**SPRAWOZDANIE**

**PROJEKT NA PRZEDMIOT „ALGORYTMY I STRUKTURY DANYCH”**

**TEMAT**

Uporządkuj zadaną tablicę wielu powtarzających się elementów, tak aby zgromadzić te same elementy zachowując kolejność ich pierwszego wystąpienia.

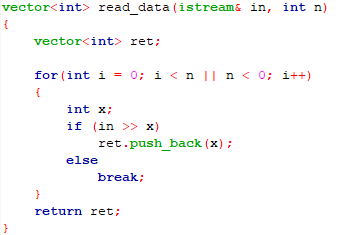
Tomasz Kozioł, Politechnika Rzeszowska, semestr 2020/2021, Inżynieria i Analiza Danych

# Wstęp

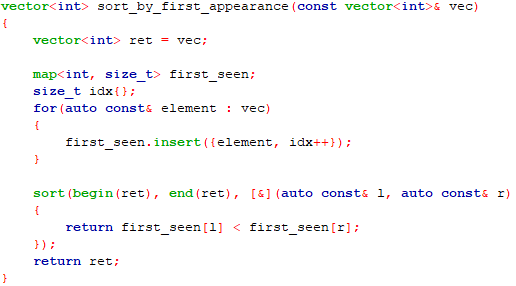
Sprawozdanie dotyczy projektu z przedmiotu Algorytmy i Struktury Danych. Celem projektu było zaimplementowanie algorytmu, który uporządkowuje zadaną tablicę wielu powtarzających się elementów, gromadząc te same elementy przy jednoczesnym zachowaniu kolejności ich pierwszego wystąpienia.

# Opis problemu

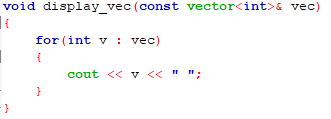
* Program rozpoczyna pracę od wczytania liczb za pomocą wektora, którego wykorzystujemy poprzez dodanie bilbioteki <vector>.



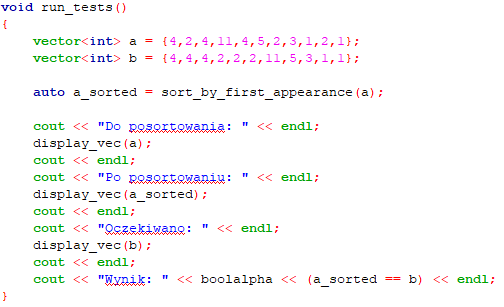
* Następnym krokiem jest stworzenie funkcji sort\_by\_first\_appearance, która zlicza liczby za pomocą map, oraz wbudowanej w bibliotekę <algorithm> funkcję sort.



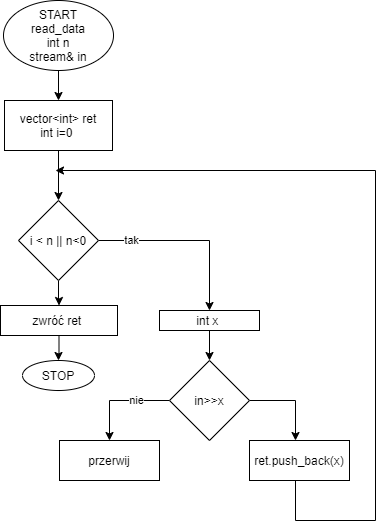
* Kolejnym elementem algorytmu jest utworzenie funkcji, która będzie wyświetlała wektor.



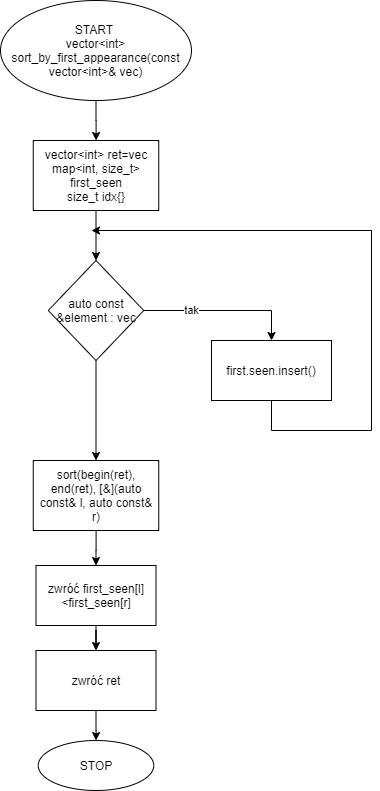
* Następną etapem, który wykona program będzie wczytanie funkcji testującej.



