

Programowanie proceduralne

Laboratorium 2-3 – Funkcje rekurencyjne

Wyświetlanie rekurencyjne

1. Zaimplementuj funkcję rekurencyjną pobierającą jako argument liczbę całkowitą dodatnią n , która wyświetla rekurencyjnie liczby całkowite od 1 do n .
2. Zmodyfikuj funkcję z zadania 1, aby wyświetlała liczby w odwrotnej kolejności (malejąco: od n do 1).

Suma liczb

3. Zaimplementuj funkcję rekurencyjną pobierającą jako argument liczbę całkowitą dodatnią n , która oblicza sumę liczb od 1 do n . Zademonstruj działanie funkcji w programie.

Wskazówka: zauważ, że $\sum_{i=1}^n i = \sum_{i=1}^{n-1} i + n$.

Algorytmy rekurencyjne

4. Zaimplementuj funkcję rekurencyjną obliczającą wartość n -tego wyrazu ciągu Fibonacciego. Zademonstruj jej działanie w programie.
5. Do programu z poprzedniego zadania dodaj zmienną globalną, w której będziesz zliczać liczbę uruchomień funkcji (przy każdym wywołaniu funkcji). Sprawdź, ile razy jest wywoływana funkcja rekurencyjna dla różnych n . Czy obliczanie n -tego wyrazu ciągu Fibonacciego funkcją rekurencyjną jest dobrym pomysłem?
6. W programie z poprzedniego zadania dodaj drugi argument do funkcji obliczającej n -ty wyraz ciągu Fibonacciego. Argument ten wykorzystaj do obliczania głębokości rekurencji. Przy każdym uruchomieniu funkcji, wyświetlaj komunikat zawierający wartość liczbę n , głębokości rekurencji oraz liczbę wszystkich uruchomień funkcji.
7. Zaimplementuj funkcję rekurencyjną, która pozwala wyznaczać całkowite dodatnie potęgi liczb. Zademonstruj działanie funkcji w programie.
8. Napisz program, który prosi użytkownika o podanie dwóch liczb całkowitych dodatnich, a następnie znajduje ich największy wspólny dzielnik. Obliczanie największego wspólnego dzielnika powinno być przeprowadzone w funkcji rekurencyjnej.

Algorytmy rekurencyjne – algorytmy iteracyjne

9. Zaimplementuj funkcję iteracyjną pobierającą jako argument liczbę całkowitą dodatnią n , która wyświetla liczby całkowite od 1 do n .
10. Zaimplementuj funkcję iteracyjną obliczającą wartość n -tego wyrazu ciągu Fibonacciego. Z продемонstruj jej działanie w programie.

Wieże Hanoi

W problemie wież Hanoi mamy wieżę zbudowaną z n -klocków (różnych rozmiarów), które ułożono od największego do najmniejszego. Klocki są nałożone na palik. Chcemy przenieść wieżę na inny palik. Zasady przekładania klocków są następujące:

- klocki możemy przenosić pojedynczo,
- nie możemy układać klocków większych na mniejszych.

Mamy do dyspozycji dodatkowy palik. Jakie ruchy powinniśmy wykonywać, aby przenieść całą wieżę?

11. Zaimplementuj funkcję rekurencyjną, która wypisuje sekwencję ruchów, którą należy wykonać, aby przenieść n -klocków pomiędzy dwoma wskazanymi palikami. Funkcja powinna przyjmować 4 argumenty: n (liczbę klocków) oraz oznaczenia trzech palików (początkowego, końcowego i pomocniczego).

Choinka

12. Napisz funkcję rekurencyjną, która wypisuje na ekranie choinkę z gwiazdek. Przykładowo, dla $n = 5$ choinka powinna wyglądać następująco:

```
  *
 ***
*****
*****
*****
```

Wskazówka: przekazuj jako drugi argument głębokość rekurencji, która mówi o tym o ile spacji należy odsunąć choinkę od krawędzi ekranu.