

---

## Műveletek sorozattal

Tetszőleges, de azonos típusú elemek sorozatán pozíció szerint műveleteket akarunk végezhetünk.

### Feladat

Írjunk olyan programot, amely megvalósítja az alábbi műveletet!

- `Beszur(i, x)`: az `x` paraméterben kapott elemet beszúrja a sorozat  $i$ -edik eleme elé. Feltétel:  $1 \leq i \leq \text{Elemszam}+1$
- `Elemszam`: a sorozat elemeinek számát adja.
- `Kadik(k)`: A sorozat  $k$ -adik elemét adja eredményül. Feltétel:  $1 \leq k \leq \text{Elemszam}$
- `Torol(k)`: A sorozat  $k$ -adik elemét törli. Feltétel:  $1 \leq k \leq \text{Elemszam}$
- `Modosit(k, x)`: A sorozat  $k$ -adik elemét  $x$ -re változtatja. Feltétel:  $1 \leq k \leq \text{Elemszam}$

### Megvalósítás

Pascal program esetén

```
uses sorozat;
```

Műveletek Pascal deklarációja

```
procedure Beszur(longint i; x:longint);  
function Kadik(k:longint) : longint;  
procedure Torol(k:longint);  
procedure Modosit(k:longint; x:longint);  
Elemszam:longint;
```

A műveletek C/C++ deklarációja

```
#include "sorozat.h"  
void Beszur(int i, int x);  
int Kadik(int k);  
void Torol(int x);  
void Modosit(int k, int x);  
int Elemszam();
```

### Gyakorlás

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

### Korlátozások

A számok értéke legfeljebb 10 000 000.

Az `Adat` és a `Kadik` és `Torol` 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!