Sprawozdanie

Projekt na przedmiot

„Algorytmy i struktury Danych”

Temat

„Dla zadanego ciągu liczb całkowitych (tablicy) znajdź najdłuższy podciąg o zadanej sumie”

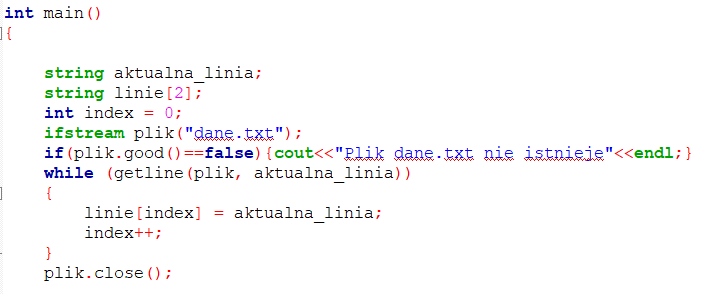
Wiktor Misiaszek, PRz semestr 2020/2021,

Inżynieria i Analiza

1.Wstęp

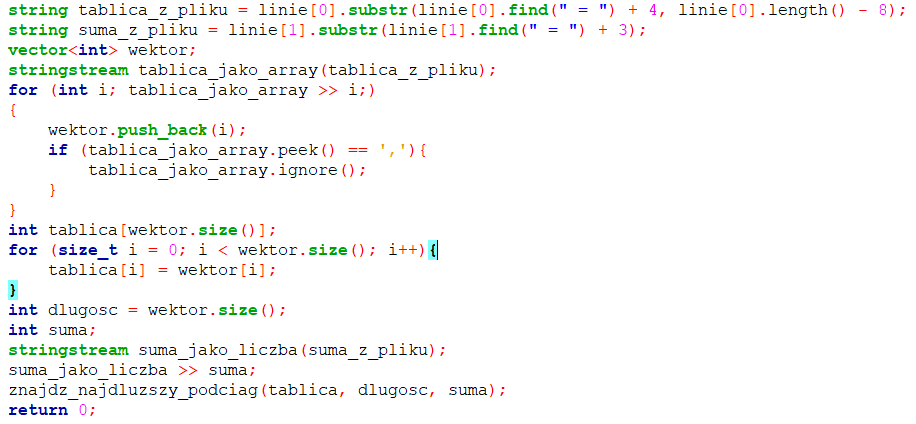
Sprawozdanie dotyczy projektu z przedmiotu Algorytmy i Struktury Danych. Celem projektu było zaimplementowanie algorytmu, który dla zadanego ciągu, wypisuje najdłuższy podciąg o podanej sumie. Następnie wyniki wpisuje do pliku tekstowego.

2.Opis programu



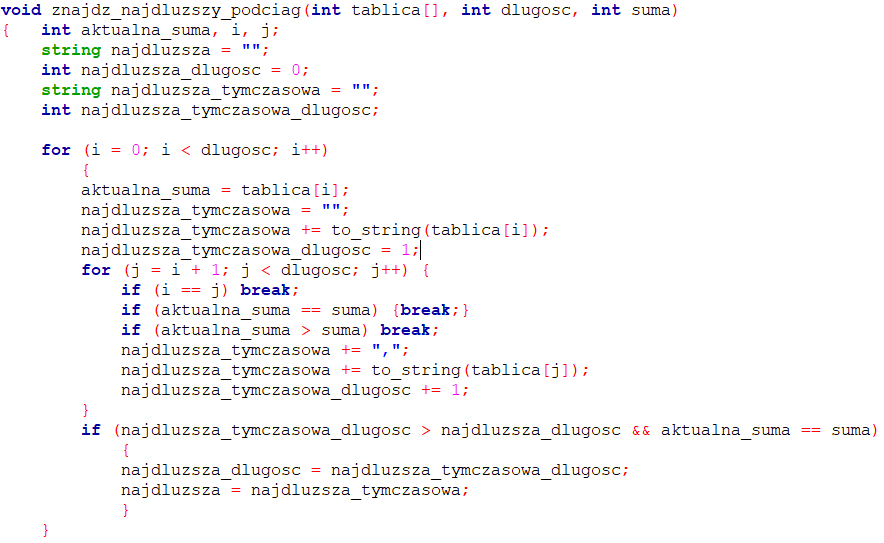
Rysunek Część pierwsza programu, Funkcja main

