

Zadanie 1

Dana jest typ `tab=array[0..2*N-1] of Integer`, gdzie $N > 0$. Napisz procedurę

`Rozjazzd(var A:tab; k:Integer),`

która dla zadanego $k > 0$ przesunie cyklicznie wartości pod indeksami parzystymi o k w prawo (tzn. `A[i]` przechodzi na `A[(i+k) mod 2N]`), a wartości pod indeksami nieparzystymi o k w lewo (tzn. `A[i]` przechodzi na `A[(i-k) mod 2N]`). (W powyższym sformułowaniu (ale nie w Pascalu) wynik mod jest zawsze nieujemny)

Zadanie 2

Dane są deklaracje:

```
type      Drzewo = ^Wezel;  
    Wezel = record  
        jest : Boolean;  
        lsyn, psyn : Drzewo  
    end;
```

Jeśli w drzewie `d:Drzewo` węzeł `v` ma pole `jest` ustawione na `true`, to oznacza, że węzeł `v` reprezentuje pewną liczbę całkowitą nieujemną bez znaku. Liczba ta ma wartość, która jest równa w układzie dwójkowym kodowi na ścieżce od korzenia `d` do `v`. Schodząc od korzenia do `v` napotykamy na coraz młodsze bity i przechodząc do lewego syna, piszemy 0, a do prawego 1. Korzeń nie ma lewego syna (więc zaczynamy od jedynki), ale sam może reprezentować liczbę 0. Napisz funkcję

`function maksymalny(d : Drzewo) : Drzewo;`

która przekaże w wyniku wskaźnik do węzła reprezentującego największą wartość spośród reprezentowanych w drzewie. Jeśli w drzewie `d` nie ma żadnej reprezentowanej wartości, to wynikiem powinien być `nil`. Dodatkowe założenie, że wszystkie kodowane liczby mogłyby się zmieścić w zakresie `Integer`, będzie kosztowało sporo punktów, ale nie komplet.

Zadanie 3

Dany jest typ `lista` reprezentujący listę zawierającą wartości typu `char`. Napisz procedurę

`procedure popraw (var l : lista),`

która usunie z `l` minimalną liczbę elementów tak by po jej wykonaniu `l` zawierała poprawne wyrażenie nawiasowe zbudowane z nawiasów jednego typu. Można założyć że w liście `l` występują wyłącznie znaki `' ('` oraz `') '`.

Zadania oddajemy na osobnych kartkach podpisane, z podanymi inicjałami prowadzącego ćwiczenia. Wszystkie rozwiązania należy uzasadnić i podać koszty wykonania algorytmów.