



**POLITECHNIKA
RZESZOWSKA**
im. IGNACEGO ŁUKASIEWICZA

WYDZIAŁ MATEMATYKI I FIZYKI STOSOWANEJ

Kierunek studiów: Inżynieria i analiza danych

Luiza Kurosad

[166669]

Algorytmy i struktury danych
Projekt, grupa 04

Sprawozdanie

Rzeszów 2020

1. TREŚĆ ZADANIA.....	2
2. SCHEMATY BLOKOWE PROGRAMU	3
3. PSEUDOKOD	10
3.1. Void wpisz() {	10
3.2. Void wczytaj_z_pliku (.....	10
3.3. Void stworzenieb() {	11
3.4. Void zapisz_do_pliku(){	12
3.5. Void funkcja(){.....	12
4. KOD PROGRAMU.....	14
4.1. void wpisz()	14
4.2. void wczytaj_z_pliku()	14
4.3. void stworzenieb()	15
4.4. void zapisz_do_pliku()	15
4.5. void funkcja()	15
5. TESTY PROGRAMU	16
6. SZCZEGÓŁOWY OPIS IMPLEMENTACJI PROBLEMU	22
7. WNIOSKI I PODSUMOWANIE PROJEKTU	24
BIBLIOGRAFIA.....	26
SPIS ILUSTRACJI	26
SPIS SCHEMATÓW BLOKOWYCH.....	26

1. Treść zadania

Zadanie

Dla zadanej tablicy A różnych liczb całkowitych, utwórz tablicę B o tej samej długości, której każdy element $B[i]$ będzie równy liczbie elementów tablicy A, leżących na prawo od elementu $A[i]$ i większych od niego.

Przykład:

