

SPRAWOZDANIE

PROJEKT 1

TOMASZ CHOLEWIŃSKI, 169768, INŻYNIERIA I ANALIZA DANYCH, GRUPA P1

1 OPIS PROBLEMU

1.1 TREŚĆ ZADANIA

Zadanie 24. Dla zadanego zbioru liczb wypisz jego najmniej liczny podzbiór, którego suma jest większa niż suma pozostałych elementów tego zbioru.

Przykład

Wejście [2, 7, 3, 1]

Wyjście [7]

Wejście [2, 4, 1, 3, 9, 0, 3]

Wyjście [4, 9], [3,9]

Wejście [1, 1, 2]

Wyjście Brak podzbiorów spełniających zadane kryteria.

1.2 TABLICE DYNAMICZNE

Tablica dynamiczna zostaje utworzona podczas działania programu a jej rozmiar może być zmieniany. Pozwala nam to na optymalne wykorzystanie zasobów pamięci komputera. Gdy tablica nie jest już potrzebna należy ją usunąć by zwolnić miejsce.

1.3 OPIS PROGRAMU

Program po uruchomieniu prosi o podanie rozmiaru tablicy, następnie użytkownik podaje jej elementy. Program po sprawdzeniu czy podane zostały prawidłowe dane sortuje tablice w kolejności malejącej a następnie szuka najmniejszego podzbioru którego suma jest większa od sumy pozostałych elementów. Dokonuje tego sprawdzając liczby po kolei zaczynając od największej czy spełnia założenie. Jeżeli takowy podzbiór istnieje wyświetla go na ekranie i zapisuje do pliku. Jeśli nie wyświetla komunikat o braku podzbiorów spełniających zadane kryteria.

2 PROGRAM

2.1 KOD

```
#include <iostream>
```

```
#include<algorithm>
```

```
#include<fstream>
```

```
using namespace std;
```

```
int suma, i, j, rozmiar;
```

```

//malejaco
bool porownaj(int a, int b)
{
    return a>b;
}

int *tworzenie() //tworzymy i wypelniamy tablice
{
    cout << "Podaj rozmiar tablicy: ";
    cin >> rozmiar;

    while(cin.fail() || rozmiar <= 1) //sprawdzamy poprawnosc wpisywanych danych
    {
        cout << "Niepoprawna wartosc" << endl;
        cin.clear();
        cin.ignore(50, '\n');
        cin >> rozmiar;
    }

    int* tab = new int[rozmiar]; //tablica dynamiczna o rozmiarze rozmiar

    for(int i = 0; i < rozmiar; i++) //uzytkownik uzupełnia dane
    {
        cout << "Podaj "<<i+1<<" element tablicy" <<endl;
        cin >> tab[i];
        while(cin.fail()) //sprawdzamy poprawnosc wpisywanych danych
        {
            cout << "Zla wartosc, wprowadz liczbe!"<<endl;
            cin.clear();
            cin.ignore(50, '\n');
            cin >> tab[i];
        }
    }
}

```

```

    }

    sort(tab, tab + rozmiar, porownaj); //sortujemy tablice w kolejnosci malejacej

    suma = 0;
    for(int i=0;i<rozmiar;i++)
        suma=tab[i]+suma;

    return tab;

}

void podZbior(int tab[]) //zadanie
{
    ofstream wynik("wynik.txt");

    int pom = tab[0]; // funkcja pomocnicza, przypisujemy jej największą wartość z tablicy
    while (pom<suma && j<rozmiar && pom!=suma-pom ) //sprawdzamy czy tablica spelnia warunki
        zadania
    {
        if(pom > suma-pom) //warunek zostaje spelniony
        {
            cout << "Najmniejszy podzbior sklada sie z "<<j+1<<" element/ow: ";
            cout << "["<<tab[0];
            wynik<<tab[0];
            for (i=1;i<=j;i++)
            {
                cout << ", "<< tab[i];
                wynik<< ", "<<tab[i] ; //wypisanie wyniku do pliku
            }
        }
    }
}

```

```

    }
    cout << "]\n";
    wynik.close();
    goto poprawnie;
}
else
{
    pom = pom+tab[i];
    j++;
}

}
cout << "Brak podzbiorow spelniajacych zadane kryteria";

```

poprawnie:

```

cout<<endl;
cout<<"Koniec zadania"<<endl;
}

```

```

int main()
{
    int* tablica = tworzenie();
    podZbior(tablica);
    delete [] tablica;

    return 0;
}

