Zadanie: LNAD
Nadlista

Wprawki WP, Runda 8: listy wskaźnikowe.

Dla omawianego na zajęciach typu list prostych

```
struct lista {
  int val;
  struct lista *next;
};
typedef struct lista Tlista;
```

napisz procedurę

```
void nadlista(int N, int A[N], int B[N], Tlista **lista_ptr);
```

która dla danej liczby całkowitej $N > 0$ oraz niemalejących tablic A i B ustawi wartość `*lista_ptr` na nowo utworzoną przez Ciebie listę, która będzie najkrótszą możliwą niemalejącą listą, w której istnieje podlista (podciąg) zawierająca wszystkie elementy z tablicy A oraz podlista (podciąg) zawierająca wszystkie elementy z tablicy B.

Na przykład mając dane:

```
int A[] = {6,6,8,8,11,12};
int B[] = {6,7,7,8,9,11};
Tlista *l;
```

wywołanie

```
nadlista(6, A, B, &l);
```

powinno ustawić zmienną l na listę

```
6 -> 6 -> 7 -> 7 -> 8 -> 8 -> 9 -> 11 -> 12 -> *
```

Zadanie

Należy dostarczyć plik z implementacją funkcji nadlista, używając pliku nagłówkowego lnad.h dostarczonego w pakiecie lnad_public.

Opis zawartości lnad_public (znajduje się też w pliku README.md):

lnad.h - plik nagłówkowy zawierający deklarację typu listowego (jak na zajęciach) oraz deklarację funkcji `nadlista`;

lnadmain.c - plik zawierający funkcję `main`, która uruchamia funkcję `nadlista` na przykładzie wczytanym ze standardowego wejścia i wypisuje rezultat jej działania;

lnad_dummy.c - implementacja funkcji `nadlista`, która działa wyłącznie dla przykładu z treści zadania;

lnad0a.in - plik wejściowy z przykładem z treści zadania.

Aby uruchomić dostarczoną pseudo-implementację funkcji nadlista należy wydać polecenie

```
gcc -o lnad lnadmain.c lnad_dummy.c
```

i następnie uruchomić powstały program wykonywalny

```
./lnad
```

Podobnie, gdy napiszesz własną wersję funkcji nadlista np. w pliku o nazwie lnad17.c należy go skompilować analogiczną komendą:

```
gcc -o lnad lnadmain.c lnad17.c
```

i uruchomić jak wyżej.