1. Napisać program, który losuje liczby pseudolosowe z zakresu [-150, 150] dopóty, dopóki nie zostanie wylosowana liczba 0. Następnie dokonuje dokładnie tyle samo losowań liczb pseudolosowych z zakresu [-10, 10]. W wyniki działania programu należy wyświetlić stosunek sumy liczb parzystych dodatnich, losowanych w pierwszej kolejności do iloczynu liczb nieparzystych ujemnych, losowanych w drugiej kolejności.

Przykład:

Losowanie 1:	2	-42	11	124	0
Losowanie 2:	-1	7	-3	-2	

Wynik:
42,00000 // (2 + 124) / (-1 * -3) = 126 / 3 = 42

---

2. Napisać funkcję 'konwersja', która jako parametr otrzymuje listę łańcuchów znaków oraz dwie frazy pierwotną i docelową (wielkość liter ma znaczenie). W wyniku działania funkcji zmieniona zostają frazy pierwotne na docelowe.
Jeśli w danym elemencie listy fraza pierwotna występuje kilkukrotnie, to należy skonwertować każde jej wystąpienie.

Ponadto, w elementach listy tablicy o parzystych indeksach wszystkie wielkie litery (alfabetu angielskiego) powinny zostać zamienione na małe, a małe na wielkie.

Funkcja zwraca liczbę dokonanych konwersji, a zmiany dokonane w komórkach listy powinny zostać zachowane.

Przykład, wejście:
[„Ala i Ola”, „Ala i Alan”, „Asia ma kota”, „Kolokwium fala”]
„Ala”
„Asia”

wyjście:
[„Asia i Ola”, „aSIA I aLAN”, „Asia ma kota”, „kOLOKWIUM FALA”]
3

---

3. Napisać program, który dla funkcji kwadratowej $f(x) = ax^2 + bx + c,$ wyznacza pierwiastki równania. Wyznaczanie odbywa się w taki sposób, że dla wczytanych wartości a, b i c wyznaczane są wszystkie pierwiastki równania dla wartości z zakresu [1 – a], [0 – b] i [0 – c]. Pierwiastki mają zostać zapisane w liście, która ma trzy wymiary (jeden dla a, drugi dla b i trzeci dla c), natomiast wartości pierwiastków mają zostać zapisane w postaci krotki (lub listy).

Przykład dla a=2, b=1 i c=1

```
[
  [
    [ (0.0, 0.0), (”brak”, ’brak”) ], # Dla a=1 i b=0, odpowiednio c=0, c=1
    [ (0.0, -1.0), (”brak”, ’brak”) ] # Dla a=1 i b=1, odpowiednio c=0, c=1
  ],
  [
    [ (0.0, 0.0), (”brak”, ’brak”) ], # Dla a=2 i b=0, odpowiednio c=0, c=1
    [ (0.0, -0.5), (”brak”, ’brak”) ] # Dla a=2 i b=1, odpowiednio c=0, c=1
  ]
]
```

lub inaczej:
[[[(0.0,0.0),(”brak”,’brak”)],[(0.0,-1.0),(”brak”,’brak”)]],[[(0.0,0.0),(”brak”,’brak”)],[(0.0,-0.5),(”brak”,’brak”)]]]

W takim przypadku odczyt pierwiastków wyglądałby następująco:
pierwiastki[w1-1][w2][w3]
gdzie w1, w2 i w3 podaje użytkownik, jako kolejne wartości współczynników całkowitych.