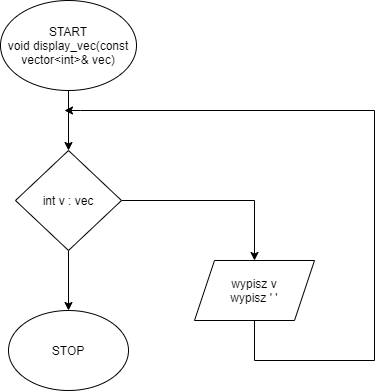
# Schemat blokowy



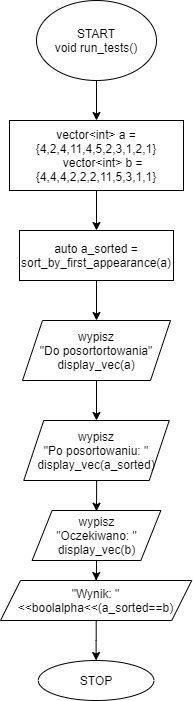
Rysunek 1. Schemat blokowy funkcji read\_data



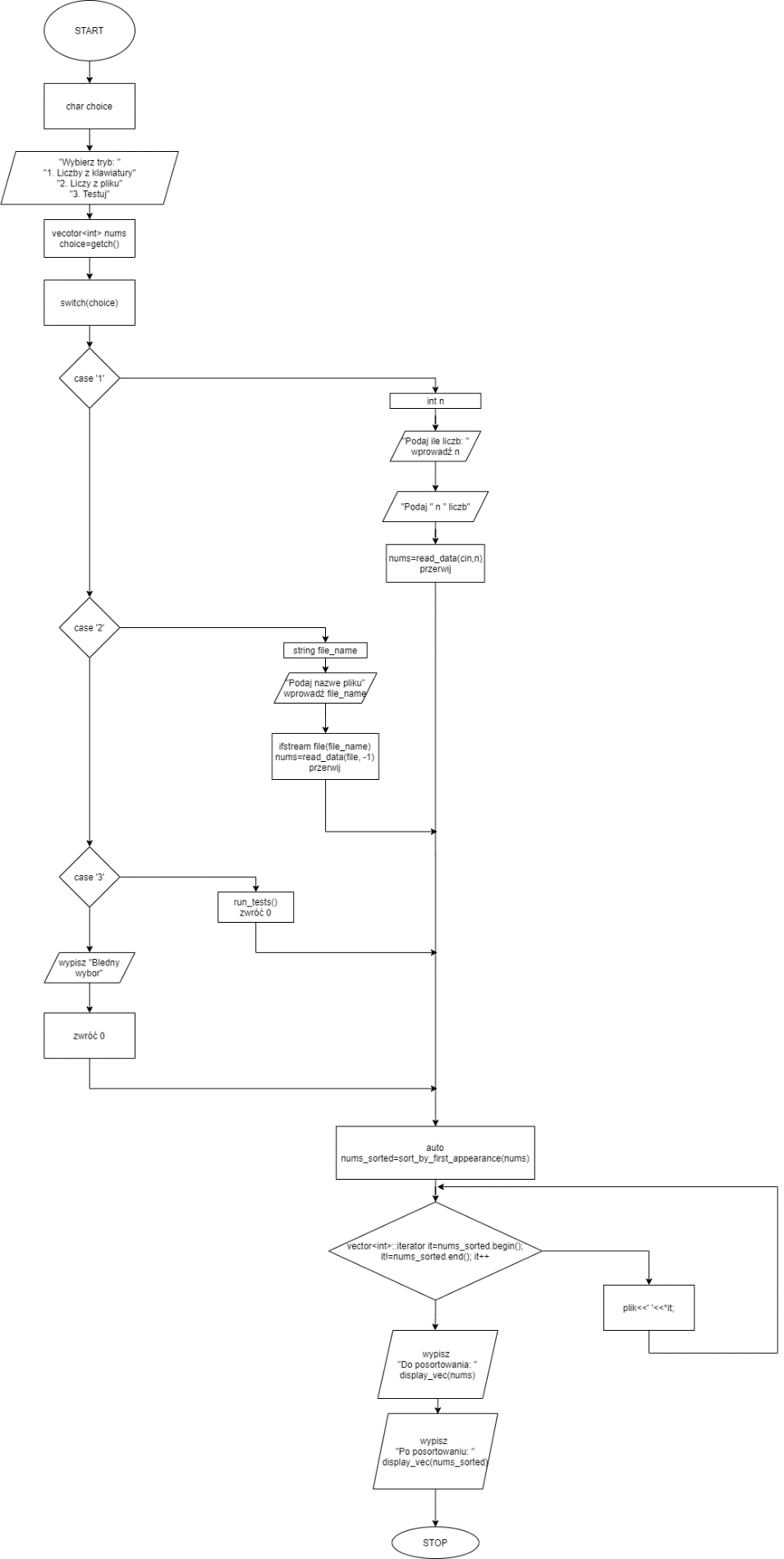
Rysunek 2. Schemat blokowy funkcji sort\_by\_rist\_appearance



