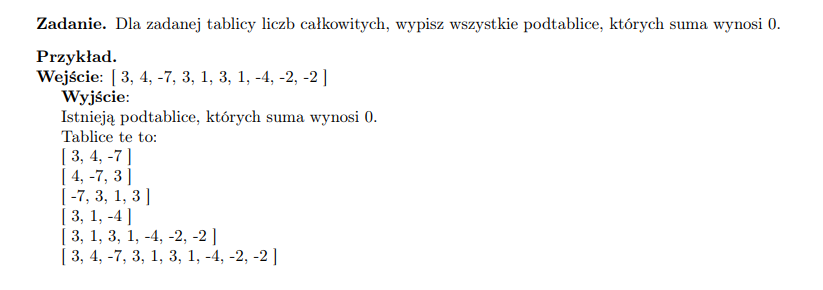
|  |
| --- |
|  |
| Sprawozdanie z projektu |
| Zadanie projektowe numer 1. |

|  |
| --- |
| Katarzyna Wojciechowska, 166714  Inżynieria i analiza danych, grupa P8 |

# Opis problemu

## Treść zadania



## Tablice w informatyce

**Tablica** – kontener uporządkowanych danych takiego samego typu, w którym poszczególne elementy dostępne są za pomocą kluczy (indeksu). Indeks najczęściej przyjmuje wartości numeryczne. Rozmiar tablicy jest albo ustalony z góry (tablice statyczne), albo może się zmieniać w trakcie wykonywania programu (tablice dynamiczne). Tablice jednowymiarowe mogą przechowywać inne tablice, dzięki czemu uzyskuje się tablice wielowymiarowe.   
W tablicach wielowymiarowych poszczególne elementy są adresowane przez ciąg indeksów.

**Tablica dynamiczna** jest tworzona w czasie uruchomienia programu. Jej rozmiar może być wyliczany. Co więcej, gdy przestanie być potrzebna możemy ją usunąć z pamięci. Dzięki tym własnościom program efektywniej wykorzystuje zasoby pamięciowe komputera.

## Krótki opis programu

Po uruchomieniu programu należy podać rozmiar tablicy dynamicznej, a następnie jej elementy. Program sprawdza czy zostały wprowadzone poprane dane i przy użyciu pętli wyszukuje podtablic, których suma elementów jest równa 0. Wyniki jego działania zostają zapisane w pliku tekstowym o nazwie zadeklarowanej na początku kodu (wyniki.txt).

# Program

## Kod

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

bool istniejeTakaTablica=false; //zmienna służąca do przechowywania informacji o tym czy dla dannej tablicy istnieją podtablice

bool wyswietlono=false; // zmienna pomocnicza pozwalające na jednorazowe wyświetlenie informacji

int rozmiar=0;

const string NazwaPliku="wyniki.txt"; //nazwa pliku do którego zapisany zostanie wynik

int\* pobierzTablice()// Funkcja służąca do stworzenia i wypełnienia tablicy

{

cout << "Ile elementow ma posiadac tablica? ";

cin >> rozmiar;

while(cin.fail() || rozmiar<=0) { // sprawdzamy czy dane są wczytane niepoprawnie, jeśli tak, wykonujemy zawartość pętli

cout << "Blad! podaj poprawną liczbe!" << endl;

cin.clear(); // czyścimy flagę błędu na cin

cin.ignore(256,'\n'); // ignorujemy wszystkie błędne dane (w tym przypadku maksymalnie 256 znaków) aż dno napotkania znaku nowej lini, czyli klawisza enter

cin >> rozmiar; // wprowadzamy wartość ponownie

}

int \*tablica = new int[rozmiar]; // deklarujemy tablicę dynamiczną o podanym rozmiarze

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

cout << "\nPodaj element nr." << i+1 <<": ";

cin >> tablica[i];//wpisujemy dane do 'i' indeksu tablicy

while(cin.fail()) {// takie samo sprawdzanie jak wyżej, tylko bez warunku że liczba musi być większa od zera

cout << "Blad! podaj liczbę!" << endl;

cin.clear();

cin.ignore(256,'\n');

cin >> tablica[i];

}

}

return tablica;

}

void znajdzPodtablice(int tab[], int n)//funkcja znajdująca podtablice i wypisująca je do pliku

{

ofstream plik;

plik.open (NazwaPliku);

plik << "\n\nGlowna tablica: [ ";

for(int i=0; i<n;i++)// wypisujemy pokolei główną tablicę

{

if(i+1==n) // jeśli kolejny indeks nie istnieje w tej tablicy, przy aktualnej wartości nie dodajemy przecinka ale ']' i łamiemy wiersz

{

plik << tab[i] <<" ]\n\n";