```

2.2 ZAPROGRAMOWANE OPERACJE

- Podanie rozmiaru tablicy dynamicznej
- Dodanie kolejnych elementów tablicy

- Sprawdzenie poprawności wprowadzanych danych
- Wyszukanie i wyświetlenie podzbiorów
- Wypisanie wyniku programu do pliku tekstowego

2.3 WYNIK DZIAŁANIA PROGRAMU

```
Podaj rozmiar tablicy: 4
Podaj 1 element tablicy
2
Podaj 2 element tablicy
1
Podaj 3 element tablicy
3
Podaj 4 element tablicy
7
Najmniejszy podzbiór składa się z 1 element/ow: [7]
Koniec zadania
```

Rysunek 1 Działanie programu w konsoli



Rysunek 2 wynik działania w dokumencie

2.4 PSEUDOKOD

Porównaj – odwróć kolejność tablicy

```
{
  Zwróć a > b
}
```

Funkcja tworzenie()

```
{
```

Wyświetl Podaj rozmiar tablicy

rozmiar – podana wielkość

dopóki (cin.fail lub rozmiar<1) jeśli podano znak lub liczbę mniejszą od 1

```

wyświetl niepoprawna wartość
cin.clear()
wyczyść flagę błędu i ignoruj błędne dane
ponownie wprowadzamy wartość
twórz tablice o rozmiarze podanym przez użytkownika
i = 0 i wykonuj dopóki i < rozmiar
{
    wyświetl podaj i+1 element tablicy
    wpisz liczbę do tab[i]
dopóki cin.fail – użytkownik podaje coś innego niż liczba
{
    wyświetl niepoprawna wartość
    cin.clear()
    ignoruj znaki
    wprowadź wartość do tablicy
}
Dodaj 1 do i
}
Sortuj tablice w kolejności malejącej
Suma = 0
I = 0 dopóki i < rozmiar
    Suma = tab[i] + suma
    dodaj 1 do i
Zwróć tablice

Funkcja podZbior(od int tab[])
    Utwórz plik tekstowy wynik.txt
Zmienna całkowita pom = tab[0]

Dopóki (pom<sumy i j<rozmiaru i pom!= suma-pom)
{

```

```

Jeżeli (pom > suma-pom)
{
    Wyświetl „Najmniejszy podzbior składa się z j+1 element/ow:”
    Wyświetl „ [ tab[0]”
    Zapisz tab[0] do pliku wynik
    l=1 dopóki i<=j
    {
        Wyświetl „, tab[i]”
        Zapisz „, tab[i]” do pliku wynik.txt
    }
    Wypisz „ ]”
    zamknij plik wynik.txt
    idź do poprawnie
}

```

Wyświetl „Brak podzbiorów spełniających zadane kryteria”

Poprawne:

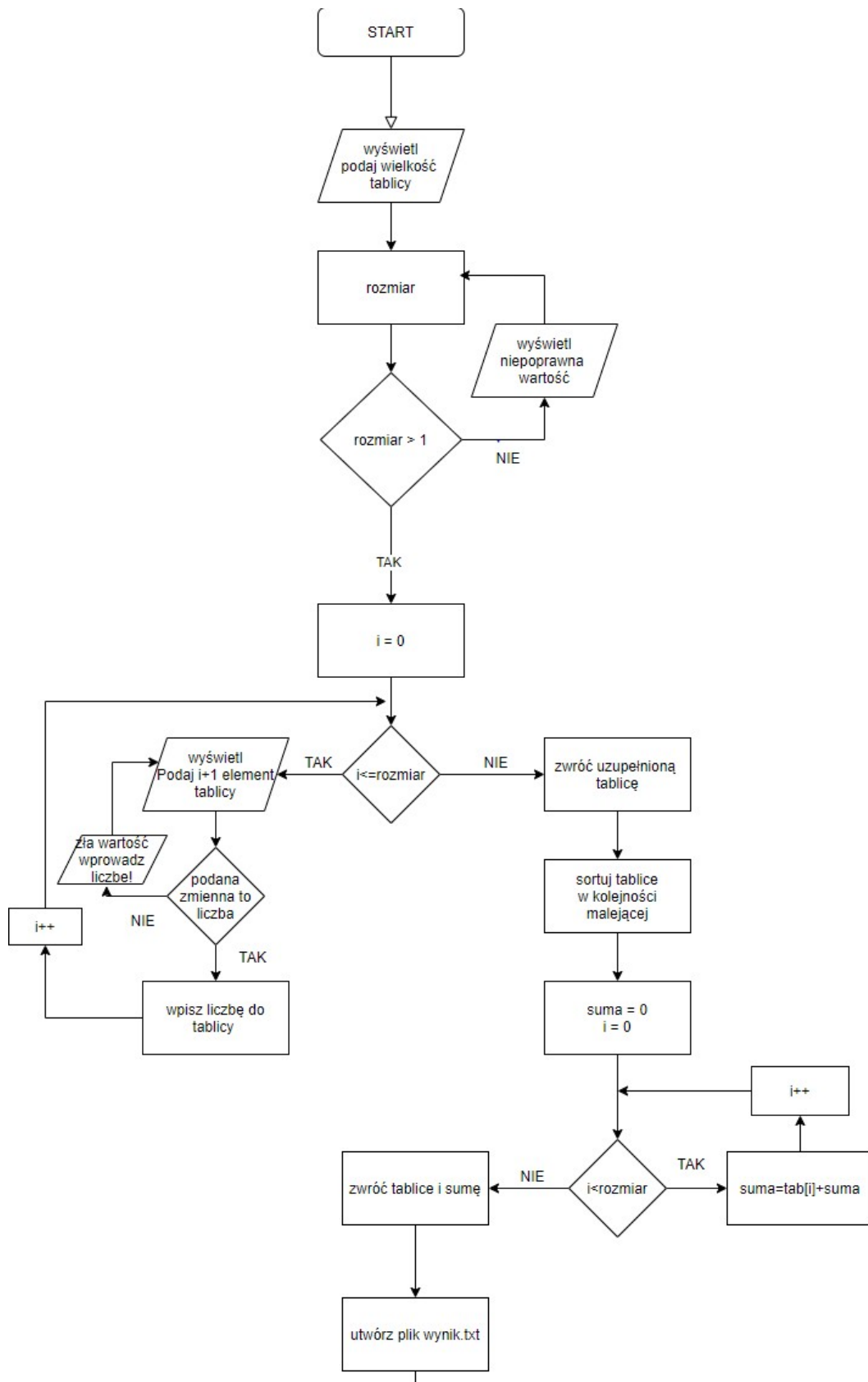
Wyświetl „koniec zadania”