Wejście $A = [4, 6, 3, 9, 7, 10]$

Wyjście: $B = [4, 3, 3, 1, 1, 0]$

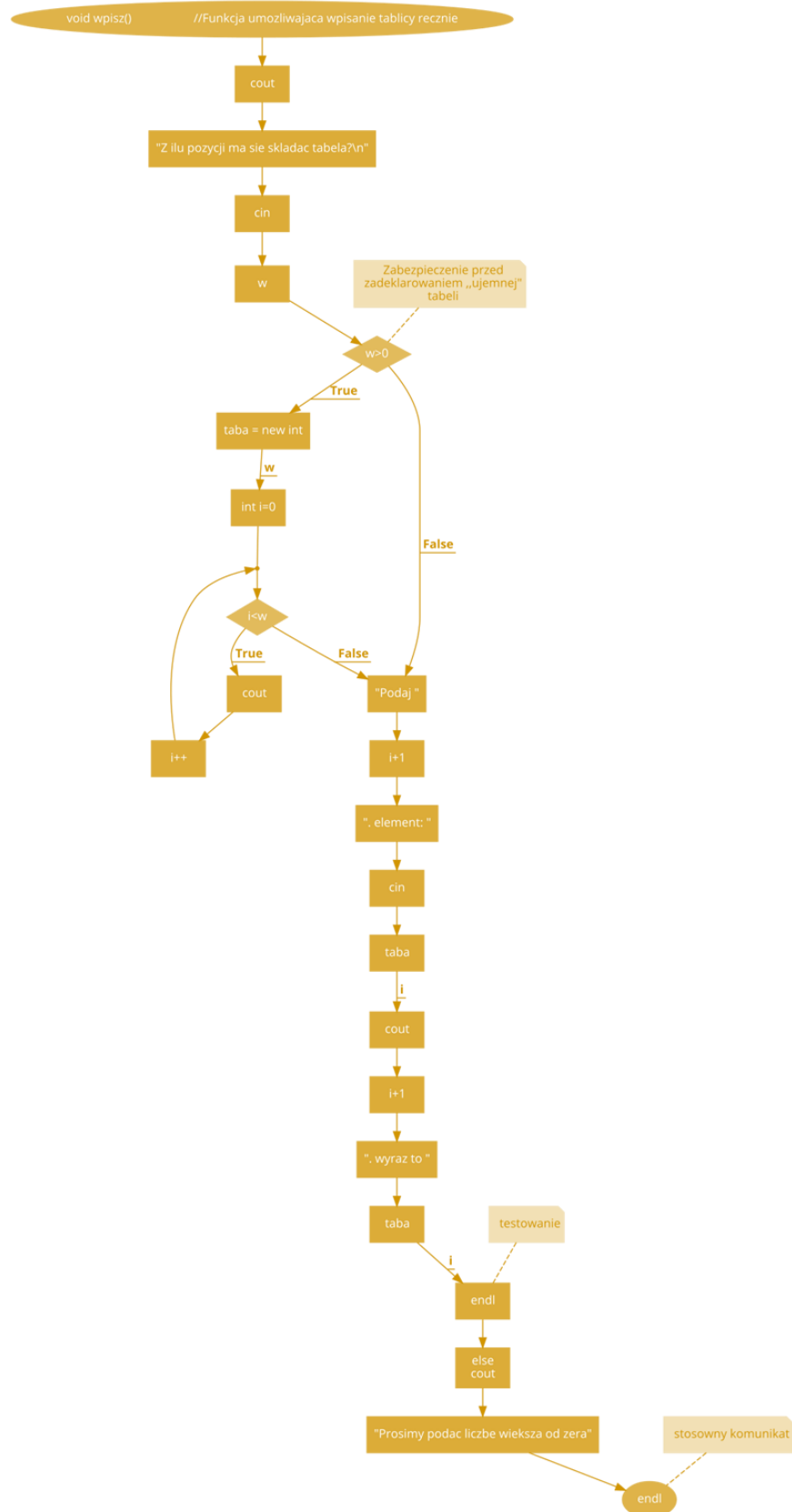
Wejście $A = [6, 5, 8, 4, 4, 2]$

Wyjście: $B = [1, 1, 0, 0, 0, 0]$

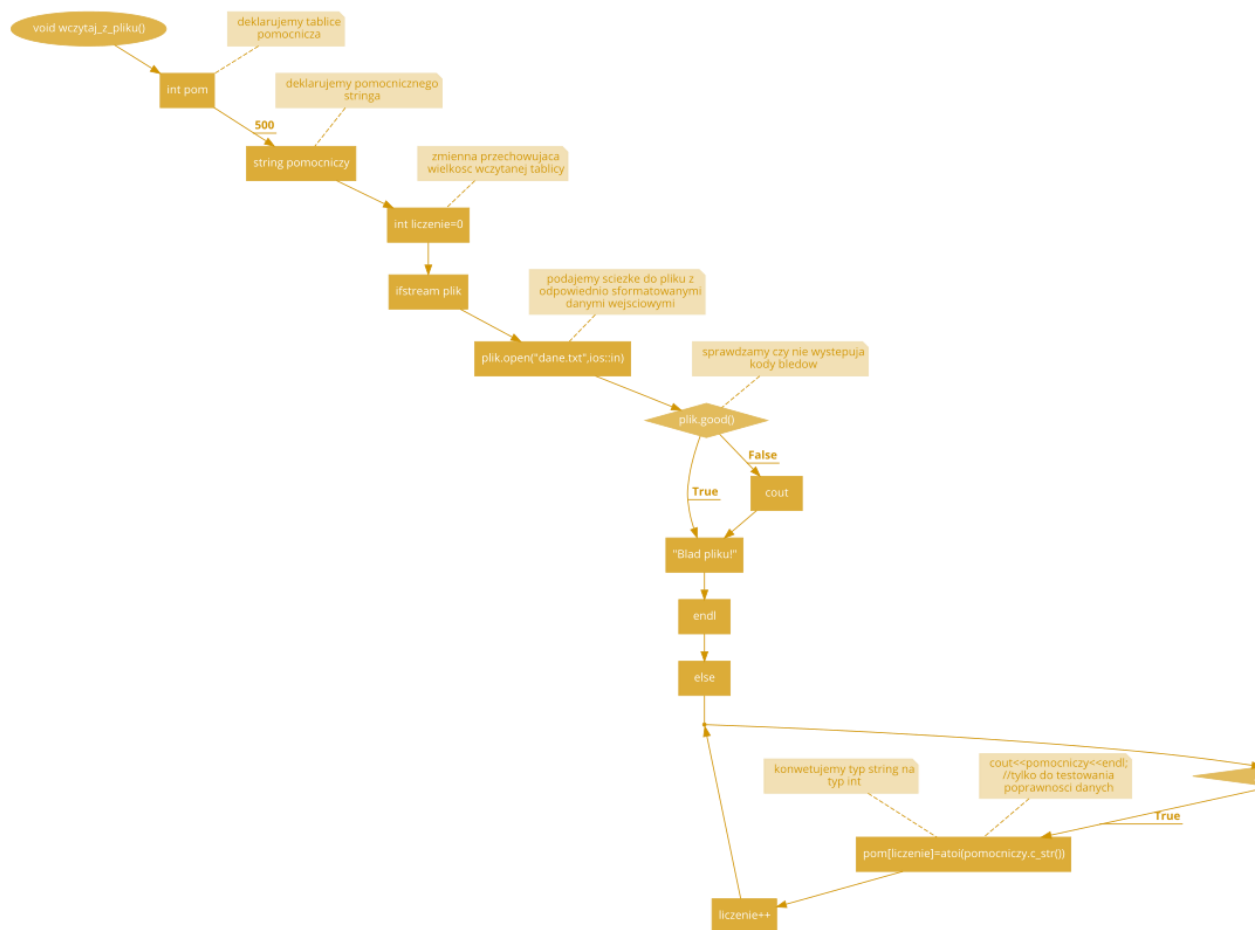
Kryteria dotyczące działania programu:

