

5. Toteuta yksinkertainen jonotusjärjestelmä ohjelmointikurssin harjoituksia varten, jota käsitellään kahdella funktiolla seuraavasti:

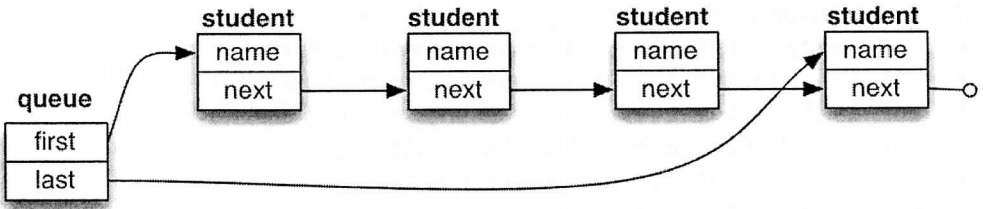
a) void enqueue(struct queue \*q, const char \*name) , joka lisää jonon 'q' loppuun opiskelijan nimeltä 'name'.

b) struct student \*deque(struct queue \*q) , joka poistaa jonon 'q' ensimmäisen alkion, ja palauttaa sen paluuarvonaan.

Edellä esiteltyt funktiot käyttävät seuraavanlaisia tietorakenteita:

```
struct student {  
    char name[20];  
    struct student *next;  
};  
  
struct queue {  
    struct student *first;  
    struct student *last;  
};
```

Jono on siis toteutettu linkitettyinä listana, jossa kukin alkio on '**struct student**' -tyyppiä. '**struct queue**' ylläpitää osoitinta listan ensimmäiseen alkioon (**first**), ja listan viimeiseen alkioon (**last**). Lähtötilanteessa, kun lista on tyhjä, sekä first että last voidaan olettaa olevan NULL. Kun listassa on vain yksi alkio, molempien kenttien tulisi osoittaa samaan alkioon.



Funktio '**enqueue**' varaa tarvittavan muistin uudelle alkion (**struct student**) perustuen annettuun nimeen (**name**) ja linkittää sen **struct queue** -listan viimeiseksi alkion. Huomioi, että name-kentän pituus on rajattu. Kannattaa muistaa myös next-osoittimien päivittäminen.

Funktio '**deque**' palauttaa jonon ensimmäisen alkion ja poistaa sen linkitetystä listasta. Tämän johdosta '**first**'-osoitin muuttuu. deque-funktion ei tarvitse vapauttaa muistia, vaan oletetaan että funktion kutsuja vapauttaa muistin aikanaan.

Voit olettaa että muistin varaus onnistuu aina.

Tehtävänä on siis toteuttaa kaksi edellä mainittua funktiota.

Onnistunut enqueue-funktion toteuttaminen tuottaa maksimissaan 3 pistettä.

Onnistunut deque-funktion toteuttaminen tuottaa maksimissaan 3 pistettä.