Program rozpoczyna prace od pobrania danych z pliku tekstowego „dane.txt”. W pierwszej kolejności sprawdza czy plik istnieje, jeśli nie to wypisuje informacje zwrotną, że plik nie istnieje. Jeśli plik istnieje pobiera z niego dwie linie. Następnie plik zostaje zamknięty.



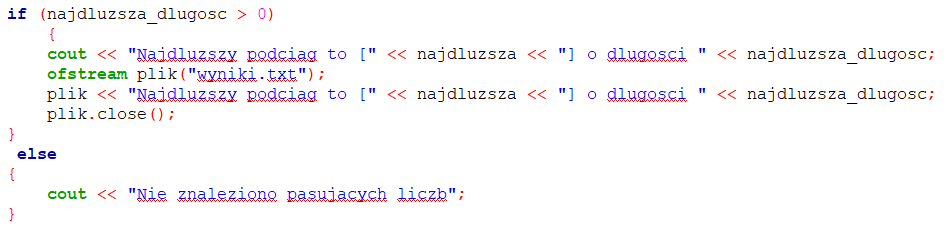
Rysunek Część pierwsza programu, Funkcja main

Program bierze pierwszą linie, którą pobrał z pliku i wyciąga z niej samą tablice liczb. Pozbywa się zatem „A[] =” oraz nawiasów kwadratowych. Następnie bierze drugą linie i pozbywa się z niej napisu „Suma =” zostawiając samą wartość sumy po czym wykonuje parsowanie tablicy czyli pozbywa się przecinków i zamienia dane, które były napisami na liczby. Na końcu maina wywołujemy funkcje szukającą najdłuższego podciągu.



