

Zadanie 1

Mamy do dyspozycji funkcję o nagłówku

```
bool r(int i, int j),
```

która dla danych liczb całkowitych z domkniętego przedziału $[0, \dots, n-1]$, przyjmuje wartości logiczne takie, że dla każdego $i \neq j$ z tego przedziału mamy $r(i, j) \neq r(j, i)$.

Napisz funkcję

```
bool PoKolei(int A[n]),
```

która, jeśli jest to możliwe, tak wypełni tablicę A permutacją liczb $\{0, 1, \dots, n-1\}$, że dla każdego k należącego do przedziału $[1, \dots, n-1]$ zachodzić będzie warunek $r(A[k-1], A[k]) == \text{TRUE}$, i przekaże wtedy TRUE jako swój wynik lub przekaże wartość FALSE , jeśli to nie jest możliwe; w tym przypadku zawartość tablicy A może być dowolna. Postaraj się wewnątrz funkcji PoKolei wywołać funkcję r jak najmniej razy.

Zadanie 2

Dane są tablice $\text{int } b[n]$ oraz $\text{int } a[k]$ ($n \geq 0$, parzyste, $k == 2*n+1$), obie posortowane ściśle rosnąco. Ponadto zbiór B wartości w tablicy b , jest podzbiorem zbioru A wartości w tablicy a . Napisz funkcję

```
int MedianaRóżnicy(int b[n], int a[k]),
```

która wyznaczy wartość równą medianie zbioru $A \setminus B$, tzn. takiej liczbie $m \in A \setminus B$, że

$$|\{x \in A \setminus B : x < m\}| = |\{x \in A \setminus B : x > m\}| = n/2.$$

Zadanie 3

Dane są deklaracje:

<pre>struct wezel { int w; struct wezel *lsyn, *psyn; }; typedef struct wezel wezel;</pre>	<pre>struct listad { wezel *d; struct lista *nast; }; typedef struct listad listad;</pre>
--	---

Przyjmujemy wariant definicji drzewa BST, w którym wartości w prawym poddrzewie każdego węzła są większe od jego wartości a w lewym są od niej mniejsze lub równe, przy czym równe mogą leżeć tylko na ścieżce samych lewych synów.

Zdefiniuj funkcję

```
listad *usun_powtorzenia(wezel **d);
```

która usunie niektóre węzły z drzewa BST przekazanego jako parametr d , pozostawiając po jednym egzemplarzu każdej wartości. Zbiór wartości drzewa po przekształceniu ma być taki, jak przed nim, drzewo ma nadal spełniać warunek BST i nie może być wyższe, niż przed przekształceniem. Usuniętych z drzewa węzłów nie należy wyrzucać z pamięci, tylko utworzyć z nich posortowaną niemalejąco listę będącą wynikiem działania funkcji.

Zadania oddajemy na osobnych kartkach czytelnie podpisane i skomentowane. Każde rozwiązanie należy uzasadnić i podać pesymistyczny koszt pamięciowy i czasowy.