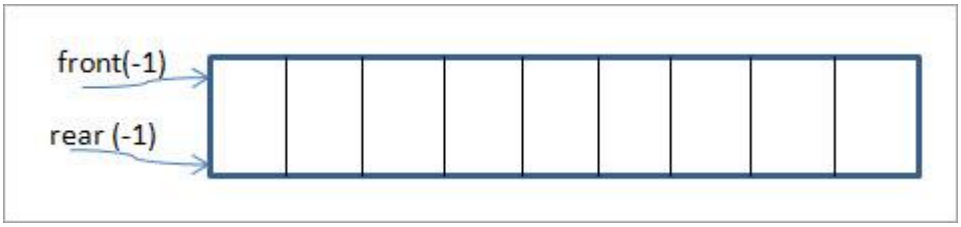
Ред (queue) и основни операции кај ред – час 3

Ред (queue) е линеарна листа во која елементите се ставаат на крајот, а се земаат од почетокот на листата. Работат на принцип „кој прв влезе, прв излегува“ (First In First Out – FIFO). Редот има два покажувачи: едниот покажува на почеток (рос), а другиот на крај на редот (kraj). Редот може да биде со фиксна големина или со неограничена ако е имплементиран како поврзана листа.



Кога редот е празен, двата покажувачи имаат вредност -1. Покажувачот (kraj) покажува на местото од каде се додаваат елементите во редот. Операцијата на додавање на елементи се нарекува “enqueue”.

Покажувачот рос покажува на местото од каде елементите се вадат (бришат) од редот. Операцијата на вадење (бришење) на елементи се нарекува “dequeue”.

Кога покажувачот kraj има вредност -1, тогаш редот е полн. Ако вредноста на покажувачот рос е 0, редот е празен.

Операции кои се извршуваат со ред:

- Иницијализација на празен ред
- Проверка дали редот е полн
- Ставање податок на крајот на редот
- Проверка дали редот е празен
- Земање податок од почетокот на редот

EnQueue: Додава елемент во редот. Додадениот елемент секогаш е на крајот на редот.
DeQueue: Бриши елемент во редот. Елементот секогаш се бриши од почетокот на редот.
isEmpty: проверува дали редот е празен.
isFull: проверува дали редот е полн.
peek: зема елемент од почетокот на редот но не го бриши.

Enqueue

При извршувањето на оваа операција:

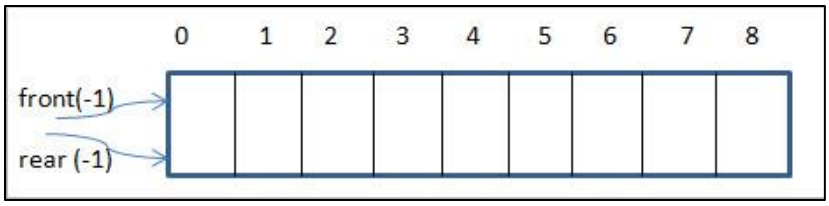
- Се проверува дали редот е полн
- Ако е полн дава overflow грешка и завршува програмата
- Инаку зголемува “kraj” (покажувачот покажува на следната позиција).
- Додава елемент на позицијата на која покажува “kraj”

Dequeue

При извршувањето на оваа операција:

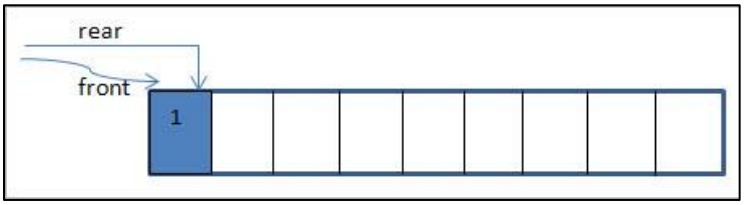
- Се проверува дали редот е празен
- Ако е покажува underflow и завршува програмата

- Инаку пристапува до елементот на кој покажува “рос”
- Го поместува покажувачот на да покажува на следниот податок



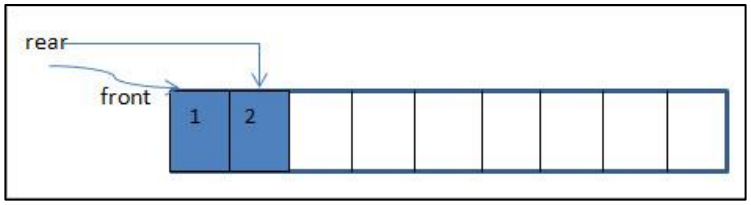
празен ред

Се додава елемент со вредност 1 и покажувачот “kraj” се поместува на следната позиција.