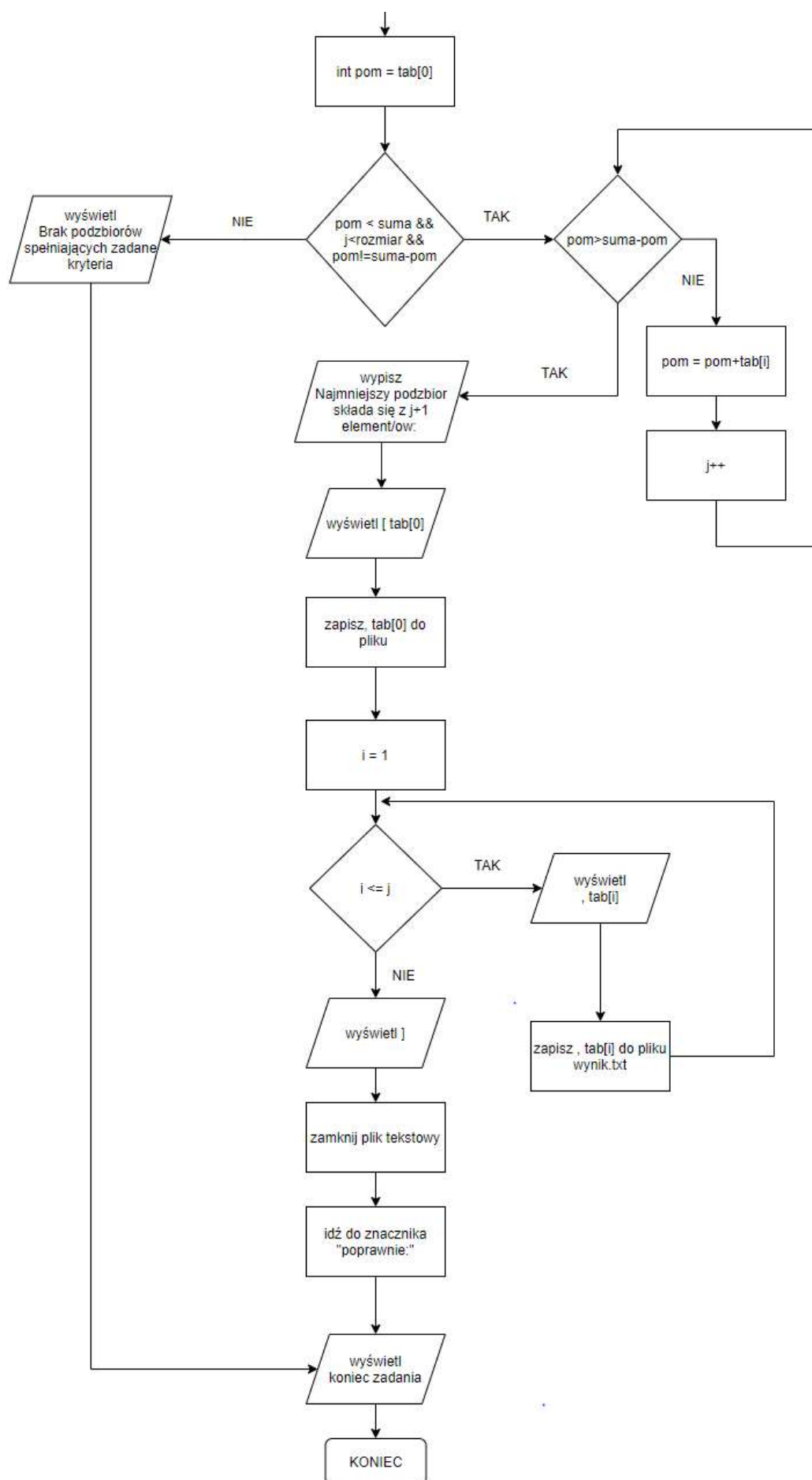
Tablica = tworzenie()