Rysunek 3 część druga programu, Funkcja szukająca

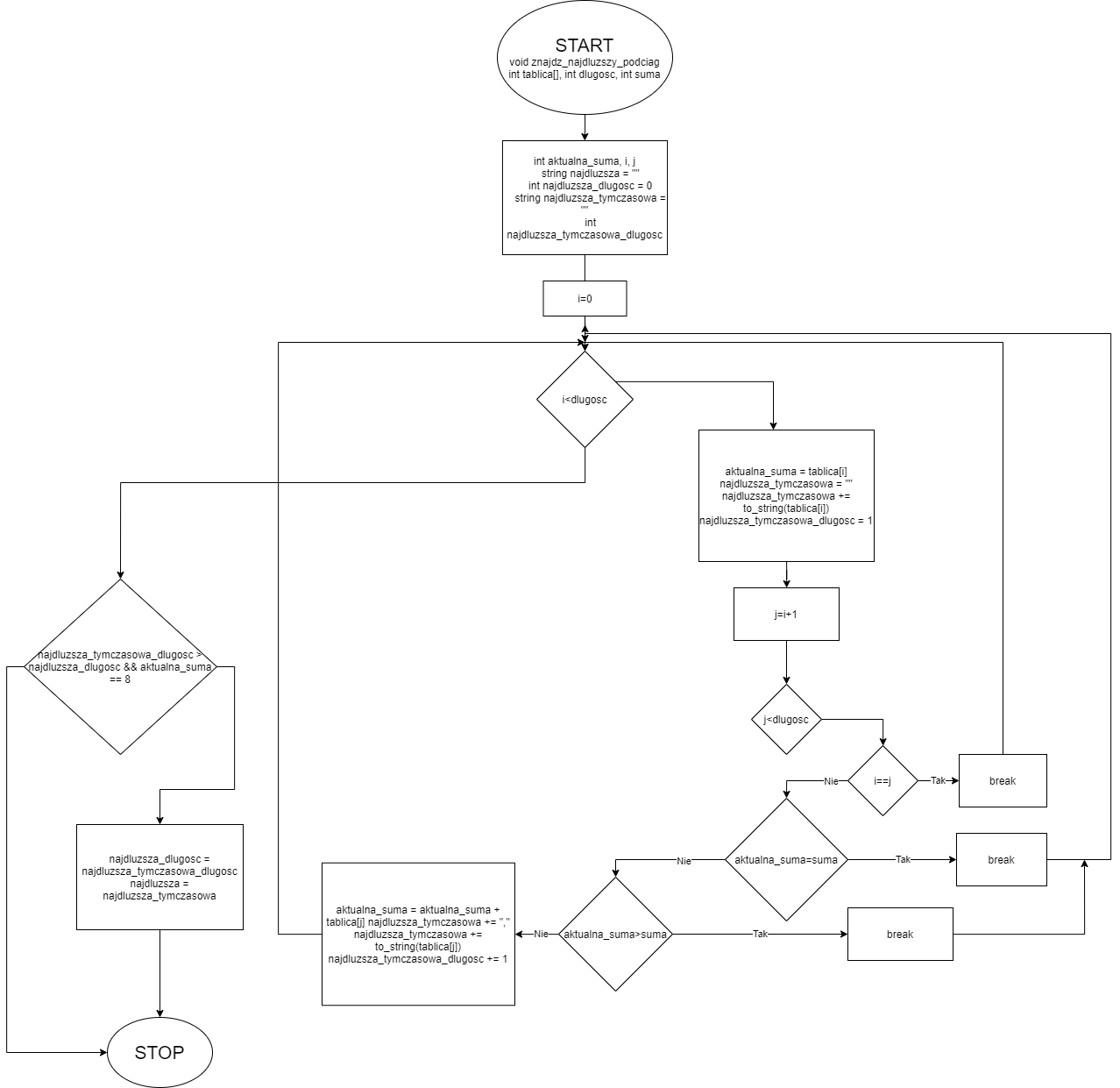
Funkcja „znajdz\_najdluzszy\_podciag” szuka najdłuższego podciągu o podanej sumie. Po znalezieniu algorytm drukuje wynik, jeśli nie znajdzie podciągu o pasującej sumie wyświetla informacje o braku wyniku. Działa w następujący sposób. Pierwsza pętla for wykonuje się tyle razu ile mamy elementów w naszym ciągu. Ustawia aktualny element jako aktualną sumę i dodaje do kolekcji tymczasowej, która przechowa nam aktualny zbiór. Po wybraniu jednego elementu z tablicy druga funkcja for przeszukuje resztę tablicy w poszukiwaniu najdłuższej sekwencji elementów pasujących do sumy. Jeśli obie liczby to ten sam element funkcja wychodzi z pętli. Jeśli aktualna suma jest równa oczekiwanej to nasza funkcja znalazła wynik i przerywa działanie pętli. Jeśli aktualna suma jest większa od oczekiwanej lub wyszliśmy poza długość tablicy funkcja przerywa działanie pętli. W przeciwnym razie dodaje do aktualnej sumy aktualny element i robi kolejny obrót pętli by znów sprawdzić aktualną sumę. Program podstawia tymczasową tablice do naszej końcowej wynikowej, jeśli jest dłuższa.