}

else

{

plik << tab[i] << ", ";

}

}

for (int i = 0; i < n; i++)// pętla przechodzi przez wszystkie elementy tablicy zaczynając od pierwszego (o indeksie 0)

{

int sum = 0;

for (int j = i; j < n; j++) // pętla przechodzi przez wszystkie elementy tablicy zaczynając od elementu 'i'

{

sum += tab[j];//dodajemy wartość elementu 'j' tablicy do zmiennej trzymającej sumę

if (sum == 0) {// jeśli suma jest równa zero

istniejeTakaTablica=true;//zaznaczamy że tablica spełniająca warunek istnieje

if(!wyswietlono) // jednorazowo wyświetlamy wiadomość że ta tablica ma przynajmniej jedną podtablicę której suma elementów jest równa 0

{

plik << "Istnieja podtablice, ktorych suma wynosi 0\n";

wyswietlono=true;// ustawiamy wartość zmiennej na true, co zapobiegnie wyświetleniu tej wiadomości przy znalezieniu kolejnej podtablicy

}

plik << "Podtablica: [" << i << ".." << j << "] czyli: ";// wyświetla przedział indeksów głównej tablicy w którym zawarta jest podtablica

plik << "[ ";

for(int l=i; l<=j;l++)// przechodzimy przez podany wyżej przedział i wyświetlamy każdy element

{

if(l+1>j)

{

plik << tab[l] <<" ]\n"; // jeśli kolejny indeks nie istnieje w tej tablicy, przy aktualnej wartości nie dodajemy przecinka ale ] i łamiemy wiersz

}

else

{

plik << tab[l] << ", ";

}

}

}

}

}

if(!istniejeTakaTablica) //Jeżeli nigdy nie ustawiono flagi na true oznacza to że nie istnieja podtablica o sumie elementów równej 0

{

plik << "Nie istnieja podtablice, ktorych suma wynosi 0\n";

}

plik.close(); // zamykamy plik

}

int main()

{

int \*tablica = pobierzTablice();

znajdzPodtablice(tablica, rozmiar);// przekazujemy tablicę do funkcji która znajdzie i wpisze do pliku jej podtablice

delete [] tablica; // usuwamy tablicę dynamiczną

//wypisanie na ekran

ifstream wyniki(NazwaPliku);

if (wyniki.is\_open())//jeśli plik się poprawnie otworzył

{

cout << wyniki.rdbuf(); //wypisujemy na ekran całą jego zawartość

wyniki.close(); // zamykamy plik

}

else//Jeśli nie udało się otworzyć pliku wyświetlamy błąd

{

cout << "Nastapil blad przy probie otwarcia pliku " << NazwaPliku << " przechowujacego wyniki dzialania programu!";

}

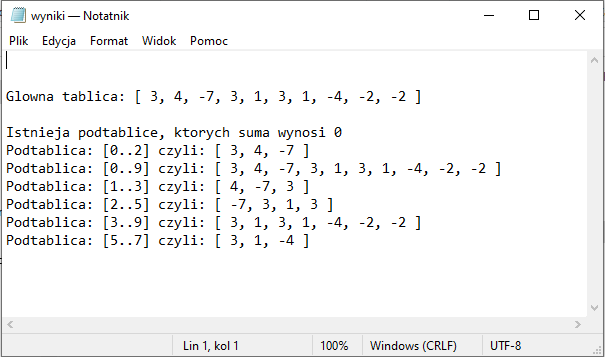
return 0;

}

## Zaprogramowane operacje

* Podawanie rozmiaru tablicy dynamicznej
* Dodawanie kolejnych elementów tablicy
* Sprawdzanie poprawności wpisywanych danych
* Tworzenie pliku tekstowego, w którym zapisywane są wyniki działania programu
* Wyszukiwanie i wypisywanie odpowiednich podtablic

## Wyniki działania programu



## Pseudokod

instniejeTakaTablica ← false – zmienna służąca do przechowywania informacji czy dla danej tablicy istnieją podtablice

wyświetlono ← false – zmienna pomocnicza; jednorazowo wyświetla informacje

rozmiar ← 0

NazwaPliku ← ”wyniki.txt”

funkcja pobierzTablice()

{

wpisujemy rozmiar tablicy

rozmiar ← wpisana wartość

dopóki (cin.fail() || rozmiar<=0) –jeśli rozmiar<=0 lub podano znak inny niż liczba

wypisz „Blad! Podaj poprawna liczbe”

cin.clear()

cin.ignore(256, ‘\n’) – czyścimy flagę błędu i ignorujemy błędne dane

ponownie wprowadzamy wartość ponownie

int \*tablica = new int[rozmiar] – deklarujemy tablicę dynamiczną

dla i=0 dopóki i<rozmiar wykonuj

Podajemy element tablicy nr. i+1

tablica[i] = podany element nr. i+1

dopóki cin.fail() –sprawdzamy czy podany znak jest liczbą

wyświetl „Blad! Podaj liczbe!”

