

Programowanie I

Zajęcia nr 1

Rafał Masełek

3. marca 2022r.

Na podstawie zadań przygotowanych przez B. Zglinickiego oraz T. Kazimierczuka

Zadanie 0 – Hello World

a)

W trybie interaktywnym wyświetl na standardowe wyjście napis "Hello World".

b)

Zrób to co wyżej, ale teraz stwórz plik z rozszerzeniem .py

c)

Rozszerz program tak, żeby zapytał użytkownika o imię, po którego wprowadzeniu wyświetli komunikat "Hello IMIE"

d)

Rozszerz program, żeby pytał o wiek. Następnie wyświetl komunikat o roku urodzenia (dokonaj konwersji typów i obliczeń), np. "You were born in 1969."

e)

Zmodyfikuj program tak, żeby obliczenia były wykonywane za pomocą funkcji.

Zadanie 1 – BMI

Wskaźnik masy ciała (ang. *Body Mass Index*, *BMI*) to współczynnik stosowany do określania poprawności masy ciała. Oblicza się go zgodnie ze wzorem

$$BMI = \frac{\text{masa ciała}}{\text{wzrost}^2} \quad (1)$$

przy czym masa ciała wyrażona jest w kilogramach, zaś wzrost – w metrach. W przypadku młodej osoby dorosłej wartości BMI interpretowane są w następujący sposób:

poniżej 18,5 – niedowaga
18,5-25 – waga prawidłowa

25 – 30 – nadwaga
powyżej 30 – otyłość

Napisz program *bmi*, który prosi użytkownika o podanie masy ciała w kilogramach i wzrostu w metrach, a następnie wypisuje wartość BMI odpowiadającą tym danym oraz jej interpretację. Pamiętaj, żeby sprawdzić czy podane przez użytkownika wartości mają sens, a jeśli nie mają to wyświetl stosowne komunikaty i pytaj do skutku.

Przykładowe wykonanie:

Podaj masę ciała w kilogramach

77

Podaj wzrost w metrach

1.8

Twoje BMI wynosi: 23.77

Waga prawidłowa

Zadanie 2 – Rozwiązywanie równań kwadratowych (i liniowych)

Napisz program *qeq*, którego celem jest rozwiązywanie równań kwadratowych. Program pyta użytkownika o 3 rzeczywiste współczynniki i zwraca pierwiastki równania. Jeżeli pierwiastki są tożsame, to wyświetla tylko jeden. Program ma działać również w przypadku zdegenerowanym, gdy równanie kwadratowe przechodzi w liniowe. Uwzględnij rozwiązania zespolone.

Przykładowe wykonanie:

To jest program do rozwiązywania równań postaci $ax^2 + bx + c = 0$

Podaj współczynnik a

-2

Podaj współczynnik b

4

Podaj współczynnik c

1

Równanie kwadratowe $-2.0x^2+4.0x+1.0$

Dwa rozwiązania:

$x_1=2.224744871391589$

$x_2=-0.22474487139158894$

Zadanie 3 – Algorytm Euklides

Zaimplementuj algorytm Euklidesa do szukania największego wspólnego dzielnika dwóch liczb całkowitych dodatnich. Opis algorytmu. Wypisz NWD na standardowe wyjście.

Zadanie 4 – najmniejsza różnica liczb zmiennoprzecinkowych

Napisz program *epsilon*, który znajduje i wypisuje na ekranie najmniejszą liczbę, jaką należy dodać do liczby zmiennoprzecinkowej 1.0, by wynik został zinterpretowany jako liczba zmiennoprzecinkowa większa od 1.0.

Zadanie 5 – nextPrime – Następna liczba pierwsza.

Napisz funkcję *nextPrime*, która przyjmuje jako argument liczbę naturalną *n* i zwraca najmniejszą liczbę pierwszą większą od *n*. Korzystając z tej funkcji, napisz program *nextPrime*, który prosi użytkownika o podanie liczby naturalnej *n*, a następnie wypisuje najmniejszą liczbę pierwszą większą od *n*.

Zadanie 6 – Ciąg Fibonacciego

Ciąg Fibonacciego zdefiniowany jest następująco:

$$F_n = \begin{cases} 1, & \text{dla } n = 1 \text{ lub } n = 2 \\ F_{n-1} + F_{n-2}, & \text{w pp} \end{cases} \quad (2)$$

Pierwsze kilka wyrazów ciągu: (1,1,2,3,5,8,13,21)

a)

Napisz program fib, który prosi użytkownika o podanie liczby naturalnej n , a następnie oblicza i wypisuje n -ty wyraz ciągu Fibonacciego.

b)

Napisz program fibsum obliczający i wypisujący sumę wyrazów ciągu Fibonacciego spełniających jednocześnie dwa warunki: wskaźnik wyrazu jest parzysty, a wartość wyrazu jest mniejsza od 3×10^6 .

Zadanie 7 – Odwzorowanie logistyczne

Napisz program, który przeprowadzi symulację rozwoju populacji zwierząt wg modelu logistycznego. Model jest zdefiniowany poprzez ciąg rekurencyjny:

$$x_{n+1} = ax_n(1 - x_n), \quad (3)$$

gdzie $0 < a < 4$ jest parametrem wzrostu, a $0 \leq x_k \leq 1$ populacją unormowaną do jedynki ($x = 1$ odpowiada zużyciu całych zasobów i maksymalnej dopuszczalnej wielkości populacji).

