

Vježba 4

Cilj vježbe je upoznavanje sa strukturom podataka red, pri čemu će studenti za pripremu uraditi implementaciju red strukture podataka, a zatim na tutorijalu primijeniti implementaciju za rješavanje nekoliko problema

Zadatak 1.

Napisati funkciju koja prima red i modifikuje ga tako da budu izbačene sve vrijednosti veće od srednje vrijednosti njegovih elemenata. Ustanoviti koja je vremenska kompleksnost rješenja.

Napomena: Zadatak riješiti koristeći samo ulazni red, nije dozvoljeno korištenje pomoćnih kolekcija.

Zadatak 2.

Napisati funkciju koja prima stek, red i varijablu `pocetnaDuzinaReda` tipa `int` koja će elemente iz reda ubaciti na dno steka u obrnutom redoslijedu. Primjer: Neka je dat red sa elementima {1, 2, 3, 4} gdje je element 1 na čelu reda i stek sa elementima {5, 6, 7, 8} gdje je element 8 na vrhu steka. Rezultantni stek se sastoji od elemenata {1, 2, 3, 4, 5, 6, 7, 8} gdje je element 8 na vrhu steka. Ustanoviti koja je vremenska kompleksnost rješenja.

Napomena: Zadatak riješiti koristeći samo ulazne parametre, nije dozvoljeno korištenje pomoćnih kolekcija.

Zadatak 3.

Napisati funkciju koja prima red, listu i varijablu `n` tipa `int` i modifikuje red tako da elementi iz liste i reda budu naizmjenično poredani gdje je svaki isprepleteni dio reda i liste veličine `n` a počinje se sa elementima reda. U slučaju razlike u dužini reda i liste, višak elemenata se postavlja na kraj reda. Primjer 1: Neka je dat red sa elementima {1, 2, 3, 4} gdje je element 1 na čelu reda i lista sa elementima {1, 2, 3, 4} i neka je `n=1`. Rezultantni red je {1, 1, 2, 2, 3, 3, 4, 4}. Primjer 2: Neka je dat red sa elementima {A, A, A, A, A, A, A} gdje je element A na čelu reda i lista sa elementima {B, B, B, B} i neka je `n=2`. Rezultantni red je {A, A, B, B, A, A, B, B, A, A, A}.

Napomena: Zadatak riješiti koristeći samo ulazne parametre, nije dozvoljeno korištenje pomoćnih kolekcija. Također, nije dozvoljena ni modifikacija liste.

Zadatak 4.

Razviti klasu `StampacServer` koja obavlja funkciju servera za štampanje u jednoj kompaniji. Klasa implementira funkciju `void posaljiDokument(Prioritet p, string sadrzaj)`. `Prioritet` je tip `enum` {`Sef`, `Radnik`, `Pripravnik`}. Ovaj `enum` se treba nalaziti u javnom interfejsu klase. Sadržaj dokumenta za printanje je pojednostavljeno samo tekst - pa je tipa `string`. Ova funkcija treba da prihvati dokument sa odgovarajućim prioritetom i pohrani ga za kasnije

štampanje.

Osim toga, ova klasa treba da implementira funkciju `string dajDokument()` koja od dokumenata koji su poslani za štampanje vraća neodštampani dokument sa najvišim prioritetom, a ako ima više dokumenata sa istim prioritetom, onda najstariji. Logično, najviši prioritet je `Sef`, a zatim `Radnik` i najniži `Pripravnik`. Ako ne postoji niti jedan neodštampani dokument, treba vratiti **prazan string**.

Klasa bi trebala biti korištena tako što će driver za štampače na računarima uposlenika pozivati funkciju `posaljiDokument`, a štampači čim završe sa prethodnim dokumentom, pozivati funkciju `dajDokument`. Ovakvo ponašanje treba simulirati u main programu.