Rysunek 3. Schemat blokowy funkcji display\_vec



Rysunek 4. Schemat blokowy funkcji run\_tests



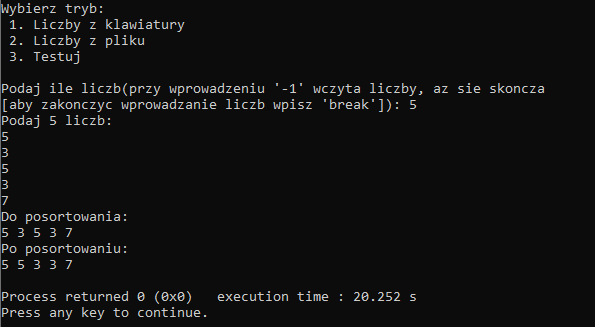
Rysunek 5. Schemat blokowy głównej funkcji main

# Dokumentacja z doświadczeń

Przeprowadziłem sześć testów dla kolejno: liczb dodatnich, ujemnych oraz mieszanych dla wejścia z klawiatury jak i z pliku.

Ad. Test 1

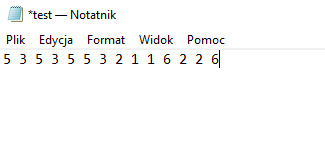
Program pobrał dane z klawiatury(liczby dodatnie), po czym posortował te liczby w kolejności ich pierwszego wystąpienia.



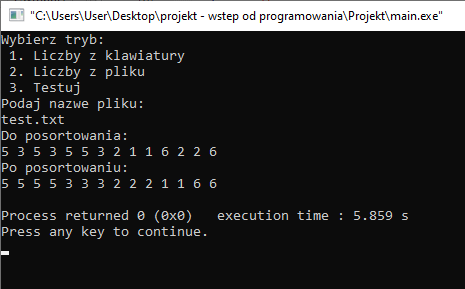
Rysunek 6. Test na liczbach dodatnich podanych z klawiatury

Ad. Test 2

Program pobrał dane z pliku tekstowego test.txt(liczby dodatnie), po czy posortował je w kolejności ich pierwszego wystąpienia.