- główny kod programu powinien być zaimplementowany w oddzielnej funkcji, którą powinna być wywoływana wewnątrz programu,
- w głównym programie powinno zostać wykonane kilka testów sprawdzających działanie funkcji,
- program powinien mieć możliwość odczytywania danych wejściowych i zapisu wyników do plików tekstowych,
- kod powinien być opatrzony stosownymi komentarzami

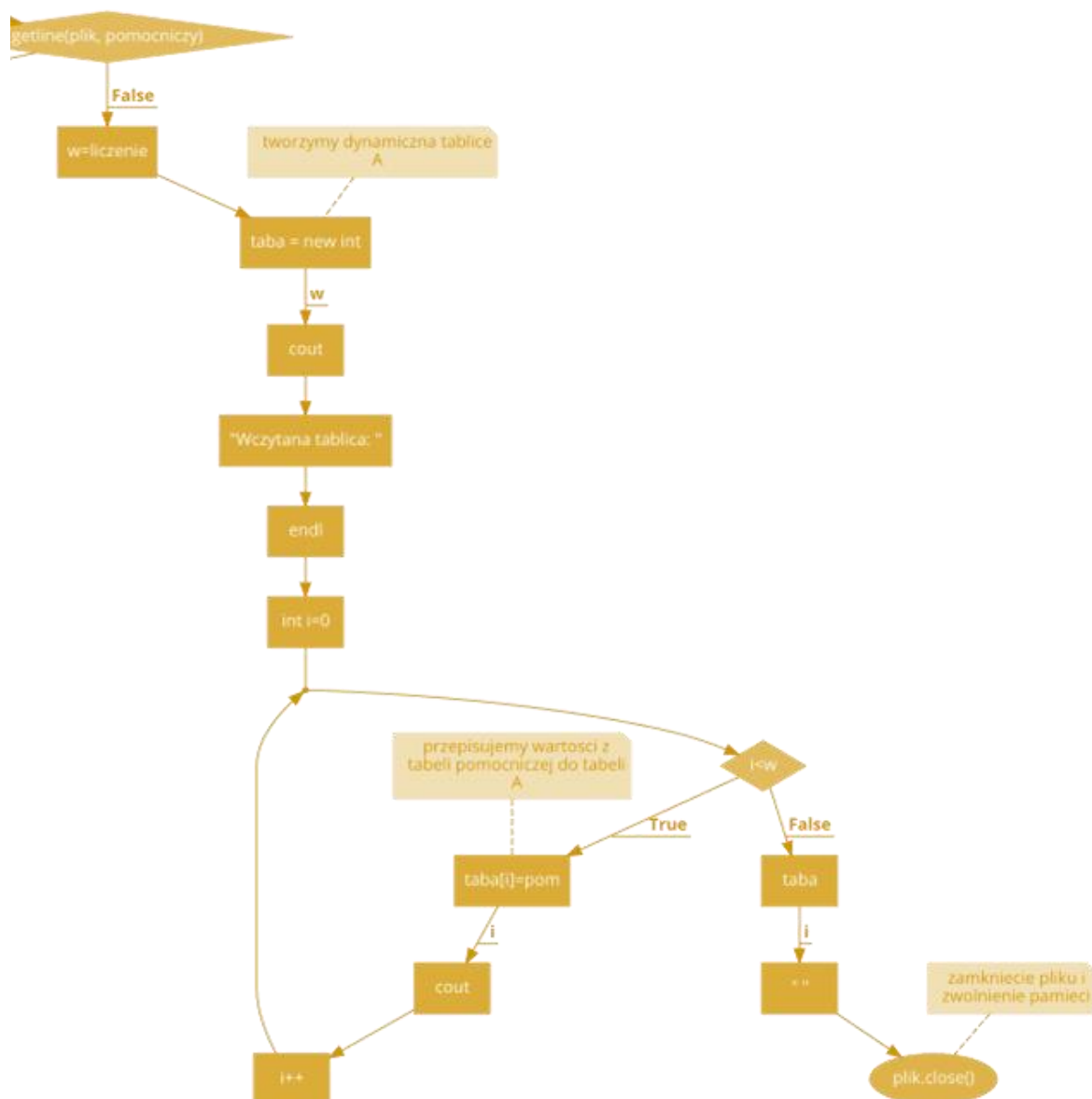
2. Schematy blokowe programu



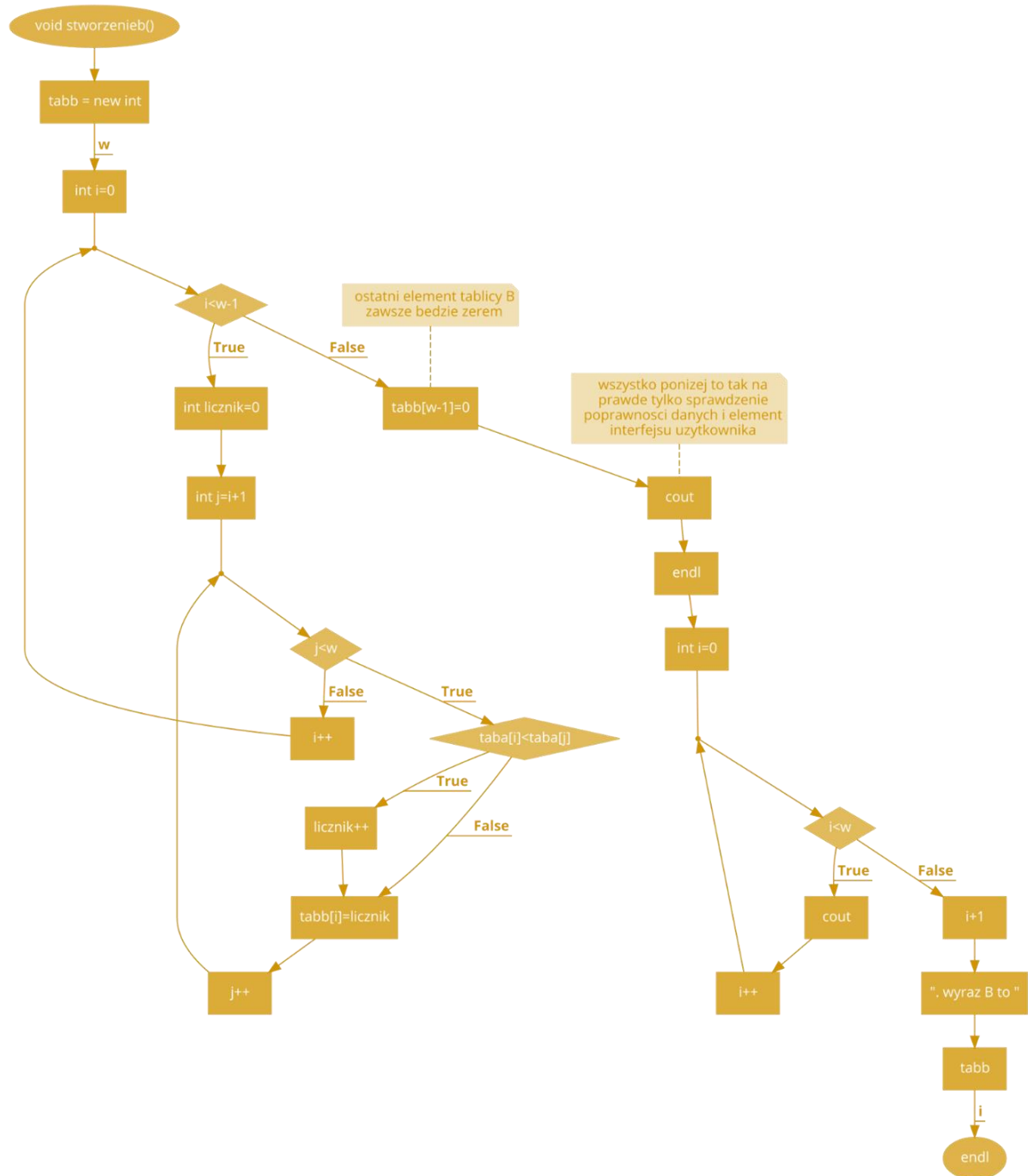
Schemat blokowy 1 Funkcja wypisz()