Program powinien pobierać wartości a i x_0 oraz sprawdzić ich zakresy. Program powinien również prosić o podanie liczby iteracji (≥ 0). Do liczenia wyrazów ciągu użyj rekurencji. Wynikiem działania programu powinna być informacja o numerze iteracji i wielkości populacji (każda iteracja w nowej linii). W przypadku podania niepoprawnych argumentów program powinien kończyć pracę z wyświetleniem komunikatu o błędzie.

Przykład działania:

ii x

0 0.2

1 0.112

2 0.0696192

3 0.0453407

4 0.0302994

5 0.020567

6 0.0141008

7 0.00973136

8 0.00674566

9 0.00469011

Zaobserwuj następujące zjawiska:

1. Dla $0 < a < 1$ populacja wyginie. Oznacza to, że ciąg będzie zbieżny do zera. Fachowo mówi się, że zero jest *atraktorem*.
2. Dla $1 < a < 2$ populacja zbiega szybko do $\frac{a-1}{a}$ niezależnie od początkowej populacji.
3. Dla $2 < a \leq 3$ populacja zbiega powoli do $\frac{a-1}{a}$ wcześniej oscylując.
4. Dla $3 < a < 3.44949$ populacja będzie oscylować wokół dwóch wartości zależnych od r . Rozszczepienie atraktora na dwa nazywa się *bifurkacją*.
5. Dla $3.44949 < a < 3.54409$ (*approx*) populacja będzie oscylować wokół czterech wartości zależnych od r . Kolejna bifurkacja. Dla większych wartości będzie 8, 16, 32 punkty oscylacji itd.
6. Dla $a > 3.56995$ większość¹ wartości a i x_0 prowadzi do zjawiska *chaosu deterministycznego* (atraktor nazywany chaotycznym). Oznacza to, że chociaż kolejne wyrazy ciągu są ściśle zdefiniowane, to ciąg wydaje się być losowy, nie występują w nim oscylacje (można powiedzieć, że są oscylacje o okresie nieskończonym).

Wartość x_0 ma mniejsze znaczenie dla zachowania ciągu niż wartość a .

Przy sprawdzaniu unikaj podawania brzegowych wartości a . Zaczynj od małego n , powiedzmy 30. Jeżeli nie widzisz zbieżności/oscylacji to zwiększ n kilka razy.

Zadanie 8 – pomiar czasu wykonywania operacji arytmetycznych

Napisz program `speed`, który prosi użytkownika o podanie dwóch liczb zmiennoprzecinkowych, następnie mierzy czasy wykonywania dwóch operacji arytmetycznych na tych liczbach: dodawania i mnożenia, po czym wypisuje na ekranie wyznaczone czasy i ich różnicę.

Zadanie 9 – silnia

Napisz funkcję `ifactorial`, która przyjmuje jako argument liczbę naturalną n , oblicza silnię tej liczby, korzystając z odpowiedniego algorytmu iteracyjnego, a następnie zwraca otrzymany wynik.

Napisz także funkcję `rfactorial`, która działa podobnie, jak funkcja `ifactorial`, wykorzystując jednak do obliczenia silni odpowiedni algorytm rekurencyjny.

Korzystając z obu tych funkcji, napisz program `factorial`, który prosi użytkownika o podanie liczby naturalnej n , a następnie dwukrotnie, iteracyjnie i rekurencyjnie, oblicza silnię liczby n , za każdym razem wypisując wynik obliczeń oraz czas wykonania kodu obliczającego silnię.

Zadanie 10 – Ciąg Collatza

Niech k będzie liczbą naturalną. Ciągiem Collatza nazywamy ciąg $(c_n^k)_{n=0}^\infty$ określony wzorem

$$c_n^k = \begin{cases} k & \text{gdy } n=0, \\ \frac{1}{2}c_{n-1} & \text{gdy } c_{n-1} \text{ jest liczbą parzystą} \\ 3c_{n-1} + 1 & \text{gdy } c_{n-1} \text{ jest liczbą nieparzystą} \end{cases} \quad (4)$$

Na przykład:

$$(c_n^{12}) = (12, 6, 3, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots), \quad (5)$$

¹Dla niektórych wartości parametrów istnieją tzw. okna stabilności, gdzie nie obserwuje się chaosu. Jeżeli trafiłeś na takie wartości to gratulacje! Zainteresowanych odsyłam np do Wikipedii. Jest też wiele fajnych książek o chaosie na różnym poziomie zaawansowania matematycznego. Sam chaos deterministyczny wywodzi się z badań nad prognozowaniem pogody, ale pojawia się w wielu dziedzinach fizyki, a jak pokazuje przykład również w innych naukach.

$$(c_n^{13}) = (13, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots), \quad (6)$$

$$(c_n^{15}) = (15, 46, 23, 70, 35, 106, 53, 160, 80, 40, 20, 10, 5, 16, 8, 4, 2, 1, 4, 2, 1, 4, 2, 1, 4, 2, 1, \dots). \quad (7)$$

Gdy pewien wyraz ciągu Collatza ma wartość 1, wyrazy następujące po nim mają wartości 4, 2, 1, 4, 2, 1, . . . – sekwencja 4, 2, 1 powtarza się w nieskończoność. Przypuszcza się, że ciąg Collatza osiąga wartość 1 dla dowolnego k , hipoteza ta pozostaje jednak problemem otwartym.

Napisz program `collatz`, który prosi użytkownika o podanie liczby naturalnej k , a następnie oblicza i wypisuje początkowe wyrazy ciągu Collatza aż do pierwszego z wyrazów o wartości 1 włącznie.