

Programowanie i metody numeryczne

Ćwiczenia 1.

Podstawy programowania – powtórzenie.

Zadanie 1. Kula trójwymiarowa.

W przypadku trójwymiarowej kuli o promieniu R prawdziwe są następujące wzory:

$$\begin{aligned} \text{pole powierzchni: } P &= 4\pi R^2, \\ \text{objętość: } V &= \frac{4}{3}\pi R^3. \end{aligned}$$

- a) Napisz program `balli`, który wczytuje ze standardowego wejścia liczbę rzeczywistą R , a następnie oblicza i wypisuje na standardowe wyjście pole powierzchni i objętość trójwymiarowej kuli o promieniu R .
- b) Napisz program `balla`, który przyjmuje jako argument wywołania liczbę rzeczywistą R , a następnie oblicza i wypisuje na standardowe wyjście pole powierzchni i objętość trójwymiarowej kuli o promieniu R .

Zadanie 2. bmi – Wskaźnik masy ciała.

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

$$\text{BMI} = \frac{\text{masa ciała w kilogramach}}{(\text{wzrost w metrach})^2}.$$

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 wczytuje ze standardowego wejścia masę ciała w kilogramach i wzrost w metrach, a następnie wypisuje na standardowe wyjście wartość BMI odpowiadającą tym danym oraz jej interpretację.

Przykładowe wykonanie

Wejście

Podaj masę ciała (w kilogramach): 70

Podaj wzrost (w metrach): 1.8

Wyjście

BMI = 21.60

Waga prawidłowa.

Zadanie 3. qeq – Rozwiązywanie równań kwadratowych.

Napisz program `qeq`, który wczytuje ze standardowego wejścia rzeczywiste współczynniki $a \neq 0$, b i c , a następnie wypisuje na standardowe wyjście wszystkie (również zespolone) rozwiązania równania kwadratowego $ax^2 + bx + c = 0$.

Zadanie 4. series – Szeregi.

Napisz funkcję `sum_a`, która przyjmuje jako argumenty liczbę rzeczywistą x i liczbę naturalną n oraz oblicza i zwraca sumę szeregu

$$A_n(x) = \sum_{k=0}^n \frac{x^k}{k!}.$$

Napisz także funkcję `sum_b`, która przyjmuje jako argumenty liczbę rzeczywistą x i liczbę naturalną n oraz oblicza i zwraca sumę szeregu

$$B_n(x) = \sum_{k=0}^n (-1)^k \frac{x^{2k}}{(2k)!}.$$

Korzystając z obu tych funkcji, napisz program `series`, który przyjmuje jako argumenty wywołania liczbę rzeczywistą x i liczbę naturalną n , a następnie oblicza i wypisuje na standardowe wyjście sumy określonych powyżej szeregów $A_n(x)$ i $B_n(x)$.

Definicje funkcji `sum_a` i `sum_b` powinny znajdować się w osobnym pliku.

Zadanie 5. pi – Obliczanie liczby π .

Jednym ze sposobów obliczania wartości liczby π jest wykorzystanie tzw. *wzoru Machina*

$$\frac{\pi}{4} = 4\arctg\frac{1}{5} - \arctg\frac{1}{239}$$

w połączeniu z rozwinięciem funkcji arcus tangens w szereg Maclaurina:

$$\arctg x = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} x^{2n+1} = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots, \quad \text{gdzie } |x| \leq 1.$$

Napisz program `pi`, który przyjmuje jako argument wywołania liczbę całkowitą N , a następnie oblicza powyższą metodą liczbę π , biorąc K pierwszych wyrazów rozwinięcia funkcji arcus tangens, dla $K = 1, 2, \dots, N$. Program powinien narysować wykres otrzymanych wartości liczby π w funkcji K , nanieść na nim prostą $y = \pi$, zaś w tytule wykresu umieścić oszacowanie π dla $K = N$.

Zadanie 6. factorial – 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 przyjmuje jako argument wywołania liczbę naturalną n , a następnie dwukrotnie, iteracyjnie i rekurencyjnie, oblicza silnię liczby n , za każdym razem wypisując na standardowe wyjście wynik obliczeń oraz czas wykonania kodu obliczającego silnię.

Korzystając z obu tych funkcji, napisz także program `factorialgraph`, który przyjmuje jako argument wywołania liczbę naturalną N , a następnie przygotuje i zapisze na dysku dwa rysunki: pierwszy powinien zawierać wykresy wartości oraz czasu znajdowania silni w funkcji n dla algorytmu iteracyjnego i $n = 1, 2, \dots, N$, zaś drugi – to samo, jednak dla algorytmu rekurencyjnego.

Definicje funkcji `ifactorial` i `rfactorial` powinny znajdować się w osobnym pliku i być wykorzystywane w obu programach.

Zadanie 7. Ciąg Fibonacciego.

Ciągiem Fibonacciego nazywamy ciąg $(F_n)_{n=1}^{\infty}$ określony wzorem

$$F_n \stackrel{\text{def}}{=} \begin{cases} 1, & \text{gdy } n = 1 \text{ lub } n = 2, \\ F_{n-1} + F_{n-2}, & \text{w pozostałych przypadkach.} \end{cases}$$

Piętnaście początkowych wyrazów tego ciągu to

$$(F_n) = (1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, \dots).$$

Napisz funkcję `fibi`, która przyjmuje jako argument liczbę naturalną n , oblicza n -ty wyraz ciągu Fibonacciego, korzystając z odpowiedniego algorytmu iteracyjnego, a następnie zwraca otrzymany wynik.

Napisz także funkcję `fibr`, która działa podobnie, jak funkcja `fibi`, wykorzystując jednak do obliczenia n -tego wyrazu ciągu Fibonacciego odpowiedni algorytm rekurencyjny.

Korzystając z obu tych funkcji, napisz program `fib`, który przyjmuje jako argument wywołania liczbę naturalną n , a następnie dwukrotnie, iteracyjnie i rekurencyjnie, oblicza n -ty wyraz ciągu Fibonacciego, za każdym razem wypisując na standardowe wyjście wynik obliczeń oraz czas wykonania kodu obliczającego wyraz ciągu.

Korzystając z obu tych funkcji, napisz także program `fibgraph`, który przyjmuje jako argument wywołania liczbę naturalną N , a następnie przygotowuje i zapisze na dysku dwa rysunki: pierwszy powinien zawierać wykresy wartości oraz czasu znajdowania n -tego wyrazu ciągu Fibonacciego w funkcji n dla algorytmu iteracyjnego i $n = 1, 2, \dots, N$, zaś drugi – to samo, jednak dla algorytmu rekurencyjnego.

Definicje funkcji `fibi` i `fibr` powinny znajdować się w osobnym pliku i być wykorzystywane w obu programach.

Napisz ponadto program `fibsum` obliczający i wypisujący na standardowe wyjście sumę wyrazów ciągu Fibonacciego spełniających jednocześnie dwa warunki: wskaźnik wyrazu jest parzysty, a wartość wyrazu jest mniejsza od 3×10^6 . Postaraj się wykorzystać jak najbardziej efektywny algorytm.

Opracowanie: Bartłomiej Zglinicki.