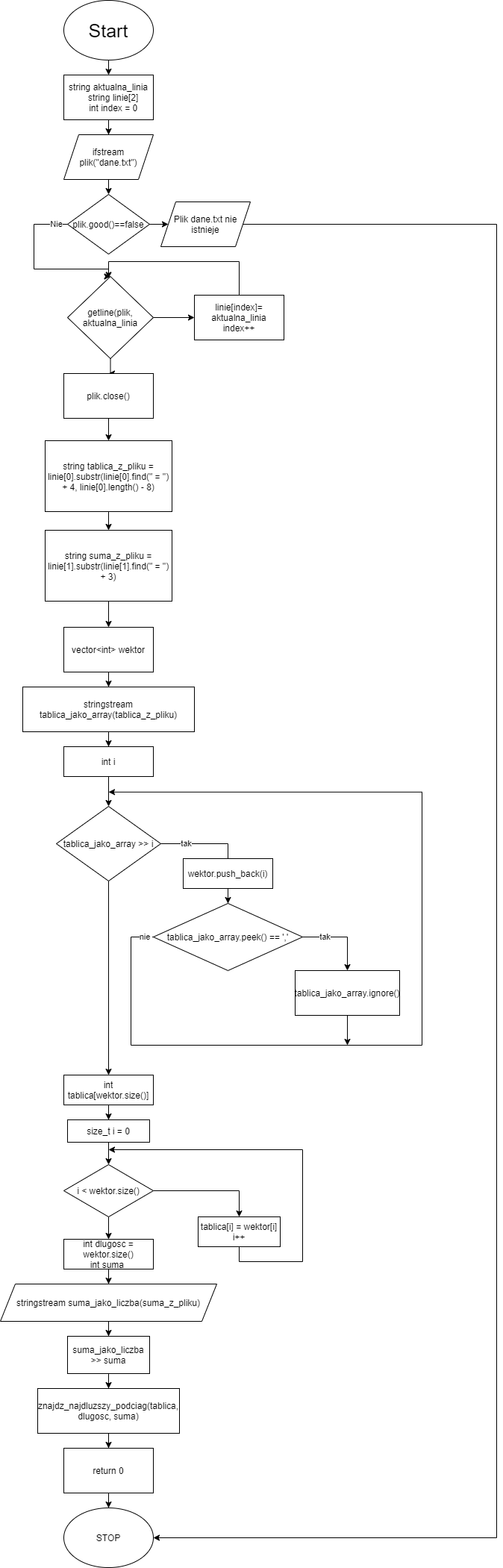
Rysunek 4 Część druga programu, Funkcja szukająca

Po znalezieniu najdłuższego podciągu program wpisuje go do pliku tekstowego „wynik.txt”. Jeśli suma była za duża, lub ciąg był za krótki funkcja wypisze że nie znaleziono pasujących liczb.

3.Schematy blokowe



Rysunek 5 Schemat blokowy Funkcja szukająca



Rysunek Schemat blokowy Funkcja main

4.Pseudokod

Funckja znajdz\_najdluzszy\_podciag (tablica, dlugosc, suma)

najdluzsza ← „”

najdluzsza\_dlugosc ← 0\

najdluzsza\_tymczasowa ← „”

dla i = 0 do i < dlugosc wykonuj

aktualna\_suma = tablica[i]

najdluzsza\_tymczasowa = ””

najdluzsza\_tymczasowa = tablica[i] + najdluzsza\_tymczasowa

najdluzsza\_tymczasowa\_dlugosc = 1

dla j = i + 1 do j < dlugosc wykonuj

Jeżeli i = j

przerwij pętle

Jeżeli atulana\_suma = suma

przerwij pętle

Jeżeli aktualna\_suma > suma

przerwij pętle

W przeciwnym razie

aktualna\_suma = aktualna\_suma + tablicja[j]

najdluzsza\_tymczaswa += „,”

najdluzsza\_tymczasowa += tablicja[j]

najdluzsza\_tymczasowa\_dlugosc += 1

Jeżeli najdluzsza\_tymczasowa\_dlugosc > najdluzsza\_dlugosc i aktualna\_suma = suma

najdluzsza\_dlugosc = najdluzsza\_tymczasowa\_dlugosc

najdluzsza = najdluzsza\_tymczasowa

Jeżeli najdluzsza\_dlugosc > 0

Wypisz „najdluzszy podciąg to [„ najdluzsza „] o dlugosci” najdluzsza\_dlugosc

Otworz plik wyniki”txt”

Zapisz do pliku

Zamknij plik

Int main ()

{

index ← 0

Otwórz plik”dane.txt”

Jeżeli nie istnieje

Wypisz na ekranie „plik dane.txt nie istnieje”

W przeciwnym przypadku

Dopóki (koniec pliku)