cin.clear()

cin.ignore(256,'\n')

wprowadzamy wartość ponownie

zwróć tablicę

}

Funkcja znajdzPodtablice (int tab[], int n)

{

utwórz plik (NazwaPliku)

wpisujemy do pliku tablicę główną

dla i=0 dopóki i<n wykonuj

jeżeli i+1==n

wypisz „]” i zakończ tablicę

w przeciwnym razie

wypisz „ , ”

dla i=0 do i<n wykonuj

sum ← 0

dla j=i dopóki j<n wykonuj

sum=sum+tab[j]

jeżeli sum=0

istniejeTakaTablica=true

jednorazowo wyświetlamy wiadomość że taka podtablica istnieje

wpisz podtablice do pliku

wyświetl przedział (i,j) indeksów gł. tablicy, w którym zawarta jest podtablica

dla l=i dopóki l<=j

jeżeli (l+1>j)

wypisz „]”

w przeciwnym wypadku

wypisz „ ,”

jeżeli istniejeTakaTablica nie zostało nigdy ustawione na true

wyświetl „Nie istnieja podtablice, ktorych suma wynosi 0”

zamknij plik

}

int main ()

{

tablica ← pobierzTablice()

znajdzPodtablice(tablica, rozmiar)

delate [] tablica

wyniki(NazwaPliku)

