RZESZÓW, 27.11.2020

Projekt Algorytmy i struktury danych

Program wykonujący określone zadanie

Inżynieria i Analiza danych, P03

Numer indeksu 166646

Agnieszka Gabryś

1. Wstęp

Celem projektu jest implementacja kodu, to znaczy stworzenie algorytmu, który ma na celu wykonanie poszczególnych operacji, co doprowadzi nas do ostatecznego rozwiązania problemu. Proces pisania kodu zaczyna się od przygotowania teoretycznego w postaci schematu blokowego, aby następnie przejść do zapisania programu w środowisku, jakim jest „Codebloks". „Codebloks" jest darmowym ciągle rozwijającym się środowiskiem do tworzenia programów w języku C++. Pozwala on również bez przeszkód kompilować i otwierać programy stworzone w innych środowiskach programistycznych.

2. Opis problemu i treść zadania

Celem programu jest zapisanie przez algorytm, zaimplementowany w oddzielnej sekcji, liczb ujemnych na koniec tablicy. Treść zadania brzmi:

Dla zadanej tablicy liczb całkowitych przesuń wszystkie elementy mniejsze od 0 na jej koniec. Należy jednak przy tym pamiętać o zachowaniu kolejności występowania poszczególnych liczb wpisanych do programu. Wejście: A[] = |-10,5,8, -4,1,3,0, -7| Wyjście:|5,8,1,3,0,-10,-4,-7|

3. Pseudokod

Pseudokod jest to sposób zapisu algorytmu podobny wizualnie do kodu zapisanego w języku programowania jednak zapisany w języku prostym i możliwym do przyswojenia osobie niemającej styczności z programowaniem.

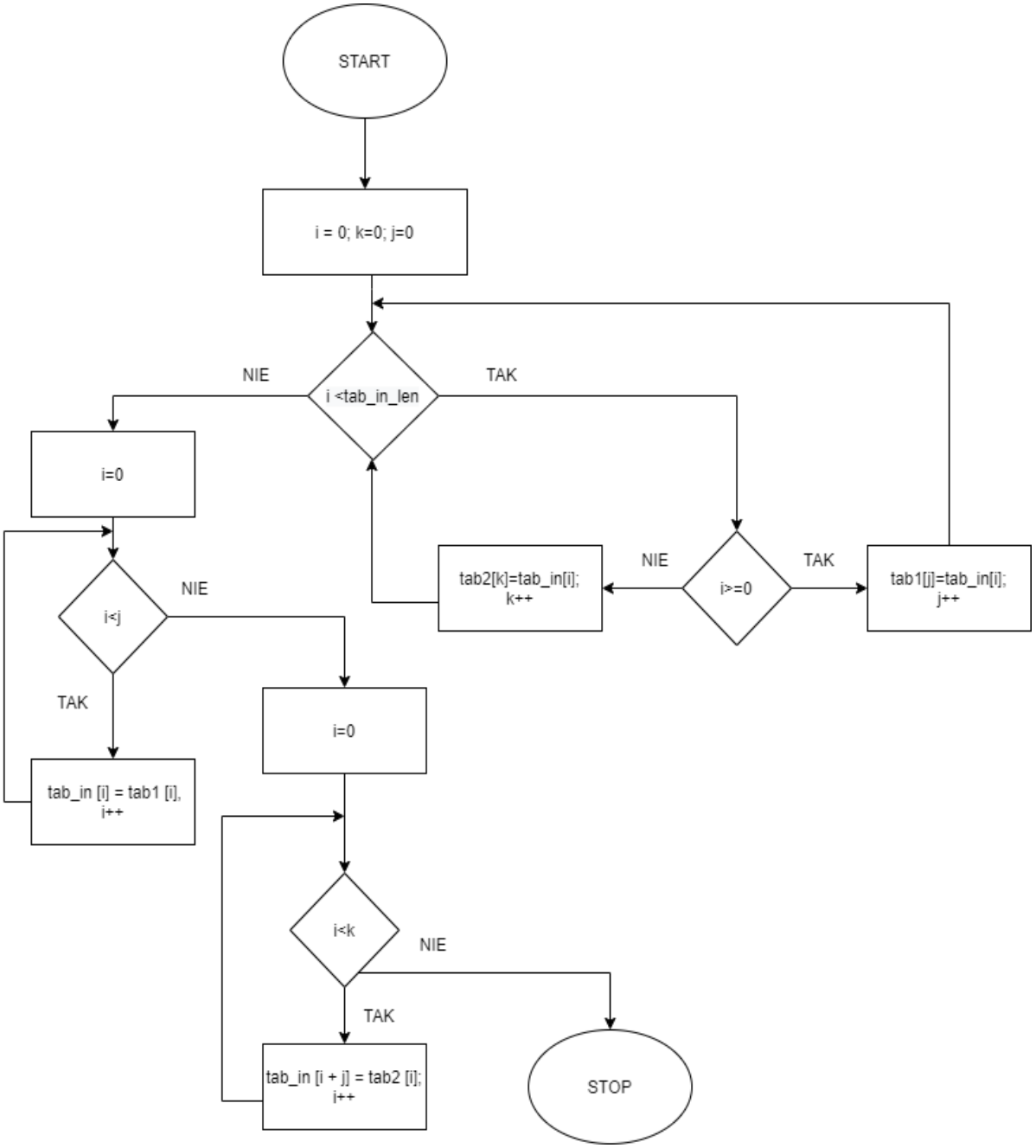
```
Ujemne_na_koniec (tab_in, tab_in_len)
Dla i= 0 dopóki i< tab_in_len wykonuj
        Jeżeli Tab_in[i]>=0 wtedy
                tab1[j] = tab_in[i]
        W przeciwnym razie
                tab2[k] = tab_in[i]
Dla i=0 dopóki i< j wykonuj
        tab_in[i] = tab1[i]
Dla i=0 dopóki i< k wykonuj
        tab_in[i+j] = tab2[i]
Usuń tab1 oraz tab2
Dla i=0 do 1<tab_in_len
        Zapisz wyniki do zapis liczb.txt
```

