Ćwiczenie 3: Złożone struktury danych cz. 1.

Zagadnienia:

- Listy, pętla **`for`**, proste algorytmy na wektorach wartości

Uwagi do zadań:

- Na początku każdego programu dodać komentarz opisujący w jednym zdaniu, co program robi.
- W miarę potrzeby dodawać inne komentarze w kodzie źródłowym

Zadanie na rozgrzewkę
Wypisz w pętli kwadraty liczb od 1 do 199 włącznie.
Następnie należy przerobić program, tak by użytkownik mógł wybrać, do jakiej potęgi podnieść liczby.

Zadanie1
Napisać program, który cyklicznie pobiera od użytkownika ciąg znaków (napis bądź liczbę). Jeżeli użytkownik wprowadził liczbę (można wykorzystać metodę isnumeric()) to należy ten ciąg znaków zamienić na liczbę całkowitą (**`int`**) i dodać ją do listy **`liczby`**. W przeciwnym wypadku należy dodać wprowadzony tekst do listy **`wyrazy`**. Następnie program prosi użytkownika o kolejny ciąg znaków.

Jeżeli użytkownik wprowadził ciąg pusty (nacisnął Enter bez wprowadzania znaków), należy zakończyć wprowadzanie i wypisać na ekranie:

- Liczbę wprowadzonych wyrazów,
- Łączną liczbę znaków we wszystkich wprowadzonych wyrazach.
- Liczbę wprowadzonych liczb.
- Sumę wszystkich wprowadzonych liczb.

Zadanie 2 Napisać program, realizujący obliczenia N kolejnych wyrazów następujących ciągów.

a) $a_n = 2 * a_{n-1}$
b) $a_n = (a_{n-1} + a_{n-2})/2$

Obliczenia należy wykonać na dwa sposoby:

1) Zapisując kolejne wartości ciągu do listy
2) Bez zapisywania kolejnych wartości do listy. Takie rozwiązanie wymaga zapamiętywania tylko tych poprzednich wartości, które są konieczne do obliczeń i ich odpowiedniej podmiany.

Zadanie 3. Napisać program wczytujący kolejne liczby całkowite z klawiatury i zapamiętujący te liczby w wektorze. W trakcie wpisywania należy na bieżąco zliczać ilość wprowadzonych jedynek. Program powinien zakończyć wpisywanie liczb po wprowadzeniu przez użytkownika trzeciej jedynki.

Następnie należy obliczyć sumę oraz średnią wartości zgromadzonych w wektorze.

Uwaga! :
Należy dodać sprawdzenie czy wczytana liczba jest liczba całkowitą.

Zadanie. 4. Napisać program, który dla danego wektora (listy wartości liczbowych) obliczy wartość iloczynu wszystkich wartości.

Jeżeli jakaś wartość w wektorze równa się zeru, to wartości tej nie należy brać jako składnika iloczynu.

Jeżeli przynajmniej jedna wartość w wektorze jest zerem, na zakończenie programu należy wyświetlić użytkownikowi stosowne ostrzeżenie.

Zadanie 5. Napisać program, który dla danej sekwencji nukleotydów (zapisanej jako lista wartości „A" „T" „G" „C", reprezentującej pojedynczą nić DNA, wygeneruje drugą sekwencję nukleotydów komplementarną do zadanej.

Przykład:
Zadana sekwencja: A T T G G C C C T T A A A
Sekwencja komplementarna: T A A C C G G G A A T T T

Zadanie 6. Napisać program, który na podstawie danej sekwencji nukleotydów (zapisanej jako lista wartości „A" „T" „G" „C", reprezentującej pojedynczą nić DNA, oblicza procentową zawartość poszczególnych nukleotydów.

Zadanie 7 * (zadanie demonstrowane na wykładzie). Napisać program, który dla zadanej sekwencji nukleotydów generuje listę opisującą łańcuch polipeptydowy kodowany przez tą sekwencję.

Zadanie 8. Napisać program, który poszukiwał będzie określonej sekwencji wartości w innej sekwencji.

Przykład:
L1 = [1, 2, 3]
L2 = [1, 2, 1, 2, 3, 2, 1, 2, 3, 2, 2, 1, 4, 3, 4, 5,4 ,1, 2, 3, 4]

Program powinien zwracać indeksy wszystkich miejsc, w których znajdują się poszukiwane sekwencje.