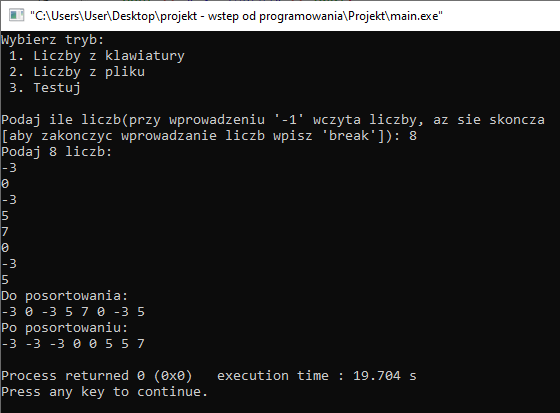
Rysunek 7. Liczy w notatniku które zostaną przetestowane



Rysunek 8.. Wynik testu

Ad. Test 3

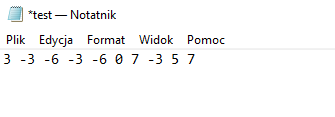
Program pobrał dane z klawiatury(liczby dodatnie jak i ujemne), po czym posortował te liczby w kolejności ich pierwszego wystąpienia.



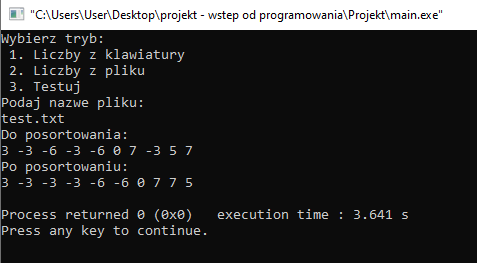
Rysunek 9. Test na liczbach dodanich jak i ujemnych

Ad. Test 4

Program pobrał dane z pliku tekstowego test.txt(liczby dodatnie jak i ujemne), po czy posortował je w kolejności ich pierwszego wystąpienia.



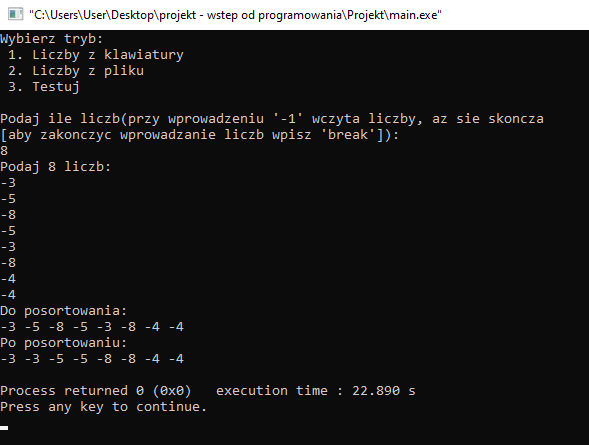
Rysunek 10.



Rysunek 11.

Ad. Test 5

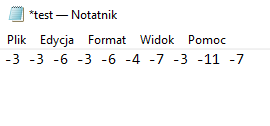
Program pobrał dane z klawiatury(liczby ujemne), po czy posortował je w kolejności ich pierwszego wystąpienia.



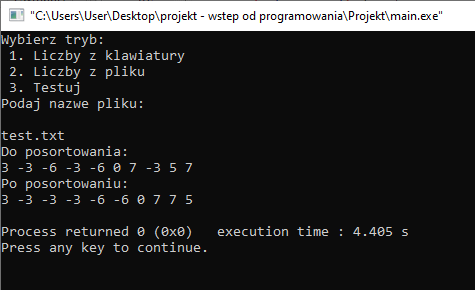
Rysunek 12.

Ad. Test 6

Program pobrał dane z pliku tekstowego test.txt(liczby ujemne), po czy posortował je w kolejności ich pierwszego wystąpienia.



Rysunek 13.



Rysunek 14.

# Wnioski

Projekt został zrealizowany, działanie algorytmu jest poprawne dla liczb dodatnich, mieszanych jak i ujemnych. W programie główny algorytm został zaimplementowany w osobnej funkcji, która jest wywoływana w późniejszych etapach działania programu. Projekt posiada możliwość odczytu z pliku tekstowego. Testy uwidocznione są za pomocą zrzutów ekranu. Najtrudniejszym zadaniem w pisaniu tego programu było nauczenie się nowych bibliotek i wykorzystanie ich na potrzeby projektu.