1 Pseudokod programu

4. Schemat blokowy

Schematem blokowym nazywamy graficzne przedstawianie kolejnych procesów przeprowadzonych w algorytmie. Schemat blokowy jest przedstawiony za pomocą figur geometrycznych, które odpowiadają za określone działania. Figury połączone są ze sobą strzałkami, które informują o kolejności wykonywanych procesów.

Schemat blokowy przedstawiony poniżej obrazuje graficzne przedstawienie algorytmu, który grupuje liczby do dwóch różnych tablic pomocniczych. Następnie zwraca je z powrotem do głównej tablicy, najpierw liczby dodatnie, a następnie ujemne.



2 Schemat blokowy

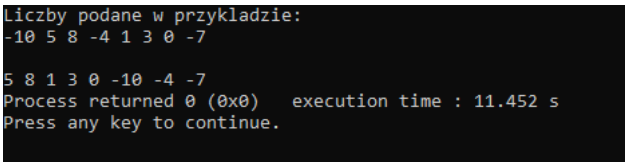
5. Kod programu

```cpp
#include <fstream>
#include <iostream>
using namespace std;
//Tworzymy tablicê dynamiczna do której wprowadzane beda liczby calkowite do posortowania
void ujemne_na_koniec(int* tab_in, int tab_in_len)
{
    int j = 0, k = 0;
    //Tworzymy 2 pomocnicze tablice
    int* tab1= new int[tab_in_len];
    int* tab2= new int[tab_in_len];
// Warunek dla którego liczby wieksze i równe 0 maja zoatac przeniesione do tab1
    for(int i=0; i<tab_in_len; i++)
    {
        if(tab_in[i]>=0)
        {
            tab1[j]=tab_in[i];
            j++;
        }
//Ujemne liczby zosta przeniesione do tab2
        else
        {
            tab2[k]=tab_in[i];
            k++;
        }
    }
//Liczby z pomocniczych tablic zoataj¹ z powrotem przeniesione do tab_in , w odpowiedniej kolejnoœci

    for(int i=0; i<j; i++)
        tab_in[i]=tab1[i];
    for(int i=0; i<k; i++)
        tab_in[i+j]=tab2[i];

//Dla pewności usuwamy tab1 i tab2 żeby nie zajmowały niepotrzebnego miejsca
    delete[] tab1;
    delete[] tab2;
// Zapisujemy wyniki do pliku zapis liczb.txt
     for(int i=0; i<tab_in_len; i++)
    {
        fstream plik;
        plik.open("zapis liczb.txt",ios::out|ios::app);
        plik<<tab_in[i]<<" ,";
        plik.close();
    }

}
int main()
{
    int n, j = 0, k = 0;
    cout << "ile liczb wypisujemy\n";
    cin >> n;
    int* tab=new int[n];
    for(int i=0; i<n; i++)
    {
        cout << "Podaj liczbe " << i+1 << ": ";
        cin >> tab[i];
    }
    cout << endl << endl;
    ujemne_na_koniec(tab,n);
    cout << "Wpisane liczby po sortowaniu:" << endl;

    for(int i=0; i<n; i++)
    {
        cout << tab[i] << ' ';
    }

    cout << endl;

    cout<< "Liczby podane w przykladzie:"<<endl;
    int tab2[]= {-10,5,8,-4,1,3,0,-7};
    for(int i=0; i<8; i++)
    {
        cout << tab2[i] << ' ';
    }
    cout<<endl<<endl;
    ujemne_na_koniec(tab2, 8);

    for(int i=0; i<8; i++)
    {
        cout << tab2[i] << ' ';

    }

    return 0;

}
```