jeżeli plik wyniki otworzył się poprawnie

wypisz na ekran jego zawartość

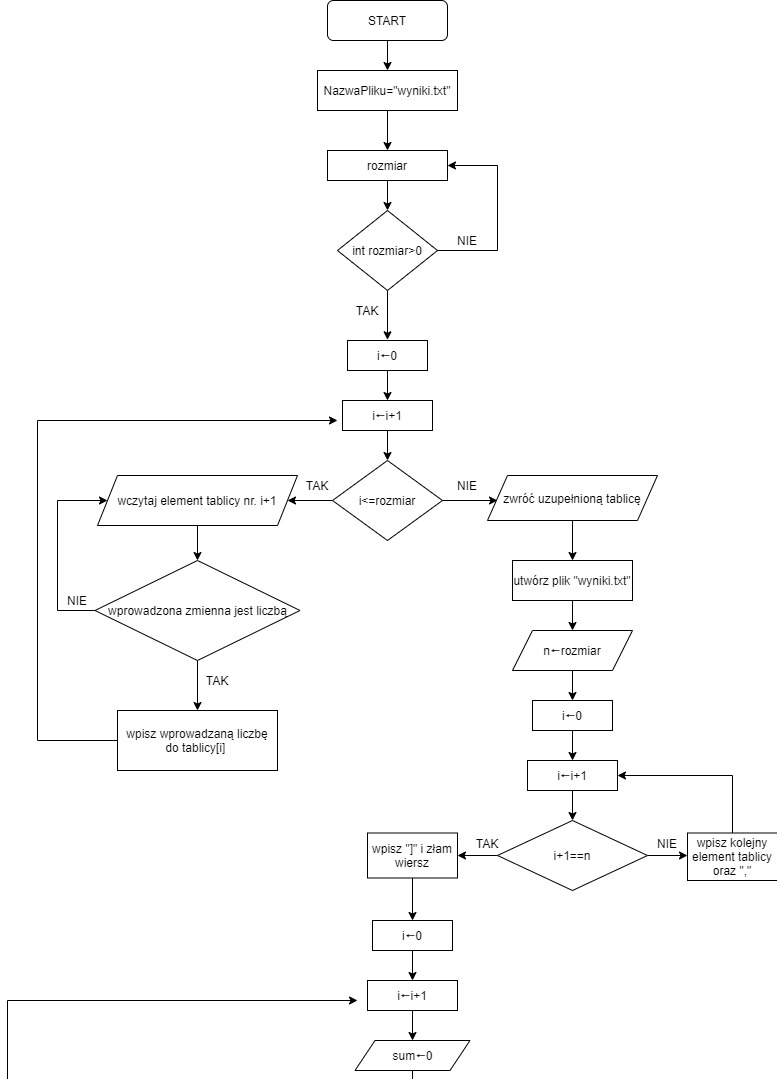
zamknij plik

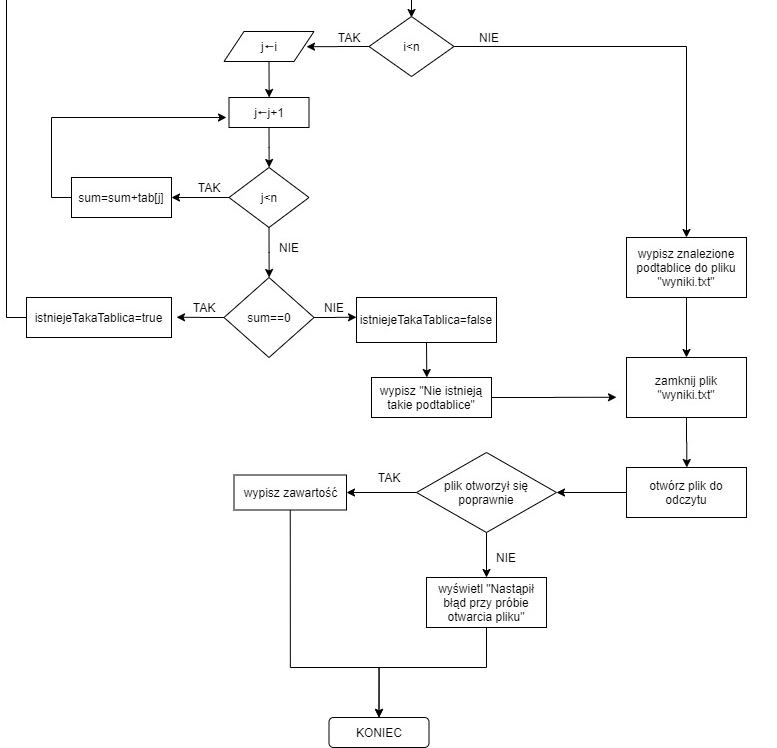
w przeciwnym przypadku

wyświetl „Nastapil blad przy probie otwarcia pliku”

}

## Schemat blokowy

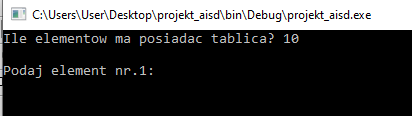




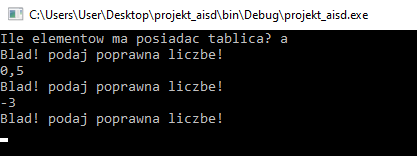
# Dokumentacja z doświadczeń

## Sprawdzanie poprawności podanego rozmiaru tablicy

Wprowadzenie poprawnej wartości (liczby całkowitej większej od 0) daje możliwość wpisania kolejnych elementów tablicy:

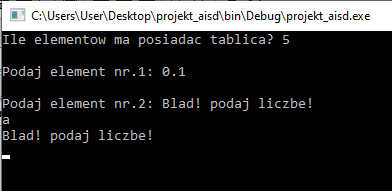


Kiedy podana zostanie liczba ujemna, liczba niecałkowita, litera bądź znak, zostaje wyświetlony błąd:

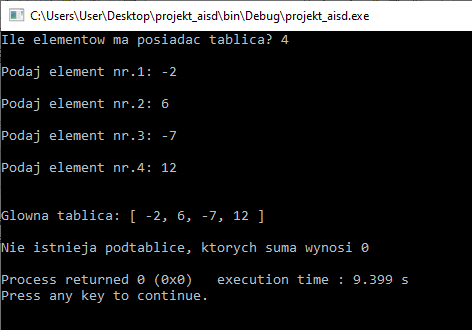


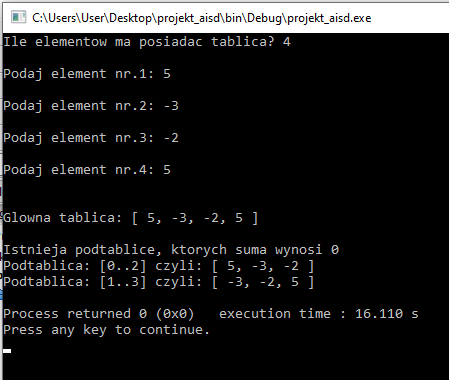
## Sprawdzanie poprawności wpisywania elementów tablicy

Wprowadzenie liczby niecałkowitej lub litery powoduje wyświetlenie błędu programu.



## Wpisywanie dowolnej tablicy złożonej z liczb całkowitych





# Wnioski

Do wykonania zadania potrzebne było użycie pętli i zdefiniowanie funkcji. W programie występuje sprawdzanie poprawności wpisywanych danych, dzięki czemu program nie psuje się przy wprowadzaniu błędnych danych, ale wyświetla informacje o błędzie i daje możliwość poprawienia wpisanych wartości. Program działa dla każdej poprawnie wpisanej tablicy.