**Összefésüléses rendezés**

Az alábbi, 10 elemű, rendezetlen, egész számokat tartalmazó tömböt rendezzük. Az alapötlet az, hogy először bontsuk két részre a tömb tartalmát, és ezeket rendezzük külön-külön, majd a két résztömb tartalmát összefésüljük. A részhalmazokat szintén ezen ötlet alapján rendezzük; rekurzív módon, szétbontogatjuk a lehető legkisebbre, ekkor ezeket is rendezzük:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 33 | 6 | 2 | 37 | 17 | 30 | 1 | 25 | 9 | 21 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 33 | 6 | 2 | 37 | 17 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 30 | 1 | 25 | 9 | 21 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 33 |  | 6 |  | 2 | 37 | 17 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 30 |  | 1 |  | 25 | 9 | 21 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  | 2 |  | 37 |  | 17 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | 25 |  | 9 |  | 21 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 30 |  |  |  |  | 9 | 21 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 6 | 33 |  |  |  |  | 17 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | 17 | 33 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 9 | 21 | 25 | 30 |

Miután rendeztük a két listát, össze tudjuk őket fésülni. Összehasonlítjuk a két lista első elemét, majd a kisebbet kiválasztjuk és az eredménytömbbe tesszük:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | 6 | 17 | 33 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | 9 | 21 | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | 6 | 17 | 33 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | **9** | 21 | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | **2** |  |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | **6** | 17 | 33 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | **9** | 21 | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | **6** |  |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | **17** | 33 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | **9** | 21 | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | **9** |  |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | **17** | 33 | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 9 | **21** | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | 9 | **17** |  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | 17 | **33** | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 9 | **21** | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | 9 | 17 | **21** |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | 17 | **33** | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 9 | 21 | **25** | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | 9 | 17 | 21 | **25** |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | 17 | **33** | 37 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 9 | 21 | 25 | **30** |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | 9 | 17 | 21 | **25** | **30** |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | 6 | 17 | **33** | **37** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 9 | 21 | 25 | 30 |

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 2 | 6 | 9 | 17 | 21 | 25 | 30 | **33** | **37** |

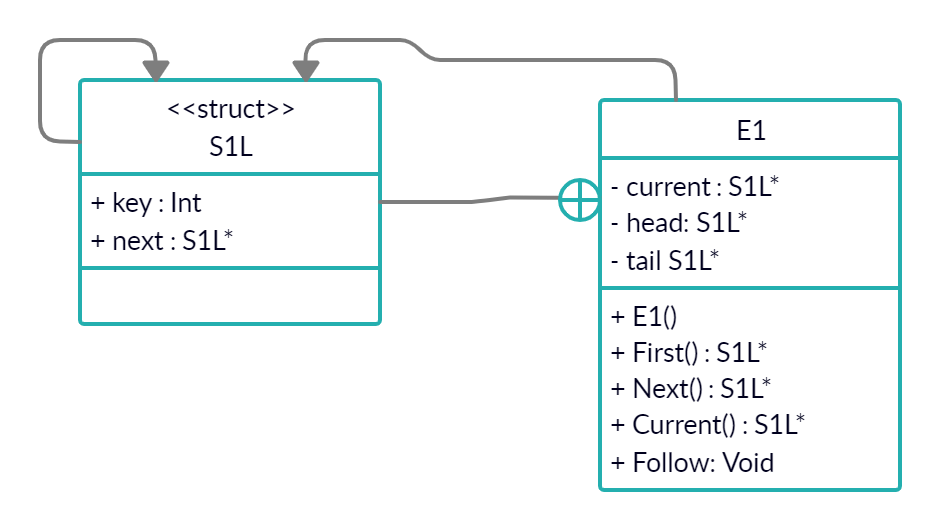
**Lengyelforma:**

Az x2 / (a-b) - (2\*b+a3-13)b kifejezés lengyelformája: 34/x2\*ab-/2b\*a3^-13b^-

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | | | |  |  | |  | |  |  | |  | |  |  | |  | |  |  | |  | |  |  | |  | |  |  | |  | |  | |  |  |  |  |
|  | | | | | | | | | | |  |  | |  | |  |  | |  | |  |  | |  | |  |  | |  | |  |  | |  | | ^ |  | |  | |  | |  |  |  |  |
|  | | | | | | | | | | |  |  | |  | |  |  | |  | |  |  | | - | |  |  | |  | | + |  | | + | | + |  | |  | | - | |  |  |  |  |
|  | | | | | | | | | | |  |  | |  | |  | ^ | |  | |  | ( | | ( | |  |  | | ( | | ( |  | | ( | | ( |  | |  | | ( | |  | ^ |  |  |
|  | | | | | | | | | | |  | / | |  | | \* | \* | |  | |  | / | | / | |  |  | | - | | - |  | | - | | - |  | |  | | - | |  | - |  |  |
|  | 3 | 4 | / | x | 2 | \* | ^ | a | b | - | | | / | | 2 | | | b | | \* | | | a | | 3 | | | ^ | | + | | | 13 | | - | | | b | | ^ | | - | | |  |  |

**Program készítés:**

Adott a P :E1\*[*n*] pointer tömb, amelynek elemei egyszerű láncolt listákat azonosítanak. (Mindegyik vagy 0 pointer, vagy egy lista első elemére mutat.)

Írjuk meg az összefűz(P, L) eljárást, ami a listákat sorban egymás után fűzi az L egyszerű listába. (A formális paraméterek pontos specifikálása a feladat része.) T *(m*, *n*) O(*m + n*), ahol *m* az eredeti listák összhossza (*m* a program számára ismeretlen).

***E1:***

**+ E1(){head = NULL; tail = NULL; current = head;}**

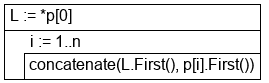
**+ S1L\* First() {current = head; return current;}**

**+ S1L\* Next() {current = current->next; return current;}**

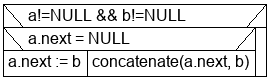
**+ S1L\* Current() {return current;}**

**+ void Follow(int a)  
{  
 S1L\* tmp = new S1L;  
 tmp->key = a;  
 tmp->next = nullptr;  
 if(head==NULL)  
 {  
 head = tmp;  
 tail = tmp;  
 }  
 else  
 {  
 tail->next = tmp;  
 tail = tail->next;  
 }  
 return;  
}**

***Megoldás:***

**Concatenate(int n, vector<E1\* >, E1 &L)**

**concatenate(S1L \*a, S1L \*b)**



***Tesztesetek:***

**Egy elemű lista, több elemű lista, üres lista, üres listák egymás után (input.txt)**