# SPRAWOZDANIE

# Politechnika Rzeszowska im. Ignacego Łukasiewicza

# Wydział Matematyki i Fizyki Stosowanej

# Inżynieria i Analiza Danych

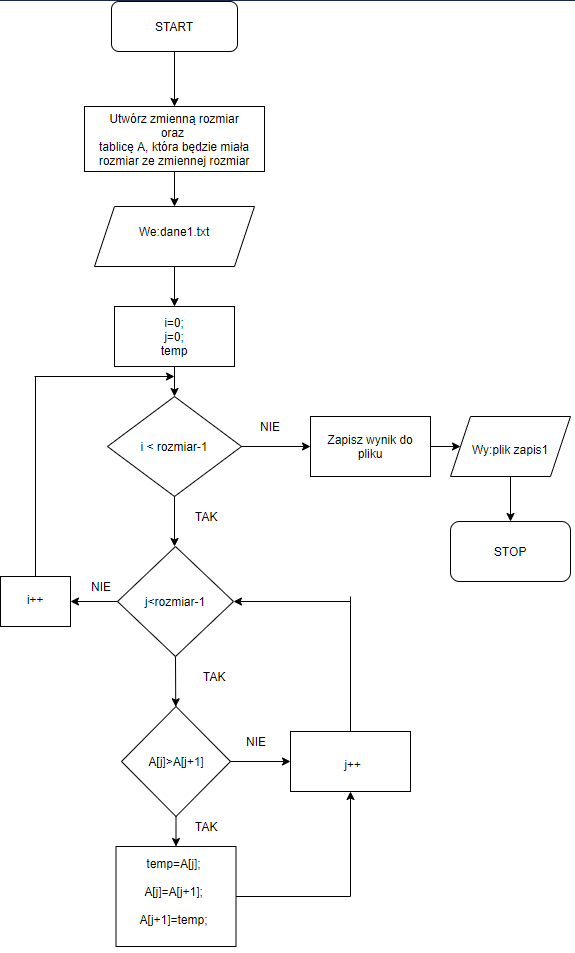
# Algorytmy i Struktury Danych

# Jakub Witkowski

Napisać program, który uporządkuje tablicę wielu powtarzających się elementów, zachowując kolejność ich pierwszego wystąpienia.

Program polega na zadeklarowaniu nowej tablicy, wpisanie do niej danych, które wczytywane są z pliku tekstowego, następnie dane te zostają posortowane przez algorytm sortujący. W programie zostało użyte sortowanie bąbelkowe, które jest wystarczająco efektywne dla plików o małej ilości danych. Po wykonaniu algorytmu, posortowane dane zostają zapisane do nowego pliku tekstowego.

Schemat blokowy algorytmu:



Pseudokod:

Utwórz stałą zmienną rozmiar,

Utwórz tablicę A[rozmiar]

We: dane z pliku dane1.txt

Utwórz zmienne: i=0; j=0; temp;

Dla i<rozmiar-1, i++

Dla j<rozmiar-1, j++

Jeśli A[j]>A[j+1], to

{

temp=A[j];

A[j]=A[j+1];

A[j+1]=temp;

}

Wpw {

j++

}

i++

Wpw

Zapisz wynik do pliku zapis 1

Wy: plik zapis1

We/Wy:Wejścia/Wyjścia

Wpw- w przeciwnym wypadku

Rezultaty testów:

Po wykonaniu programu algorytm prawidłowo posortował dane z pliku oraz zapisał je do nowego pliku tekstowego.

Kod programu:

#include <iostream>

#include <iomanip>

#include <fstream>

#include <cstdlib>

using namespace std;

const int rozmiar=10; //rozmiar tablicy

int A[rozmiar]={0}; //deklaracja tablicy

//odczytywanie danych z pliku 1

void odczyt1()

{

int x, i=0;

ifstream fin; //ifstream otwiera plik

cout << "Liczby z pliku pierwszego:" << endl;

fin.open("dane1.txt", ios::in);

while (fin>>x)

{

A[i++] = x;

cout << setw(5) << x;

}

cout<<endl;

fin.close(); //ifstream zamyka plik

}

//zapisywanie danych do pliku

void zapis1() //funkcja zapisywania do pliku

{

int x, i=0;

ofstream fout; // ofstream otwiera plik

fout.open("wyniki1.txt", ios::out);

for (i=0; i<rozmiar; i++)

{

fout<<A[i];

cout<<setw(5)<<A[i];

}

fout.close(); //ofstream zamyka plik

}

//odczytywanie danych z pliku 2

void odczyt2()

{

int x, i=0;

ifstream fin; //ifstream otwiera plik

cout << "\nLiczby z pliku drugiego:" << endl;

fin.open("dane2.txt", ios::in);

while (fin>>x)

{

A[i++] = x;

cout << setw(5) << x;

}

cout<<endl;

fin.close(); //ifstream zamyka plik

}

//zapisywanie danych do pliku

void zapis2() //funkcja zapisywania do pliku

{

int x, i=0;

ofstream fout; // ofstream otwiera plik

fout.open("wyniki2.txt", ios::out);

for (i=0; i<rozmiar; i++)

{

fout<<A[i];

cout<<setw(5)<<A[i];

}

fout.close(); //ofstream zamyka plik

}

void sortowanie\_babelkowe() //porównuje dwa kolejne elementy oraz zamienia ich kolejność,

// jeśli kolejna liczba jest mniejsza od poprzedniej

{

int temp;

for (int i=0; i<rozmiar-1; i++)

for (int j=0; j<rozmiar-1; j++)

{

if (A[j]>A[j+1])

{

temp=A[j];

A[j]=A[j+1];

A[j+1]=temp;

}

}

}

int main()

{

odczyt1(); //wywoływanie funkcji odczytu pliku

sortowanie\_babelkowe();

cout <<"\nPosortowane liczby z pliku pierwszego:" << endl;

zapis1(); //zapis wyniku do pliku

odczyt2();//wywoływanie funkcji odczytu pliku

sortowanie\_babelkowe();

cout<<"\nPosortowane liczby z pliku drugiego:"<<endl;

zapis2(); //zapis wyniku do pliku

return 0;

}