

I. Utwórz klasę narzędziową o nazwie `FuncStat`, w której zdefiniowane zostaną statyczne funkcje:

- `public static int fiboR(int n)` wyliczającą rekurencyjnie n -tą liczbę Fibonacciego;
- `public static int fiboI(int n)` wyliczającą iteracyjnie n -tą liczbę Fibonacciego (wykorzystując pętlę);
- `public static int factR(int n)` wyliczającą rekurencyjnie liczbę $n!$;
- `public static int factI(int n)` wyliczającą iteracyjnie liczbę $n!$;
- `public static int gcdR(int a, int b)` wyliczającą rekurencyjnie największy wspólny dzielnik;
- `public static int gcdI(int a, int b)` wyliczającą iteracyjnie największy wspólny dzielnik;
- `public static int maxElem(int[] arr, int from)` zwracającą element dostarczonej tablicy `arr` o największej wartości począwszy od elementu o indeksie `from` (funkcję należy zaimplementować rekurencyjnie);
- `public static int numEven(int[] arr, int from)` – zwracającą ilość elementów parzystych w tablicy `arr`, począwszy od elementu o indeksie `from` (funkcję należy zaimplementować rekurencyjnie);
- `public static void reverse(int[] arr, int from)` – rekurencyjnie odwracającą tablicę `arr`, począwszy od elementu o indeksie `from`.

Działanie wszystkich metod przedstaw na przykładach zdefiniowanych w metodzie `main`.

II. Wprowadź z klawiatury wartość zmiennej `int n` z przedziału od 0 do 26. Następnie utwórz rekurencyjną funkcję, która wypisze na ekranie wszystkie ($n!$ permutacji) liter poczynając od litery `a`. Przykładowo dla $n = 3$ wynik mógłby wyglądać następująco:

```
1 bca cba cab acb bac abc
```