

Wstęp do programowania

Lista 11

Premia za tę listę wynosi 0.5, przyznawana jest osobom, które zdobyły co najmniej 2p za zadania z tej listy. Maksimum dla tej listy wynosi 3p. Uwaga: zajęcia w 2020 roku zaczniemy od wprawki. Ta lista jest warta 100% na pierwszych i drugich zajęciach w 2020 roku.

Zadania z kolejnych list będzie można oddawać do końca semestru.

Zadanie 1.(★pkt) Prezent Bożonarodzeniowy: rozwiąż zadanie, którego do tej pory nie robiłeś.

Zadanie 2.(0.5 + 1pkt) Przygotuj własną animację, korzystając z biblioteki turtle (możesz w dowolny sposób korzystać z kodu tworzonego na wykładzie 12. Jedynymi „sztywnymi” wymaganiami jest zachowanie płynności animacji i dodanie zmieniających się płynnie kolorów pewnych elementów animacji, reszta zależy od Twojej inwencji (w szczególności możesz zmienić również tematykę animacji). Jeżeli zachwycisz prowadzącego, może dać Ci uznaniową premię w wysokości do 1p.

Zadanie 3.(1pkt) Zdefiniujmy następujące przekształcenie na słowach (nazwiemy je *permutacyjną postacią normalną*): zamieniamy litery na liczby, w ten sposób, że:

1. Tym samym literom przypisane są równe liczby, różnym literom – różne liczby.
2. Liczby przypisywane są po kolei, licząc od lewej strony.

Otrzymane liczby sklejamy w jeden napis, wstawiając na przykład znak "-" jako separator. Przykładowe pary słowo i wartość przekształcenia: tak: 1-2-3, nie: 1-2-3, tata: 1-2-1-2, indianin: 1-2-3-1-4-2-1-2. Napisz funkcję, która zwraca w wyniku wartość opisanego przekształcenia. *To zadanie będzie miało kontynuację na kolejnej liście.*

Zadanie 4.(1pkt) Twoim zadaniem będzie zasymulowanie ostatnich wyborów parlamentarnych w Polsce (opierając się na rzeczywistych wynikach z poszczególnych okręgów, które będą umieszczone na stronie wykładu¹. W ostatnich (i w wielu innych) wyborach w Polsce obowiązywała metoda D'Hondta przydzielania mandatów. Poniżej skrócony opis tej metody (w razie potrzeby możesz zajrzeć na przykład do Wikipedii, gdzie jest ona dokładniej opisana).

- Każdy komitet ma wynik (będący liczbą rzeczywistą), rozdzielamy M mandatów pomiędzy N komitetów, zależnie od wyniku danego komitetu.
- Tworzymy **ilorazy** tych wyników przez kolejne liczby całkowite od 1 do M (dla każdego ilorazu zapamiętujemy, z jakiego komitetu pochodzi).
- Sortujemy te $M \times N$ liczb malejąco. Bierzymy pierwsze M i przydzielamy odpowiednim komitetom M mandatów.

Pomiń kolumnę Inne (to komitety, które nie przekroczyły progu wyborczego). Oblicz sumaryczną liczbę mandatów dla każdego ugrupowania. Sprawdź, czy otrzymałeś te same wyniki, co Państwowa Komisja Wyborcza (i ewentualnie zaalarmuj odpowiednie organy o wykrytym przestępstwie).

Zadanie 5.(1p ★pkt) Zadanie to wymaga rozwiązania zadania o relacjach równoważności z poprzedniej listy (możesz je rozwiązać również teraz za pełną liczbę punktów). Załóżmy, na potrzeby tego zadania, że „elektoraty się dodają”, to znaczy, że jeżeli (przykładowo) SLD oraz KWiN zdecydowałyby się na wspólny start, to na taką połączoną listę zagłosowałyby dokładnie te same osoby, co w ostatnich wyborach na pojedyncze komitety. Napisz program, który przeglądając wszystkie relacje równoważności na komitetach wyborczych (możesz skoncentrować się na sześciu, które weszły do sejmu) przeprowadza symulowane wybory dla różnych sojuszków (czyli dla różnych decyzji o połączeniu komitetów)².

Twój program powinien:

- a) Wypisać wszystkie warianty wyborów, w których istnieje koalicyjny komitet posiadający większość mandatów i niezawierający zwycięskiego ugrupowania (czyli PiS).

¹W formacie tsv, czyli tab-separated-values, pamiętaj o konstrukcji `x.split('t')`

²Zwróć uwagę, że ordynacja działa tak, że dodawanie elektoratów nie przekłada się na dodawanie mandatów

- b) Wypisać wszystkie warianty wyborów, w których istnieje koalicyjny komitet posiadający większość konstytucyjną.

Wypisując wariant wyborów wypisz sojusze i liczbę mandatów dla poszczególnych sojuszy.