

[Irányítópult](#) / [Kurzusaim](#) / [Programozás \(GKNB INTM021\)](#) / [Vizsgák](#) / [2021.január 15. 13:00](#)

1 kérdés

Hibás

1 pont szerezhető

Valósítson meg halmaz absztrakt adattípust egy irányban láncolt listával! A lista elemei tárolják azokat a racionális számokat, amik benne vannak egy halmazban. A lista elemeinek már definiáltunk egy struktúrát a forrásszöveg elején:

```
struct adat {
    double szam;
    adat* kov;
};
```

Készítsen *tartalmaz* függvényt, ami fogadja paraméterként a halmazt megvalósító lista első elemét (horgonyát) címző mutató referenciáját, valamint azt a számot, aminek a halmazban létét ellenőrizni kívánjuk. A visszatérési érték logikai típusú: igaz, ha a keresett szám benne van a halmazban, és hamis, ha nincs benne.

Annak érdekében, hogy a listában történő keresés gyorsabb legyen, az utoljára fellelt elemeket mozgassa mindig a lista elejére! Így a gyakran kikeresett elemek kevesebb listaelem átnézésével elérhetőek lesznek.

Ha a fordító szintaktikai hibát jelez a forráskód valamelyik sorában, akkor abból 20-at kivonva kapja meg a szerkesztőben olvasható sorszámot.

**For example:**

Test	Result
<pre>adat* horgony = NULL; beszur(horgony, 3); beszur(horgony, 2); beszur(horgony, 1); kiir(horgony); cout &lt;&lt; 4 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 4)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 1 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 1)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); felszabadit(horgony);</pre>	<pre>1 2 3 4? Nem eleme 1 2 3 3? Eleme 3 1 2 3? Eleme 3 1 2 1? Eleme 1 3 2</pre>
<pre>adat* horgony = NULL; beszur(horgony, 3); beszur(horgony, 2); beszur(horgony, 1); beszur(horgony, 0); beszur(horgony, -1); beszur(horgony, -2); beszur(horgony, -3); kiir(horgony); cout &lt;&lt; 42 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 42)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; -3 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, -3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 2 &lt;&lt; "? " &lt;&lt; (tartalmaz(horgony, 2)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); felszabadit(horgony);</pre>	<pre>-3 -2 -1 0 1 2 3 42? Nem eleme -3 -2 -1 0 1 2 3 3? Eleme 3 -3 -2 -1 0 1 2 3? Eleme 3 -3 -2 -1 0 1 2 -3? Eleme -3 3 -2 -1 0 1 2 2? Eleme 2 -3 3 -2 -1 0 1</pre>

**Answer:** (penalty regime: 0 %)

Reset answer

```
1 bool tartalmaz(adat*& horgony, int szam) {
2
3     bool helyes = false;
4
5     if (horgony == NULL) {
6         return horgony;
7     }
8
9     adat* akt = horgony;
10    while(akt != NULL) {
11        if (akt->szam == szam) {
12            helyes = true;
13        }
14        else {
15            helyes = false;
16        }
17        akt = akt->kov;
18    }
19    return helyes;
20 }
```

Ellenőrzés

	Test	Expected	Got	
✖	<pre> adat* horgony = NULL; beszur(horgony, 3); beszur(horgony, 2); beszur(horgony, 1); kiir(horgony); cout &lt;&lt; 4 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 4)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 1 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 1)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); felszabadit(horgony); </pre>	<pre> 1 2 3 4? Nem eleme 1 2 3 3? Eleme 3 1 2 3? Eleme 3 1 2 1? Eleme 1 3 2 </pre>	<pre> 1 2 3 4? Nem eleme 1 2 3 3? Eleme 1 2 3 3? Eleme 1 2 3 1? Nem eleme 1 2 3 </pre>	✖
✖	<pre> adat* horgony = NULL; beszur(horgony, 3); beszur(horgony, 2); beszur(horgony, 1); beszur(horgony, 0); beszur(horgony, -1); beszur(horgony, -2); beszur(horgony, -3); kiir(horgony); cout &lt;&lt; 42 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 42)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 3 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; -3 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, -3)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); cout &lt;&lt; 2 &lt;&lt; "?" &lt;&lt; (tartalmaz(horgony, 2)?"Eleme":"Nem eleme") &lt;&lt; ' '; kiir(horgony); felszabadit(horgony); </pre>	<pre> -3 -2 -1 0 1 2 3 42? Nem eleme -3 -2 -1 0 1 2 3 3? Eleme 3 -3 -2 -1 0 1 2 3? Eleme 3 -3 -2 -1 0 1 2 -3? Eleme -3 3 -2 -1 0 1 2 2? Eleme 2 -3 3 -2 -1 0 1 </pre>	<pre> -3 -2 -1 0 1 2 3 42? Nem eleme -3 -2 -1 0 1 2 3 3? Eleme -3 -2 -1 0 1 2 3 3? Eleme -3 -2 -1 0 1 2 3 -3? Nem eleme -3 -2 -1 0 1 2 3 2? Nem eleme -3 -2 -1 0 1 2 3 </pre>	✖

Some hidden test cases failed, too.

Your code must pass all tests to earn any marks. Try again.

Show differences

[◀ Mintavizsga](#)

Ugrás...

[Előadás fóliák ▶](#)