

Laboratorium 1.

Celem ćwiczenia jest zapoznanie się ze środowiskiem programowania w j. Java i dokonanie pierwszej kompilacji i uruchomienia programu. Program powinien działać w trybie tekstowym, pobierając argumenty z linii wywołania i wyświetlając wyniki na ekran.

Należy wykonać przynajmniej dwa z poniższych zadań.

Zad. 1

Zaimplementuj funkcję π , która dla danej liczby naturalnej n zlicza ilość liczb pierwszych mniejszych lub równych n . Dla $n > 2$ postać funkcji π jest następująca

$$\pi(n) = -1 + \sum_{j=3}^n \left[(j-2)! - j \left\lfloor \frac{(j-2)!}{j} \right\rfloor \right]$$

gdzie $\lfloor x \rfloor$ oznacza część całkowitą x (*Floor*). Autorami tego rozwiązania są: Hardy i Wright.

Zad. 2

Zaimplementuj algorytm generowania ciągu liczb całkowitych dla n_0 (*Hailstone sequence*)

- 1) weź n_0
- 2) jeśli n_0 jest parzyste, niech $n_1 = n_0/2$
- 3) jeśli n_0 jest nieparzyste, niech $n_1 = n_0 * 3 + 1$
- 4) niech $n_0 = n_1$
- 5) jeśli $n_0 = 1$ zakończ obliczenia, w przeciwnym przypadku skocz do 2

w taki sposób, aby na ekranie pojawiała się:

numer kolejnej iteracji, n_0 {parzyste|nieparzyste}, n_1

np. dla $n_0=10$:

- 1, 10, parzyste, 5
- 2, 5, nieparzyste, 16
- 3, 16, parzyste, 8
- 4, 8, parzyste, 4
- 5, 4, parzyste, 2
- 6, 2, parzyste, 1

Algorytm ten zaproponował Hofstadter w książce „Gödel, Escher, Bach”. Co ciekawe, nie ma dowodu, że algorytm zawsze się zatrzymuje. Jaki jest ciąg wynikowy dla liczby 27?

Zad. 3

Napisz program podający ilość wszystkich rozkładów danej liczby naturalnej n ($0 < n < 1000$) na sumę różnych między sobą składników naturalnych (różnych od 0) występujących w kolejności rosnącej.

Zad. 4

W wyrażeniu „(((1?2)?3)?4)?5)?6” zastąp znaki „?” operatorami działań arytmetycznych na liczbach całkowitych +, -, *, /, aby wartość wyrażenia wynosiła n ($-1000 < n < 1000$). Dla danego n wypisz znalezione wyrażenie lub zgłoś brak rozwiązań.