
Dinamikus rendezett minta

Dinamikus rendezett mintáról beszélünk, ha rendezett elemtípusú adatelemek halmazán az alábbi műveleteket kell végezni:

- **Adat(x)**: az x paraméterben kapott elemet hozzáveszi a halmazhoz. Ha az x elem már korábban szerepelhetett **Adat(x)** műveletben, akkor ez a művelet hatástalan.
- **Elemszam**: a halmaz elemeinek a számát adja.
- **Kadik(k)**: Az eddig beérkezett adatelemek közül a rendezés szerinti k -adikat adja eredményül. Feltétel: $1 \leq k \leq \text{Elemszam}$.
- **Hanyadik(x)**: Ha az x adat nem szerepelt eddig **Adat(x)** műveletben, akkor az eredménye 0, egyébként az x elem rendezésbeli sorsszámát.

Feladat

Írjunk olyan programot, amely megvalósítja a négy műveletet!

Megvalósítás

Pascal program esetén

```
uses rendninta;
```

Műveletek Pascal deklarációja

```
procedure Adat(x:longint);  
function Elemszam: longint;  
function Kadik(k:longint) : longint;  
function Hanyadik(x:longint) : longint;
```

A műveletek C/C++ deklarációja

```
#include "rendminta.h"  
  
void Adat(int x);  
int Elemszam();  
int Kadik(int k);  
int Hanyadik(int x);
```

Gyakorlás

A `minta.zip` fájlban letölthető egy üres minta.

Korlátozások

A számok értéke legfeljebb 10 000 000.

Az **Adat** és a **Kadik** és **Hanyadik** függvényeket legfeljebb 100000-szer hívják.

Időlimit: 0.3 mp.

Memórialimit: 64 MiB

A tesztek 30%-ában a számok értéke < 10000 .

A programod nem írhat és nem olvashat semmilyen állományt, a standard outputra sem írhat!