3 Kod programu

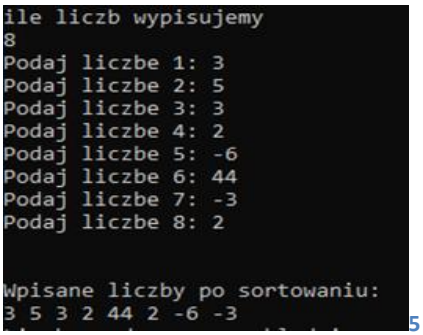
6. Działanie programu

Na początku programu tworzymy funkcje ujmne_na_koniec oraz trzy tablice dynamiczne, jedną główną, do której będą wpisywane liczby, które należy posortować, oraz dwie pomocnicze, do których będą zapisywane liczby większe lub równe zero, jak i ujemne. Deklarujemy zmienne j i k, które będą odpowiedzialne za pozycje w tablicach pomocniczych. Program następnie po rozdzieleniu liczb na nieujemne i ujemne oraz przeniesieniu ich do wyznaczonych tablic pomocniczych, ponownie zwraca je do podstawowej tablicy, która zostanie wyświetlona na końcu. Tablice pomocnicze tab1 i tab2 zostają usunięte, gdy nie są już potrzebne, żeby nie zajmowały niepotrzebnego miejsca na dysku. Po wszystkich operacjach zapisuje wyniki do pliku txt.



4 Test 1

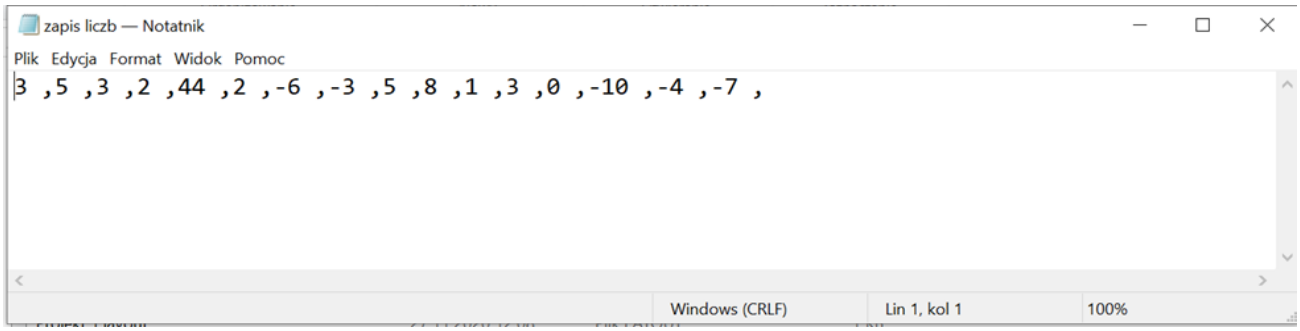
Na pokazanym powyżej przykładzie liczby są wpisane do tablicy programu zanim jeszcze program zostanie uruchomiony, ma określoną ilość miejsc zarezerwowaną na liczby. Następnie zostaje uruchomiony algorytm ujemne_na_koniec dla tab2 o ilości elementów 8. Kolejno wykonuje kroki opisane we wcześniejszym akapicie.



5 Test 2

Użytkownik ma możliwość wybrania wielkości głównej tablicy oraz liczb, które zostają posortowane. Wewnątrz programu wywoływany jest algorytm opisany w początkowym akapicie. Program wykonuje wszystkie operacje zapisane w algorytmie, który jest zaimplementowany w oddzielnej funkcji, po czym na konsoli wyświetlają się wszystkie liczby w odpowiedniej kolejności zadanej w treści głównej zadania. Natomiast do pliku txt nadpisuje się wynik kolejnej operacji, nie usuwając poprzednich przeprowadzonych testów,

tak jak jest to przedstawione w poniższym zdjęciu przedstawiającym zawartość notatnika (6)



6 plik txt, do którego wpisywane są wyniki

7. Wnioski

Program działa poprawnie, algorytm spełnia wszystkie warunki zawarte w głównej części zadania. Algorytm został zaimplementowany w odrębnej funkcji, która jest wywoływana w głównym programie. Największym napotkanym problemem było zespolenie tablic pomocniczych aby wyświetlała się tylko jedna pełna tablica. Program posiada możliwość zapisywania wyników w plikach txt, oraz użytkownik ma możliwość wprowadzenia dowolnej ilości liczb do zrealizowania założenia.