Wczytaj linia[index]=aktualna\_linia

zwiększ index o 1

Zamknij plik

tablica\_z\_pliku ← Usuń „A[]=” oraz nawiasy z linii o indexie=0

suma\_z\_pliku ← Usuń „Suma=” z linii o indexie=1

tablica\_jako\_array ← tablica\_z\_pliku

Dopóki z niepoprawnie nazwanej zmiennej tablica\_jako\_array da się wczytać int

Dodaj element do wektor

Jeśli tablica\_jako\_array = ”,” to ignoruj element

Dla i = 0 do i < rozmiar wektora wykonuj

tablica[i] = wektor[i]

dlugosc=wektor.size

suma\_jako\_liczba ← suma\_z\_pliku

suma ← suma\_jako\_liczba

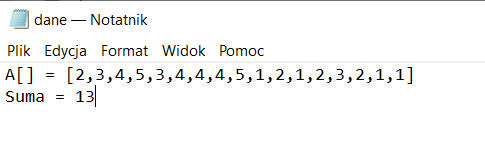
znajdz\_najdluzszy\_podciag (tablica, dlugosc, suma)

5.Dokumentacja z doświadczeń

Przeprowadzam dwa testy dla liczb dodatnich, i mieszanych. Testy zostały wykonane dla ciągów o różnych długościach i różnych sumach.

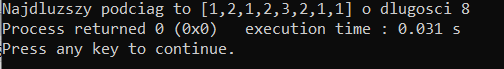
Test 1.

Podany ciąg A[] = [-2,3,4,5,1,-1,-1,3,-4,4,-4,4,4,5,1,2,1,2,3,2,1]. Podana Suma = 13.



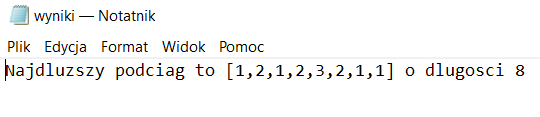
Rysunek Test 1, Dane wejściowe

Algorytm po przeprowadzeniu operacji zwraca nam w konsoli najdłuższy podciąg w tym przypadku jest to „[1,2,1,2,3,2,1,1] o długości 8”



Rysunek Test 1, Dane wyjściowe cmd

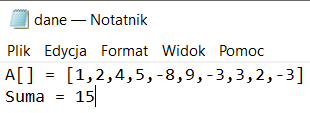
Nasz wynik zostaje również zapisany do pliku „wynik.txt”, który program sam stworzył.



Rysunek 9 Test 1, Dane wyjściowe wynik.txt

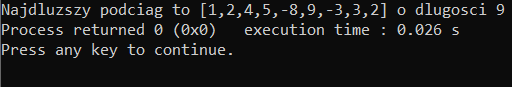
Test 2.

Podany ciąg A[] = [1,2,4,5,-8,9,-3,3,2,-3]. Podana Suma = 15.



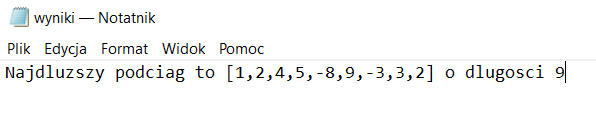
Rysunek 10 Test 2, Dane wejściowe

Program wypisuje w konsoli podciąg „[1,2,4,5,-8,9,-3,3,2] o długości 9”



Rysunek 11 Test 2, Dane wyjściowe cmd

Wyniki zostały wpisane do pliku tekstowego wyniki.txt



Rysunek 12 Test 2, Dane wyjściowe wynik.txt

6. Wnioski

Projekt został zrealizowany, działanie algorytmu jest poprawne dla liczb dodatnich jak i ujemnych. W programie główny algorytm został zaimplementowany w osobnej funkcji, która jest wywoływana w późniejszych etapach działania programu. Program odczytuje dane z pliku a następnie zapisuje je do innego. Testy uwidocznione za pomocą zrzutów ekranu w „Dokumentacji z doświadczenia” udowadniają poprawne działanie algorytmu. W programie jak i w sprawozdaniu zostały umieszczone opisy, które pozwalają na lepsze zrozumienie działania kodu. Najcięższe w tym zadaniu było napisanie funkcji sortującej tablice, równie trudne okazało się napisanie pseudokodu, ponieważ robiłem to po praz pierwszy.