**მონაცემთა სტრუქტურები**

**რიგი-QUEUE: FIFO (First in First out)**

*ცარიელი რიგის ინიციალიზაცია*

**QUEUE\_INIT**

{

Head=0;

Tail=0;

}

*ელემენტის ჩამატება რიგში*

**ENQUEUE (x)**

{

Q[Head]=x;

Head++;

}

*ელემენტის წაშლა რიგიდან*

**DEQUEUE**

{

Tail++;

Return Q[Tail-1];

}

*შემოწმება, ცარიელია რიგი თუ არა*

**QUEUE\_EMPTY**

{

If (Tail<Head)

Return false;

Else

Return true;

}

**სტეკი-STACK: FILO (First in Last out)**

*ახალი სტეკის ინიციალიზაცია*

**STACK\_INIT**

{

Top=-1;

}

*სტეკში ახალი ელემენტის ჩამატება*

**PUSH (x)**

{

Top++;

S[Top]=x;

}

*ლემენტის წაშლა სტეკიდან*

**POP**

{

Top--;

Return S[Top+1];

}

*შემოწმება, ცარიელია სტეკი თუ არა*

**STACK\_EMPTY**

{

If (Top>=0)

Return false;

Else

Return true;

}

**ძებნის ორობითი ხეები**

**ძებნის ორობითი ხის თვისება:**

თუ *y* წვერო ეკუთვნის *x* წვეროს მარცხენა ქვე-ხეს, მაშინ **key[y]<=key[x]** და თუ *y* წვერო ეკუთვნის *x* წვეროს მარჯვენა ქვე-ხეს, მაშინ **key[y]>=key[x].**

*გასაღების ძებნა ძებნის ორობით ხეში (რეკურსიული ალგორითმი)*

**TREE-SEARCH (x, k)**

{

If (x==nil || k==key[x]) return x;

If(k<key[x]) return TREE-SEARCH (left[x], k);

Else

return TREE-SEARCH (right[x], k);

}

*გასაღების ძებნა ძებნის ორობით ხეში (იტერაციული ალგორითმი)*

**ITERATIVE-TREE-SEARCH (x, k)**

{

while (x!=nil || k!=key[x])

If(k<key[x])

x=left[x];

Else

x=right[x];

Return x;

}

*მინიმალური გასაღების ძებნა ძებნის ორობით ხეში*

**TREE\_MINIMUM(x)**

{

While (left[x]!=nil)

x=left[x];

Return x;

}

*მაქსიმალური გასაღების ძებნა ძებნის ორობით ხეში*

**TREE\_MAXIMUM(x)**

{

While (right[x]!=nil)

x=right[x];

Return x;

}

1.

#include <iostream>

using namespace std;

int main()

{

int n;

cout << "sheiyvanet masivis sigrdze: ";

cin >> n;

int mas[n];

for (int i = 0; i < n; i++) {

cin >> mas[i];

}

int x;

cout << "sheiyvanet romel ricxvs edzebt: ";

cin >> x;

mas[n] = x;

bool s = true;

int k = 0;

while (s && k < n) {

if (mas[k] == x) {

s = false;

k++;

}

k++;

}

if (k == n) {

cout << "ar aris masivshi es elementi" << endl;

} else {

cout << "masivshi aris elementi indeqsad: " << k - 2 << endl;

}

cout<<mas[1];

return 0;

}

2.

#include <iostream>

using namespace std;

int main() {

int n;

cout << "masivis sigrdze ";

cin >> n;

int mas[n];

cout << "sheiyvanet masivis elementebi";

for (int i = 0; i < n; i++) {

cin >> mas[i];

}

int k;

int mx;

for (int i = 0; i < n - 1; i++) {

k = i;

mx = mas[i];

for (int j = i + 1; j < n; j++) {

if (mas[j] < mx) {

mx = mas[j];

k = j;

}

}

mas[k] = mas[i];

mas[i] = mx;

}

for (int i = 0; i < n; i++) {

cout << mas[i] << " ";

}

return 0;

}

5.

int a=580;

int b=105;

int usg;

while(a!=0 && b!=0 ){

if(a>b){

a=a%b;// aq gayofisas dawers 580 nashts 55 a gaxdeba 55

//45 25ze gaoyofs nashttit a=15

} //15 10ze miigebs 5

}

else{

b=b%a ;// exla 105 nashtit gayofs 45 miigebs b= 25

//shemdeg 25 nashtit gayofs 15 miigebs 10

} daukve rodesac 15 s gayofs 5 gaxdeba 0 da damtavrdeba

usg=a+b;

return 0;

7.

QUEUE-INIT{

Head=0;

tail=0;

}

ENQUEUE(x) {

Q[tail]=x;

tail++;}