

Решавање на проблеми со користење на ред – час 1

Пр. програма со која се креира ред и се печати разликата меѓу првиот и последниот елемент

```
#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++)
    {
        cout<<"vnesi element "<<endl;
        cin>>x;
        q.push(x);
    }

    if (q.front() > q.back()) {
        cout << "razlikata e "<<q.front() - q.back();
    }
    else if (q.front() < q.back()) {
        cout << "razlikata e "<<q.back() - q.front();
    }
    else
        cout << "razlikata e 0";
    return 0;
}
```

Пр. Програма за креирање ред, додавање на елементи, бришење, печатење прв и последен елемент на редот со користење на покажувачи

```
#include <iostream>
#include <queue>
using namespace std;

// kreira klasa Queue
class Queue
{
public:
    int poc, kraj, golemina;
```

```

int dol;
int* array;
};

// funkcija so koja se kreira red so dadena dolzina.
// goleminata na redot se inicijalizira na 0
Queue* kreirajQueue(int dol)
{
Queue* queue = new Queue();
queue->dol = dol;
queue->poc = queue->golemina = 0;
queue->kraj = dol - 1; // vazno e
queue->array = new int[(queue->dol * sizeof(int))];
return queue;
}

// redot e poln koga goleminata ke bide ednakva na dol
int epolna(Queue* queue)
{
return (queue->golemina == queue->dol);
}

// redot e prazen ako goleminata e 0
bool eprazna(Queue* queue)
{
return (queue->golemina == 0);
}

// funkcija koja dodava elementi vo redot
// go smenuva krajot na redot
void enqueue(Queue* queue, int element)
{
if (epolna(queue))
return;
queue->kraj = (queue->kraj + 1) % queue->dol;
queue->array[queue->kraj] = element;
queue->golemina = queue->golemina + 1;
cout << element << " dodaden na redot " << endl;
}

// funkcija koja brisi element od redot
// gi menuva pocetokot i goleminata
int dequeue(Queue* queue)
{
if (eprazna(queue))
return 0;
}

```

```

int element = queue->array[queue->poc];
queue->poc = (queue->poc + 1) % queue->dol;
queue->golemina = queue->golemina - 1;
return element;
}

```

// funkcija koja go dava pocetniot element od redot

```

int pocetok(Queue* queue)
{
if (eprazna(queue))
return 0;
return queue->array[queue->poc];
}

```

// funkcija koja go dava krajniot element od redot

```

int kraj(Queue* queue)
{
if (eprazna(queue))
return 0;
return queue->array[queue->kraj];
}

```

//glavna programa

```

int main()
{
Queue* queue = kreirajQueue(1000);
int i,n,x;
cout<<"vnesi broj na elementi "<<endl;
cin>>n;
for(i=0;i<n;i++)
{
cout<<"vnesi element "<<endl;
cin>>x;
enqueue(queue, x);
}
cout<<" izbrisan element e "<<dequeue(queue)<<endl;
cout<<endl;
cout << "poceten element e " << pocetok(queue) << endl;
cout<<endl;
cout << "Kraen element e " << kraj(queue) << endl;
return 0;
}

```

Задачи за домашна работа:

1. Пр. програма со која се креира ред од n елементи цели броеви и се печати колку елементи се негативни броеви

2. Пр. програма со која се креира ред од n елементи цели броеви и се печати збир на елементите кои се делат со 4.
3. Пр. програма со која се креира ред од n елементи - знаци и да се изброи и испечати колку од внесените знаци се големи, а колку мали букви.
4. Пр. програма со која се креира ред од n елементи реални броеви и да се испечати најмалиот од нив

Прашања поврзани со наставните единици може да се испраќаат на email:

anetastojceska@gmail.com

Решенијата на задачите за домашна работа да се испратат најдоцна до 22.04.2020 година

на email: anetastojceska@gmail.com