void SmjestiMinMax* koja prima 4 parametra, pokazivač na početak niza, pokazivač iza kraja niza, te dva pokazivača u koja treba smjestiti pokazivač na najveći i najmanji element niza (razmislite kako posljednja dva parametra treba proslijediti funkciji). Na primjer, ako je dat niz {6,2,4,8,6,1,3}, nakon poziva *SmjestiMinMax(niz, niz+7, Min, Max)*, pokazivači *Min* i *Max* trebaju pokazivati na šesti i četvrti element niza, redom. Na kraju, promijenite implementaciju ove funkcije tako da ona bude generička.
3. Napisati funkciju *void Sortiraj(vector<vector<int>> &vektori)* koja sortira vektor vektora. Jedan vektor smatramo manjim od drugog ako mu je prva koordinata manja (npr. [5,10,3] < [6,1]), a ako im je prva koordinata ista, onda poredimo druge koordinate (npr. [5,10,3] < [5,11,1]), ako su one iste, poredimo treće koordinate, i tako dalje, pri čemu ako pri tom poređenju neki vektor ostane bez elemenata, on je manji od drugog (npr. [4,1,2] < [4,1,2,0]). Sortiranje se vrši na sljedeći način: najprije se porede prva dva vektora, te ako nisu u dobrom poretku (tj. ako je prvi veći od drugog) zamijenimo ih, zatim poredimo drugi i treći, te ih zamijenimo ako je potrebno, pa treći i četvrti, i tako dalje, sve dok ne uporedimo posljednja dva elementa (i zamijenimo ih po potrebi). Nakon jednog ovakvog prolaska nam je sigurno najveći element na kraju. Zatim pravimo drugi ovakav prolazak, ali je za njega dovoljno ići samo do predzadnjeg elementa (jer je zadnji već na svom mjestu). Ako imamo  $n$  vektora, dovoljno je napraviti  $n - 1$  ovakvih prolazaka. Ovakav način sortiranja se zove „bubble sort“ (ako vam nije jasan prethodni opis algoritma, možete izgooglati kako radi bubble sort). Kada budete vršili zamjenu vektora, nemojte koristiti funkciju *swap*, već ih „ručno“ zamijenite, koristeći tri dodjele (*pomocni = a; a = b; b = pomocni;*).  
Ovu funkciju testirati u main funkciji sa velikim brojem vektora velike dužine (npr. 1000 vektora koji imaju po 1000 elemenata). Zatim implementirati istu funkciju ali tako što ćemo umjesto vektora vektora koristiti vektor pokazivača na vektore, te uporediti vrijeme izvršavanja ovih funkcija.