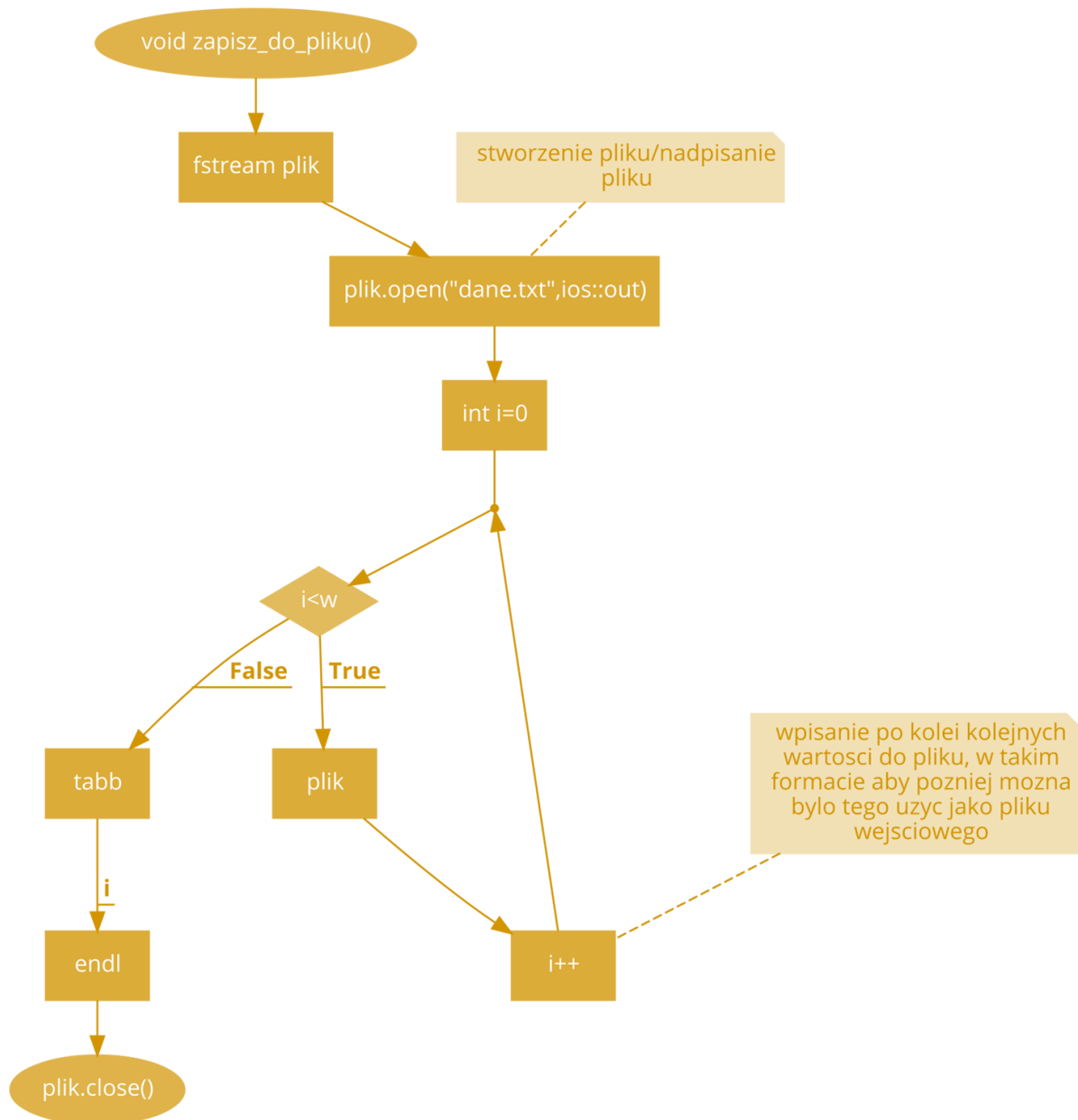
Schemat blokowy 2 Funkcja wczytaj_z_pliku (cz.1)



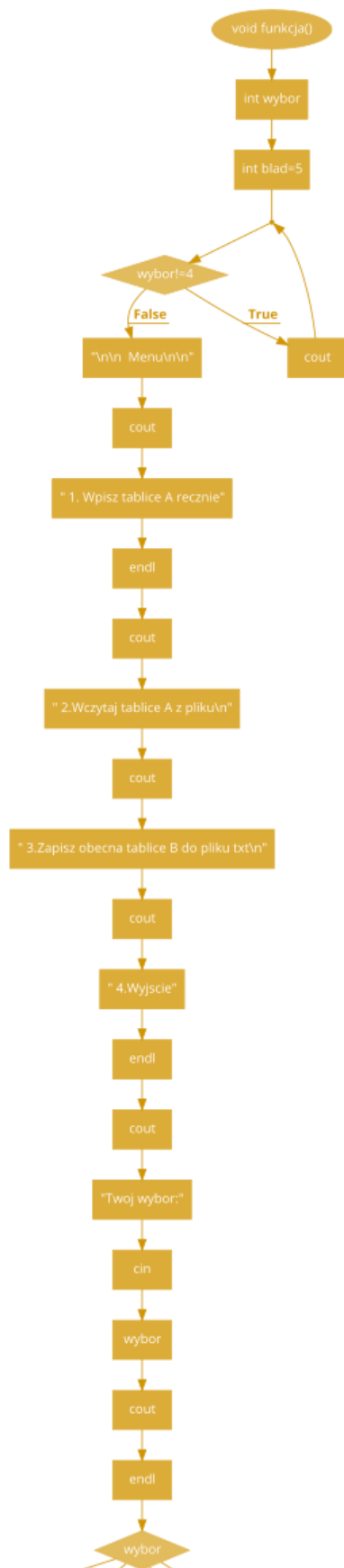
Schemat blokowy 3 funkcja wczytaj_z_pliku (cz.2)