Wykonaj funkcje podZbior(od wartości tablica)

Usun tablice

2.5 SCHEMAT BLOKOWY





3 DOKUMENTACJA Z DOŚWIADCZEŃ

3.1 SPRAWDZENIE POPRAWNOŚCI PODANEGO ROZMIARU TABLICY

```
Podaj rozmiar tablicy: 5
Niepoprawna wartosc
-10
Niepoprawna wartosc
0,1
Niepoprawna wartosc
10
Podaj 1 element tablicy
```

Rysunek 3 Podawanie różnych danych rozmiaru tablicy

Program będzie prosił o podanie cyfry większej od 1. Jeżeli użytkownik będzie wpisywał znaki, liczby inne niż dodatnie całkowite itp. Program będzie wykonywał się aż zostanie podana zadana wartość.

3.2 SPRAWDZENIE POPRAWNOŚCI WPISYWANYCH ELEMENTÓW

```
Podaj rozmiar tablicy: 5
Podaj 1 element tablicy
d
Zła wartosc, wprowadz liczbe!
1
Podaj 2 element tablicy
```

Rysunek 4 Podawanie różnych elementów tablicy

Podobnie jak w poprzednim punkcie program będzie prosił o podanie odpowiedniej wartości.

3.3 PODANIE WARTOŚCI Z PRZYKŁADU

```
Podaj rozmiar tablicy: 7
Podaj 1 element tablicy
2
Podaj 2 element tablicy
4
Podaj 3 element tablicy
1
Podaj 4 element tablicy
3
Podaj 5 element tablicy
9
Podaj 6 element tablicy
0
Podaj 7 element tablicy
3
Najmniejszy podzbior sklada sie z 2 element/ow: [9, 4]
Koniec zadania

Process returned 0 (0x0)   execution time : 6.200 s
Press any key to continue.
```

Rysunek 5 Wynik działania programu z danymi z zadania

4 PODSUMOWANIE

Program wykonuje sprawdzenie danych dla wpisywanych wartości przez co użytkownik może poprawić wartość jeśli się pomylił. Dodatkową zaletą sprawdzania jest to że program nie psuje się w przypadku podania nieoczekiwanej wartości Wykorzystuje on pętle for, while, oraz różne funkcje. Działanie programu zawarte jest w funkcji. Program sprawdza jedynie „pierwszy” przypadek który spełni założenie zadania. Niestety nie jest on w stanie znaleźć innych podzbiorów z danej tablicy które również są zgodne z założeniem.

5 SPIS TREŚCI

1	Opis problemu	1
1.1	Treść zadania	1
1.2	Tablice dynamiczne	1
1.3	opis programu	1
2	program	1
2.1	kod	1
2.2	zaprogramowane operacje.....	4
2.3	wynik działania programu	5
2.4	Pseudokod	5
2.5	Schemat blokowy	8
3	dokumentacja z doświadczeń.....	10
3.1	Sprawdzenie poprawności podanego rozmiaru tablicy	10
3.2	sprawdzenie poprawności wpisywanych elementów	10
3.3	podanie wartości z przykładu	10
4	podsumowanie	11

6 SPIS ILUSTRACJI

Rysunek 1 Działanie programu w konsoli.....	5
Rysunek 2 wynik działania w dokumencie	5
Rysunek 3 Podawanie różnych danych rozmiaru tablicy	10
Rysunek 4 Podawanie różnych elementów tablicy	10
Rysunek 5 Wynik działania programu z danymi z zadania	10