1. Napisz procedurę `lustro(l:listaw)`, która dla jednokierunkowej listy wskaźników `l` w każdym jej elemencie pole `wsk` ustawi na lustrzanie położony element względem niego. Lustrzanie położony element względem x, to element tak samo odległy od końca listy, jak x od początku listy.
2. Dane są 2 listy liczb rzeczywistych posortowane rosnąco. Reprezentują one pewne zbiory liczb. Zaimplementuj podstawowe operacje teoriomnogościowe na tych zbiorach: suma, iloczyn, różnica, różnica symetryczna. Warianty:
   - `function suma(var l1,l2:lista):lista`, która modyfikuje listy l1 i l2 scalając je w jedną posortowaną listę i usuwając ewentualne dublety.
   - `procedure suma(l1,l2:lista; var l3:lista)`, która nie niszcząc list l1 i l2 utworzy nową listę l3 bez dubletów
   - `procedure suma(var l1,l2,l3:lista)`, która zrobi to, co poprzednia, tylko że listy l1 i l2 znikną (w zasadzie bardzo podobne do pierwszej funkcji).

3. Gra polega na następującej zabawie. Uczestnicy ustawiają się na okręgu i każdy przygotowuje kartkę z napisaną liczbą całkowitą. Następnie losuje się osobę A, od której zaczynamy grę. Każdy wygrywa tyle, ile napisał, ale pod warunkiem, że spośród osób, które znajdują się między nim, a osobą A, (zgodnie z ruchem wskazówek zegara) nikt nie napisał liczby mniejszej. W liście l znajdują się kolejno zgłoszone do zabawy liczby. Jako pierwsza występuje liczba napisana przez A, a listę zamyka liczba napisana przez osobę stojącą po prawej ręce A. Należy utworzyć listę wygranych usuwając z listy l liczby przegrane.

4. Dana jest lista znakowa która może zawierać wyrażenie zbudowane z symboli stałej a, symboli 1-argumentowego prefiksowego funktora f oraz symboli 2-argumentowego prefiksowego funktora g. Oto przykłady kilku takich wyrażeń: *a, fa, gaa, gfagfaa.* (prefiksowy oznacza, że nazwa funktora poprzedza argumenty; nawiasów się nie uwzględnia). Napisz funkcję sprawdzającą, czy istnieje początkowy fragment listy, który jest prawidłowo zbudowanym wyrażeniem.

5. Z listy liczb naturalnych usuń te, które nie są sumą kilku bezpośrednich swoich poprzedników.
6. Dana jest lista S taka, że kolejne elementy listy S zawierają albo pewną liczbę rzeczywistą albo wartość MIN, będącą pewną stałą. Załóżmy dodatkowo, że żadne dwie liczby w S się nie powtarzają. Zmodyfikuj listę S tak, że w miejsce każdego symbolu MIN znajdzie się najmniejsza liczba występująca w tablicy S przed danym symbolem która nie zastąpiła wcześniej żadnego symbolu MIN lub stała ∞ jeśli nie ma takiej liczby. Przykładowo:

```
MIN 5 7 MIN 4 MIN 8  3  2 MIN MIN MIN 6 MIN MIN MIN
 ∞  5 7  5  4  4  8  3  2  2   3   7  6  6   8   ∞
```

7. Sprawdź, czy dwie listy zawierają choć jeden wspólny rekord.
8. Znajdź pierwszy wspólny rekord należący do dwóch danych list.

9. Usuń z posortowanej niemalejąco listy wszystkie powtarzające się wartości (z multizbioru zrób zbiór).

10. Zastąp w liście każdy spójny blok składający się z liczb o tych samych znakach jednym elementem o identycznej sumie.

11. (*) W liście cyklicznej są tylko zera i jedynki. Znajdź element, poczynając od którego uzyskamy największą możliwą liczbę interpretując ciąg zer i jedynek jako zapis pewnej liczby naturalnej w systemie dwójkowym. (*) dotyczy rozwiązania liniowego, które wykracza poza zakres materiału wykładu.
12. Element w liście jest chroniony, jeśli bezpośrednio przed nim znajduje się element o większej wartości. Napisz procedurę `TylkoChronione(var l:lista)`, która pozostawi w liście l tylko chronione elementy, przy czym kolejność ich występowania powinna w liście wynikowej być taka, jak na początku działania procedury. Wszystkie elementy niechronione powinny zostać usunięte.