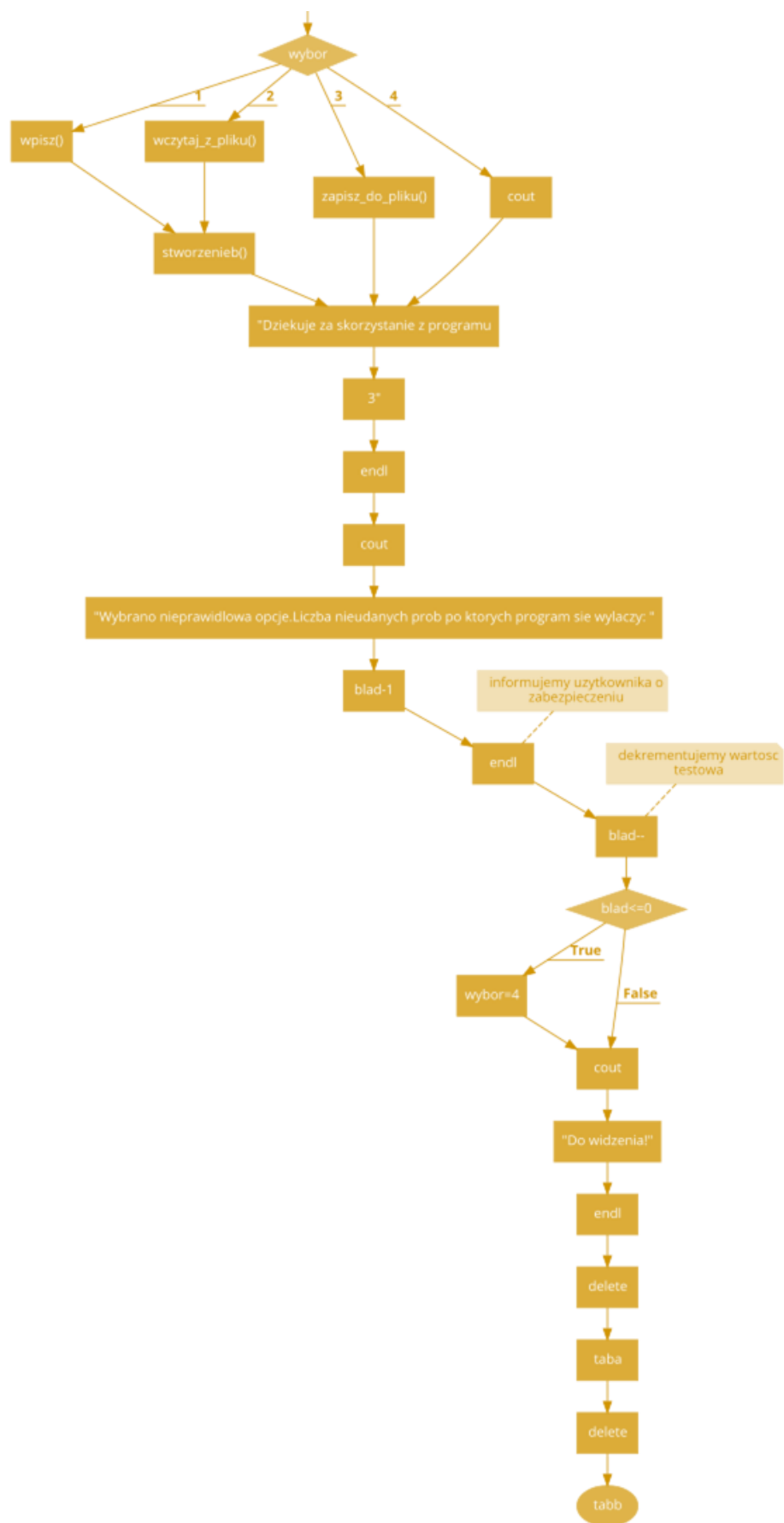
Schemat blokowy 4 Funkcja stworzenieb()



Schemat blokowy 5 Funkcja zapisz_do_pliku()



Schemat blokowy 6 Funkcja () (cz.1)



Schemat blokowy 7 Funkcja () (cz.2)

3. Pseudokod

3.1. Void wpisz() {

k01: $k \leftarrow$ rozmiar tablicy

//zmienna pomocnicza, zadeklarowana w celu braku możliwości utworzenia tablicy o niewłaściwym rozmiarze

k02: wypisz: "Z ilu pozycji ma sie składać tablica?"

k03: wczytaj k;

