

Celem ćwiczenia jest zapoznanie się z funkcjami, ich użyciem oraz rekurencją, a także typową organizacją plików w języku C.

2.1 Funkcje obliczające silnię, podwójną silnię $n!!$ i liczby Fibonacciego

Przepisać programy z poprzedniego zestawu zadań tak aby stały się samodzielnymi funkcjami. Wykorzystać definicje rekurencyjne

$$n! = n \times (n - 1)!, \quad 0! = 1! = 1 \quad (1a)$$

$$n!! = n \times (n - 2)!!, \quad 0!! = 1!! = 1 \quad (1b)$$

$$F_n = F_{n-1} + F_{n-2}, \quad F_0 = 0, F_1 = 1 \quad (1c)$$

lub/i już znane nierekurencyjne („sekwencyjne”, „iteracyjne”).

Dla każdej z funkcji zaimplementować dwie wersje (1) przyjmującą na wejściu `int` i zwracającą `int`, oraz (2) przyjmującą na wejściu `int` i zwracającą `double`.

Przykład *interface* (plik `silnia.h`) dla $n!$

```
int    silnia_i(int n);
double silnia_f(int n);
```

2.2 Użycie skompilowanych funkcji, nagłówki i organizacja plików

Przeorganizować kod z Zad. 1 tak, aby funkcje znalazły się w oddzielnym pliku z rozszerzeniem *.c. Utworzyć *pliki nagłówkowe* *.h zawierające definicje funkcji i skompilować osobno plik z funkcjami. Wykorzystać powyższe w programie, który oszacuje szybkość działania (czas wykonania) różnych implementacji z Zad. 1, linkując do niego skompilowane pliki *.o. Spróbować przenieść pliki nagłówkowe oraz pliki binarne poza dotychczas używany katalog roboczy, a następnie przekazać informację o ich lokalizacji opcjami (flagami) kompilatora `gcc`.

2.3 Przekazywanie argumentu i wartości funkcji przez wskaźniki

Napisać funkcję obliczającą wyznacznik lub/i n -tą (całkowitą) potęgę macierzy kwadratowej $N \times N$ o „dowolnym” rozmiarze $N \geq 2$, zawierającą liczby typu `int` lub/i `double`.