

Programowanie proceduralne — lista zadań laboratoryjnych I

1. Napisać program obliczający i wyświetlający sumę liczb wczytanych z konsoli wejściowej (liczba 0 kończy dane). Obudować program pętlą powtarzającą obliczenie dla kolejnej serii danych na życzenie użytkownika.
2. Napisać program generujący n początkowych liczb ciągu Fibonacciego.
3. Program czyta liczby jako dane wejściowe aż do pojawienia się 0. Następnie wyświetla następujące informacje: a) ile było liczb; b) ile było liczb dodatnich; c) średnią arytmetyczną wczytanych liczb; d) największą i najmniejszą z nich; e) odchylenie standardowe podanej populacji.
4. Dla podanych dwóch liczb m i n obliczyć $NWP(m, n)$ oraz $NWW(m, n)$. Obudować program pętlą powtarzającą obliczenie dla kolejnej pary liczb na życzenie użytkownika.
5. Wypisać wszystkie dzielniki liczby n lub stwierdzić, że jest pierwsza.
6. Zbudować tablicę n początkowych liczb pierwszych. Rozwiązać zadanie dwoma sposobami: a) metodą “naiwną”, polegającą na próbnym dzieleniu liczby m kolejno przez liczby od niej mniejsze; b) metodą udoskonaloną, która próbuje dzielić m przez już znalezione liczby pierwsze $p_1, p_2, \dots, p_l \leq \sqrt{m}$. Porównać szybkość działania obydwu wersji programu dla $n = 1\,000\,000$.
7. Podać rozkład danej liczby n na czynniki pierwsze.
8. Program czyta zmiennopozycyjne liczby a , b i c , a następnie wyświetla rozwiązania równania kwadratowego

$$ax^2 + bx + c = 0.$$

Należy czytelnie podać informację o typie rozwiązania zależnie od wartości wyróżnika Δ : dwa pierwiastki rzeczywiste, podwójny pierwiastek rzeczywisty, dwa pierwiastki zespolone.

9. Zaimplementować metodę Herona obliczania pierwiastka kwadratowego z zadanej liczby a .
10. Zaimplementować uogólnioną metodę Herona obliczania pierwiastka n -tego stopnia z a .
11. Napisać program wyznaczający przybliżone rozwiązanie równania $F(x) = 0$ metodą połowienia przedziału. Dla danej funkcji F czytamy początkowe punkty iteracji a i b , sprawdzając czy spełniony jest warunek $F(a)F(b) < 0$. Jeśli tak, pytamy o pożądaną dokładność wyniku $\varepsilon > 0$ i uruchamiamy iterację. Przetestować działanie programu dla następujących funkcji: a) $F(x) = \log x + x$; b) $F(x) = x^2 \sin x$; c) $F(x) = x^5 - 3x + 1$.
12. Wyznaczyć wszystkie “trójki pitagorejskie”, tj. liczby naturalne p, q, r spełniające równanie

$$p^2 + q^2 = r^2$$

w zakresie $1 \leq p, q, r \leq 1000$. a) Udoskonalić program tak, aby nie wypisywał wielokrotnie tych samych trójek i aby liczba wykonywanych operacji była jak najmniejsza. b) Wyeliminować trójki powiązane stałą poporcją, np. 3, 4, 5 oraz 6, 8, 10. W punkcie b) chodzi o to, że jeśli $p^2 + q^2 = r^2$, to automatycznie $(kp)^2 + (kq)^2 = (kr)^2$ dla dowolnego k . Trzeba więc wyeliminować te trójki, w których p, q i r mają wspólny dzielnik.

13. Zaimplementować dowolną metodę obliczania przybliżenia liczby π przez sumowanie odp. szeregu (na podstawie artykułu Wikipedii).
14. Wyznaczanie przybliżenia π metodą Monte Carlo polega na losowym wyborze N punktów o współrzędnych (x, y) w kwadracie $0 \leq x, y \leq 1$ i policzeniu ile z nich, N_0 , znajduje się w ćwiartce koła $x^2 + y^2 \leq 1$. Iloraz N_0/N przybliża proporcję powierzchni ćwiartki koła i kwadratu, skąd można wyznaczyć przybliżenie π . Napisać program wykorzystujący generator liczb pseudolosowych `rand()` do wyznaczenia przybliżenia π opisaną metodą. Porównać uzyskane przybliżenia dla $N = 100, 1000, \dots, 100\,000\,000$.
15. Napisać program, który symuluje ciąg 10000 rzutów parą kostek i oblicza procentową częstość występowania poszczególnych sum oczek. Wyniki należy zebrać w tablicy typu `float, T[2], ..., T[12]` i wyświetlić.
16. Wyniki zadania 15 należy przedstawić w postaci histogramu narysowanego za pomocą spacji i gwiazdek o “pionowych” słupkach. Słupek odpowiadający najczęściej występującej sumie oczek powinien mieć 20 wierszy wysokości. Pod każdym słupkiem należy wypisać odpowiadającą mu wartość procentową `T[i]`.
17. Napisać program, który przeprowadzi symulację losowania Lotto (wybieramy 6 liczb z 49) i wypełnienia 1000 kuponów, a następnie poinformuje, ile razy trafiono 3, ile razy 4 itd.