//tworzenie dynamicznej tablicy

k04: jeżeli $k \leq 0$

wypisz: "Prosimy podać liczbę większą od zera";

//wypisanie komunikatu przez konsolę

w przeciwnym razie

dla $w=k$

k05: utwórz taba[w]

//utworzenie tablicy a

k06: dla $i=0$; $i < w$ wykonaj $i+1$

//implementacja i

k07: wypisz "Podaj "i+1". element:"

// Polecenie podania kolejnego elementu tablicy

k08: wprowadź $i+1$ do taba [i]

//wprowadzanie danych do tablicy

k09: wypisz $i+1$ "wyraz to" taba[i] }

//wypisanie przez program wprowadzonego elementu tablicy

3.2. Void wczytaj_z_pliku (

k01: utwórz tabb[w]

//deklarujemy tablice pomocniczą

k02: int pom[500] wykonuj

//deklarujemy maksymalny rozmiar tablicy

k03: liczenie = 0

//ustaw liczenie na 0

k04: jeżeli (!plik.good())

wypisz "Błąd pliku!"

//poinformuj o błędzie pliku

w przeciwnym razie getline(plik, pomocniczy)

//pobierz linię z pliku

```

pom[liczenie]=atoi(pomocniczy.c_str())
//zamień wartość string na int
liczenie + 1
//implementuj liczenie
k05:  w ←liczenie
k06:  utwórz taba
      //tworzenie tablicy a
k07:  wypisz: "Wczytana tablica"
      //dla spełnionego warunku wypisz tablicę a
      dla int i=0; i<w wykonaj i + 1;
      przepisz wartość z pom[i] do taba[i];
      wypisz taba[i];
k08:  zamknij program - plik.close(); }
      //zwolnij pamięć, zamknij program;

```

3.3. Void stworzenieb() {

```

k01:  utwórz tabb[w]
      //stworzenie tablicy b
k02:  dla i=0 i i<w-1 wykonaj i+1
      //dla spełnionego warunku wykonaj implementację i
k03:  licznik←0
      //wyzerowanie początkowej wartości licznika
k04:  dla int j=i+1 i j<w wykonaj j++
k05:  jeżeli taba[i]<taba[j] to wykonaj licznik+1;
k06:  tabb[i]=licznik;
      //przypisz licznik do tabb[i]
k07:  tabb[w-1]← 0;
      //dla ostatniej wartości przypisz 0
      sprawdzenie
k08:  dla int i=0 i i<w wykonaj i+1
k09:  wypisz i+1 wyraz B to" tabb[i]}
      //wypisanie przez program elementu B przypisanego do danego elementu

```

3.4. Void zapisz_do_pliku(){

k01: deklaracja fstream plik;
//deklaracja fstream umożliwia dalsze działanie z plikiem zewnętrznym (w naszym przypadku „dane.txt”)
k02: plik.open stwórz/nadpisz plik "dane.txt"
k03: dla int i=0 i i<w wykonaj i+1
//wpisanie po kolei kolejnych wartości do pliku, tak, aby później można ich użyć jako pliku wejściowego
k04: zapisz w pliku tabb[i]
k05: zamknij program - plik.close(); }
//zwolnij pamięć, zamknij program;

3.5. Void funkcja(){

k01: int wybor; int blad=5;
k02: dopóki wybor!=4
wykonuj:
k03: wypisz "MENU GLOWNE PROGRAMU"
k04: wypisz: "*****"
k05: wypisz: "1. Wpisz tablice A recznie";
k06: wypisz: "2. Wczytaj tablice A z pliku\n";
k07: wypisz: "3. Zapisz obecna tablice B do pliku txt\n";
k08: wypisz : "4. Wyjscie"<<endl<<endl;
k09: wypisz: "*****"
//tworzenie menu programu w konsoli
k10: wypisz: "Twój wybór:"
k11: wprowadź wybór
k12: instrukcja switch
k13: jeżeli wybór to 1
wykonaj instrukcję stworzenieb
k14: przerwij pętle
k13: jeżeli wybór to 2
wykonaj instrukcję wczytaj_z_pliku();
//wczytaj tablicę za pomocą importu danych
k14: przerwij pętle
k15: jeżeli wybór to 3
wykonaj zapisz_do_pliku();

k16: przerwij pętle
//zapisz tablicę do pliku zewnętrznego

k17: jeżeli wybor to 4
 wypisz "Dziekuje za skorzystanie z programu :)"

k18: przerwij pętle
w przeciwnym razie

k19: wypisz: "Wybrano nieprawidlowa opcje."

k20: wypisz: "Liczba nieudanych prob po ktorych program sie wylaczy: " wprowadzenie
 zmiennej blad

k21: blad-1

k22: jeśli blad<=0
 to wybor=4
 //wprowadzamy liczbę możliwych błędnych prób, po których nastąpi zamknięcie programu

k23: wypisz: "Do widzenia!"

