

WSTĘP DO PROGRAMOWANIA

Laboratorium 8

Rekurencja

Zadania przygotowały: dr Anna Nenca, Laura Grzonka

Funkcje

Rekurencja to odwoływanie się do samego siebie. W programowaniu mówimy o funkcjach rekurencyjnych, odwołujących się same do siebie w swoich definicjach.

Wszystkie poniższe zadania rozwiąż, tworząc funkcje rekurencyjne.

Część I

Zad. 1. Napisz funkcję rekurencyjną `silnia` wiedząc, że:

```
silnia(0) = 1,  
silnia(n) = silnia(n - 1) * n
```

Zad. 2. Napisz funkcję rekurencyjną, która policzy n -ty wyraz ciągu Fibonacciego wiedząc, że:

```
fib(0) = 1,  
fib(1) = 1,  
fib(n) = fib(n - 1) + fib(n - 2)
```

Zad. 3. Napisz funkcję rekurencyjną, która policzy dla danego n sumę

$$\sum_{k=1}^n \frac{1}{k}$$

Zad. 4. Napisz funkcję rekurencyjną, która policzy n^2 bez użycia potęgowania i mnożenia.

Zad. 5. Napisz funkcję rekurencyjną, która policzy sumę liczb parzystych od 1 do n .

Zad. 6. Napisz funkcję rekurencyjną, która sprawdza czy x znajduje się w liście posortowanej.

Zad. 7. Napisz funkcję rekurencyjną, która liczy a do potęgi b bez użycia potęgowania.

Zad. 8. Napisz funkcję rekurencyjną, która liczy największy wspólny dzielnik dwóch liczb ($NWD(a, b)$) korzystając z zależności:

```
NWD(a, a) = a  
NWD(a, a + b) = NWD(a, b)  
NWD(a + b, b) = NWD(a, b)
```

Zad. 9. Napisz funkcję rekurencyjną, która liczy sumę cyfr danej liczby.

Zad. 10. Napisz funkcję `head`, która dla zadanej listy zwraca jej pierwszy element.

Zad. 11. Napisz funkcję `tail`, która dla zadanej listy zwraca inną listę powstałą przez usunięcie pierwszego elementu oryginalnej listy.

Zad. 12. Napisz funkcję rekurencyjną `rewers`, która dla zadanej listy zwraca inną listę, która jest oryginalną listą w odwróconej kolejności. Użyj `head` oraz `tail`. Dodatkowo napisz funkcję `IsEmpty`, która sprawdza, czy dana lista jest pusta.

Zad. 13. Napisz funkcję rekurencyjną, która odwróci ciąg znaków `s`.

Zad. 14. Napisz funkcję rekurencyjną, która oblicza współczynnik dwumianowy, np. `binomial_coefficient(5, 2)` zwraca 10.

Zad. 15. Napisz funkcję rekurencyjną, która sprawdza, czy ciąg `s` jest palindromem.

Zad. 16. Napisz funkcję rekurencyjną `zliczanie(lst, n)`, która zlicza, ile razy liczba `n` występuje w liście `lst`.

Zad. 17. Napisz funkcję rekurencyjną, która zwraca listę wszystkich permutacji ciągu znaków `s`. Np., dla `"abc"` funkcja zwróci `['abc', 'acb', 'bac', 'bca', 'cab', 'cba']`.

Część II

Zad. 18. Napisz funkcję rekurencyjną, która znajduje największy element w liście.

Zad. 19. Napisz funkcję rekurencyjną, która odwraca cyfrę liczby `n` (dla $n \geq 0$), np. dla 1234 zwróci 4321.

Zad. 20. Napisz funkcję rekurencyjną, która rozkłada zagnieżdżoną listę na płaską listę, np. dla listy `[1, [2, [3, 4], 5], 6]` zwróci `[1, 2, 3, 4, 5, 6]`.