
Zadanie 1

Napisz program wyglądający tak:

```
#include <iostream>

void rekur() {
    // ... ADD SOMETHING HERE ONLY
}

int main() {
    rekur();
}
```

Program powinien powtarzać zachętę do wpisania niezerowej liczby całkowitej aż do momentu, gdy użytkownik wpisze liczbę 0. Wtedy wszystkie wczytane liczby (bez tego kończącego zera) powinny ukazać się w jednej linii w kolejności odwrotnej do tej w jakiej zostały wprowadzone. Na przykład, jeśli użytkownik wprowadził liczby 2, 5, 7, 1, 0, to wynik powinien być

1 7 5 2

Uzupełnij *tylko* definicję *rekurencyjnej* funkcji **rekur**. Nie zmieniaj i nie dodawaj niczego innego, nie używaj żadnych pętli, tablic, napisów ani żadnego rodzaju kolekcji. Zadanie jest łatwe, jeśli nie wymagamy aby wyprowadzone na ekran dane kończyły się znakiem końca linii; tym niemniej da się je rozwiązać tak, aby ten znak był.

Zadanie 2

Napisz program wczytujący liczby całkowite aż do pojawienia się zera i wypisujący długość najdłuższej sekwencji kolejnych liczb o takich samych wartościach (i tę wartość). Na przykład dla

2 2 2 11 11 11 11 2 2 3 3 2 2 12 12

wynik powinien być **Najdluzsza sekwencja: 4 razy liczba 11**.
Nie używaj tablic, napisów ani żadnych innych kolekcji.

Zadanie 3

Napisz program, który wczytuje w pętli ciąg liczb całkowitych do momentu, gdy użytkownik poda liczbę 0, która jest tylko sygnałem końca danych i nie jest dalej brana pod uwagę. Następnie program wypisuje wartości najmniejszego i największego elementu wczytanego ciągu oraz liczbę wystąpień tych wartości w całym ciągu. Na przykład dla ciągu (2, 3, 4, 2, 7, 4, 7, 2) program powinien wypisać

Min = 2 3 razy
Max = 7 2 razy

Uwaga: nie wolno stosować żadnych tablic ani innych kolekcji!

Zadanie 4

Napisz zestaw funkcji do badania liczb pierwszych i dzielników liczb.

- `bool isPrime(int n)` sprawdza, czy podana liczba jest pierwsza;
 - `void printDivisors(int n)` wypisuje wszystkie dzielniki podanej liczby naturalnej (łącznie z jedyneką i samą liczbą);
 - `int numDivisors(int n)` zwraca liczbę dzielników podanej jako argument liczby naturalnej N (łącznie z jedyneką i samą liczbą N);
 - `int gcd(int m, int n)` znajduje największy wspólny dzielnik podanych dwóch liczb używając algorytmu Euklidesa (Księga VII i X *Elementów*):
Dla danych $a, b \in N$:
 - Dopóki $a \neq b$: większą z liczb a, b pomniejsz o wartość mniejszej
 - Zwróć a
 - `int phi(int n)` zwraca dla danej liczby naturalnej N ilość wszystkich liczb z zakresu $[1, N)$ względnie pierwszych z N (dla których największy wspólny dzielnik z N wynosi 1). Jest to słynna funkcja φ Eulera. Na przykład $\varphi(10) = 4$, bo tyle jest liczb naturalnych mniejszych od 10 i względnie pierwszych z 10 (są to liczby 1, 3, 7 i 9).
-