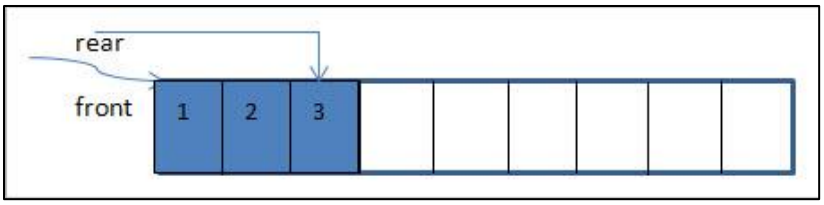
Enqueue операција

Се додава елемент со вредност 2 и покажувачот “kraj” се поместува повторно на следната позиција.



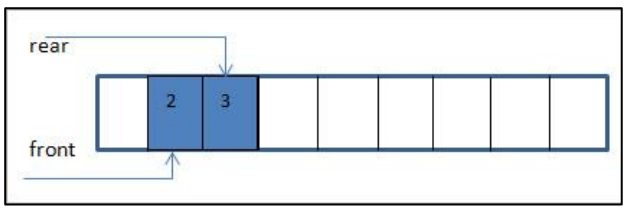
Enqueue операција

Се додава елемент со вредност 3 и покажувачот “kraj” се поместува повторно на следната позиција



Enqueue операција

Во овој момент покажувачот “kraj” покажува на позиција 2, покажувачот “рос” на 0. Сега, го бришеме елементот на кој покажува покажувачот “рос”. Бидејќи окажувачот “рос” е на позиција 0, елементот што се брише е 1.



Dequeue операција

Првиот елемент од редот т.е. 1 е прв што се бриши од редот. Сега покажувачот “рос” ќе се помести на следната позиција, т.е. покажува на позиција 1.

Основни функции на queue контејнерот

push() – додава елемент во редот. Елементот се додава на крајот од редот.

pop() – бриши елемент од почетокот на редот и големината на редот се намалува за 1. Елементот што се бриши е првиот што е внесен. Функцијата не враќа ништо како резултат.

front() – го враќа првиот елемент од редот

back() - го враќа последниот елемент од редот

Двете функции го враќаат елементот од соодветната позиција, а не го бришат.

size() – го враќа бројот на вмесени елементи во редот

empty() – проверува дали редот е празен или не е. Враќа вредност true ако редот е празен инаку враќа вредност false

swap() – ги заменува елементите на два реда

Пр. Програма со која се внесуваат n елементи во ред и се печатат елементите.

```
#include <iostream>
#include <queue>

using namespace std;

int main()
{
   queue <int> q;                    //иницијализира ред со име q чии елементи се цели броеви
   int n,i,x;
   cout << "vnesi broj na elementi" << endl;
   cin>>n;
   for(i=1;i<=n;i++)
   {
      cin>>x;                        //се внесува елементот
      q.push(x);                     //елементот се додава на крајот од редот
   }
   while(!q.empty())                 //се додека во редот има елементи
   {
      cout<<q.front()<<" ";          //се печатат елементите од почетокот на редот
      q.pop();                       //се преминува на следна позиција во редот
   }
   return 0;
```

```
}
```

Пр. Програма со која се внесуваат n елементи во ред, се вади првиот елемент и се печатат останатите елементи на редот

```
 #include <iostream>
#include <queue>

using namespace std;

int main()
{
   queue <int> q;
   int n,i,x;
   cout << "vnesi broj na elementi" << endl;
   cin>>n;
   for(i=1;i<=n;i++)
   {
      cin>>x;
      q.push(x);
   }
   q.pop();                          //се преминува на следна позиција, т.е. елементите се печатат од вториот елемент во редот
   while(!q.empty())
   {
      cout<<q.front()<<" ";
      q.pop();
   }
   return 0;
}
```

Пр. Програма со која се печати големината, првиот и последниот елемент од редот

```
 #include <iostream>
#include <queue>

using namespace std;

int main()
{
   queue <int> q;
   int n,i,x;
   cout << "vnesi broj na elementi" << endl;
   cin>>n;
   for(i=1;i<=n;i++)
   {
```

```
        cin>>x;
        q.push(x);
    }
    cout<<"goleminata na redot e "<<q.size()<<endl;
    cout<<"poceten element e "<<q.front()<<endl;
    cout<<"posleden element e "<<q.back()<<endl;
    return 0;
}
```