

Пример за лабораториска вежба 3 – Магадини и редови

Име и презиме	Број на индекс	Група

Задача 1 (во C):

Да се напише функција која како аргументи прима магацин со веќе внесени елементи и еден цел број. Во магацинот да се најде **последното** појавување на елементот кој има вредност еднаква со вториот аргумент. Ако постои елемент со таква вредност, истиот да се отстрани од магацинот и да се врати позицијата на која е најден (броејќи од горе надолу). Доколку не се најде елемент со таква вредност да се врати нула.

Дадена е главната програма (во C) во која е наполнет еден магацин со произволни вредности и повикана е функцијата.

Забелешка: Како дополнителна структура може да се користи само еден магацин. Да се работи со операциите за додавање и вадење на елемент од магацин.

Пример:

<p>Пред:</p> <p>Магацин: 10 </p> <p> 2 </p> <p> 10 </p> <p> 7 </p> <p> 10 </p> <p> 10 </p> <p> 9 број=10</p>	<p>После:</p> <p>Магацин: 10 </p> <p> 2 </p> <p> 10 </p> <p> 7 </p> <p> 10 </p> <p> 9 </p> <p>Позиција=6</p>
--	---

```
int main()
{
    int broj;
    int i;
    stack magacin;
    int pozicija;

    for (i = 0; i < 10; i++)
    {
        scanf("%d", &broj);
        push(&magacin, broj);
    }

    scanf("%d", &broj);
    pozicija = najdiPozicija(&magacin, broj);
}
```

Задача 2 (во C++):

Студентите полагаат два предмети, ПиА2 и СсП. За секој студент се води евиденција за името и предметот кој го полага. Сите студенти се веќе поставени во еден заеднички ред според времето на пријавување на анкетата на портал. Ваша задача е да креирате две посебни групи на студенти, по една за секој предмет. Студентите ги групирате според следниов критериум: од заедничкиот ред во еден момент треба да се вади еден студент кој се сместува во еден од двата реда според предметот. Ако после него има **последователни студенти** пријавени за истиот предмет, тогаш нив ги вадите и ги додавате на крај на заедничкиот ред, се дури не стигнете до студент кој полага различен предмет. Овој процес продолжува се дури не ги групирате сите студенти во двете посебни групи.

Забелешка: Задачата не работи точно за одредени поставувања на студенти во заедничкиот ред, но работи со примерот подолу. Под примерот се дадени и потребните структури во програмата и функциите за работа со редови во C++.

Пример: Заеднички ред: |("st1","pia2"), ("st2","pia2"), ("st3","ssp"), ("st4","ssp"), ("st5", "ssp"), ("st6", "pia2"), ("st7", "pia2")|

Излезен ред ПИА2: |("st1", "pia2"), ("st6", "pia2"), ("st7", "pia2"), ("st2", "pia2")|

Излезен ред ССП: |("st3","ssp"), ("st4", "ssp"), ("st5","ssp")|

```
struct Student
{
    string ime, predmet;
};

struct Queue
{
    Student array[MAX];
    int front, rear;
    void init();
    bool isEmpty();
    bool isFull();
    void enqueue(Student e);
    Student dequeue();
    Student peek();
};

void Queue::init()
{
    front = 0;
    rear = -1;
}

bool Queue::isFull()
{
    return (rear == MAX - 1);
}

bool Queue::isEmpty()
{
    return (rear == -1);
}

void Queue::enqueue(Student e)
{
    if (isFull())
    {
        cout << "Nema mesto" << endl;
        exit(-1);
    }
    array[++rear] = e;
}

Student Queue::dequeue()
{
    if (isEmpty())
    {
        cout << "Redot e prazen" << endl;
        exit(-1);
    }
    Student pom = array[front];
    for (int i = front; i < rear; i++)
```

```
        array[i] = array[i + 1];
    rear--;
    return pom;
}

Student Queue::peek()
{
    if (isEmpty())
    {
        cout << "Redot e prazen" << endl;
        exit(-1);
    }
    return array[front];
}
```