

# Programowanie II R

Zadania – seria 2.

Znak EOF. Tablice.

## Zadanie 1. stats – Statystyka ciągu liczb.

Napisz program `stats`, który wczytuje ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, a następnie wypisuje na standardowe wyjście największą i najmniejszą z nich oraz ich średnie: arytmetyczną i geometryczną. Nie używaj tablic ani kontenerów biblioteki standardowej.

## Zadanie 2. polygonal – Długość łamanej.

Napisz program `polygonal`, który wczytuje ze standardowego wejścia liczby zmiennoprzecinkowe aż do napotkania znaku końca pliku, przy czym ilość tych liczb jest parzysta, a następnie interpretuje kolejne pary liczb jako współrzędne kartezjańskie punktów na płaszczyźnie i wypisuje na standardowe wyjście długość łamanej otwartej z zadanymi punktami jako wierzchołkami. Nie używaj tablic ani kontenerów biblioteki standardowej.

## Zadanie 3. pascal – Trójkąt Pascala.

Napisz program `pascal`, który przyjmuje jako argument wywołania liczbę naturalną  $n$  i wypisuje na standardowe wyjście  $n$  początkowych wierszy trójkąta Pascala.

## Zadanie 4. egyptian – Egipska notacja ułamków.

Napisz program `egyptian`, który przyjmuje jako argumenty wywołania liczby naturalne  $m$  oraz  $n$  i wypisuje na standardowe wyjście ułamek  $m/n$  zapisany w postaci egipskiej, tzn. jako sumę liczby całkowitej i ułamków o liczniku 1.

## Zadanie 5. bubblesort – Sortowanie bąbelkowe.

*Sortowanie bąbelkowe* to jeden z algorytmów sortowania kolekcji danych. Polega on na wielokrotnym przechodzeniu przez kolekcję i porównywaniu dwóch stojących obok siebie elementów – gdy są one ustawione w złej kolejności, zostają zamienione miejscami. Proces sortowania kończy się, gdy podczas przejścia przez kolekcję nie dokonano ani jednej zamiany.

Napisz funkcję `bubbleSort` przyjmującą jako argumenty tablicę danych typu `int` oraz jej rozmiar (typu `size_t`) i zwracającą tablicę złożoną z posortowanych w kolejności rosnącej elementów wyjściowej listy. Funkcja powinna wykorzystywać algorytm sortowania bąbelkowego.

Korzystając z tej funkcji, napisz program `bubblesort`, który wczytuje ze standardowego wejścia liczby całkowite aż do napotkania znaku końca pliku i wypisuje na standardowe wyjście te same liczby w kolejności rosnącej.

## Zadanie 6. `binsearch` – Wyszukiwanie binarne.

*Wyszukiwanie binarne* to jeden z algorytmów wyszukiwania elementu w kolekcji danych, oparty na metodzie *divide et impera* (łac. *dziel i rządź*). Można go stosować wyłącznie dla posortowanych kolekcji. Polega on w uproszczeniu na dzieleniu kolekcji na coraz mniejsze przedziały tak długo, aż długość przedziału będzie równa 1 – wówczas pojedynczym sprawdzeniem można ustalić, czy szukany element należy do konkretnego przedziału. Algorytm ten można realizować rekurencyjnie.

Napisz funkcję `binSearch` przyjmującą pięć argumentów: pierwszym ma być poszukiwany element, drugim – tablica liczb całkowitych, trzecim – jej rozmiar, zaś czwartym i piątym – liczby całkowite `min` i `max`. Funkcja powinna posortować tablicę, a następnie zwrócić indeks, pod którym szukany element występuje na tej liście, biorąc pod uwagę tylko indeksy od `min` do `max` włącznie. Jeśli szukany element występuje wielokrotnie, funkcja może zwrócić dowolny z jego indeksów, jeśli zaś nie występuje wcale, funkcja powinna zwrócić `-1`.

Korzystając z tej funkcji, napisz program `binsearch`, który przyjmuje jako argument wywołania jedną liczbę całkowitą oraz czytuje ze standardowego wejścia liczby całkowite aż do napotkania znaku końca pliku i wypisuje na standardowe wyjście położenie przekazanej jako argument wywołania liczby wśród posortowanych rosnąco liczb wczytanych ze standardowego wejścia.

*Opracowanie: Bartłomiej Zglinicki.*