k24: usuń [] taba;
 //zwolnij pamięć, zamknij tablicę

k25: usuń [] tabb;}
 //zwolnij pamięć, zamknij tablicę

4. Kod programu

W celu prawidłowego funkcjonowania programu zastosowane zostały następujące funkcje:

4.1. void wpisz()

```
1  #include <iostream>
2  #include<fstream>
3  #include <cstdlib>
4  #include<string>
5
6  using namespace std;
7  //Obsługa plików
8
9  int w;//wielkość tablic
10 int *taba; //Tablica A
11 int *tabb; //Tablica B
12 //a właściwie ich deklaracje
13
14
15 void wpisz() //Funkcja umożliwiająca wpisanie tablicy ręcznie
16 {
17     int k; //Ta zmienna pomocnicza powstała na wypadek gdyby
18     //jednak ktoś postanowił, dać nieprawidłowy rozmiar
19     cout<<"Z ilu pozycji ma się składać tablica?\n";
20     cin>>k; //gdymy tutaj było w to kompilator wyrzuciłby błąd alokacji
21
22     if(k<=0){ //Zabezpieczenie przed zadeklarowaniem „ciemnej” tablicy
23         cout<<"Prosimy podać liczbę większą od zera"<<endl; } //stosowny komunikat
24     else{
25         w=k;
26         taba = new int[w]; //deklaracja tablicy dynamicznej
27         for(int i=0; i<w; i++)
28         {
29             cout<<"Podaj " <<i+1<<". element: ";
30             cin>>taba[i];
31             cout<<i+1<<". wyraz to "<<taba[i]<<endl; //testowanie
32         }
33     }
34 }
35
36
```

Obraz 1 Kod void wpisz()

4.2. void wczytaj_z_pliku()

```
36 void wczytaj_z_pliku()
37 {
38     int pom[500]; //deklarujemy tablice pomocnicza
39     string pomocniczy; //deklarujemy pomocnicznego stringa
40     int liczenie=0; //zmienna przechowująca wielkość wczytanej tablicy
41
42     ifstream plik;
43     plik.open("dane.txt",ios::in); //podaemy ścieżkę do pliku z odpowiednio sformatowanymi danymi wejściowymi
44     if(!plik.good()) //sprawdzamy czy nie wystąpiła kody błędów
45     {
46         cout << "Błąd pliku!" << endl;
47     }
48     else
49     {
50         while(getline(plik, pomocniczy))
51         {
52             //cout<<pomocniczy<<endl; //tylko do testowania poprawności danych
53             pom[liczenie]=atoi(pomocniczy.c_str()); //konwertujemy typ string na typ int
54             liczenie++;
55         }
56         w=liczenie;
57         taba = new int[w]; //tworzymy dynamiczną tablicę A
58
59         cout<<"Wczytana tablica: "<<endl;
60         for(int i=0; i<w; i++)
61         {
62             taba[i]=pom[i]; //przenisujemy wartości z tabeli pomocniczej do tabeli A
63             cout<<taba[i]<<" ";
64         }
65
66         plik.close(); //zamknięcie pliku i zwolnienie pamięci
67     }
68 }
69
70
71
```

Obraz 2 Kod void wczytaj_z_pliku

4.3. void stworzenieb()

```
76 void stworzenieb()
77 {
78     tabb = new int[w];
79
80     for(int i=0; i<w-1; i++)
81     {
82         int licznik=0;
83         for(int j=i+1; j<w; j++)
84         {
85             if(taba[i]<taba[j])
86             {
87                 licznik++;
88             }
89             tabb[i]=licznik;
90         }
91     }
92
93     tabb[w-1]=0; //ostatni element tablicy B zawsze bedzie zerem
94     //wszystko ponizej to tak na prawde tylko sprawdzenie poprawnosci danych i element interfejsu uzytkownika
95     cout<<endl;
96
97     for(int i=0; i<w; i++)
98     {
99         cout<<i+1<<". WYRAZ B to "<<tabb[i]<<endl;
100     }
101 }
102
103
104
```

Obraz 3 Kod void stworzenieb()

4.4. void zapisz_do_pliku()

```
106 void zapisz_do_pliku()
107 {
108     fstream plik;
109     plik.open("dane.txt",ios::out); //stworzenie pliku/nadpisanie pliku
110     for(int i=0; i<w; i++) //wpisanie po kolei kolejnych wartosci do pliku, w takim formacie,
111     { //aby pozniej mozna bylo tego uzyc jako pliku wejsciwego
112
113         plik<<tabb[i]<<endl;
114     }
115
116     plik.close();
117 }
118
119
```

Obraz 4 Kod void zapisz_do_pliku

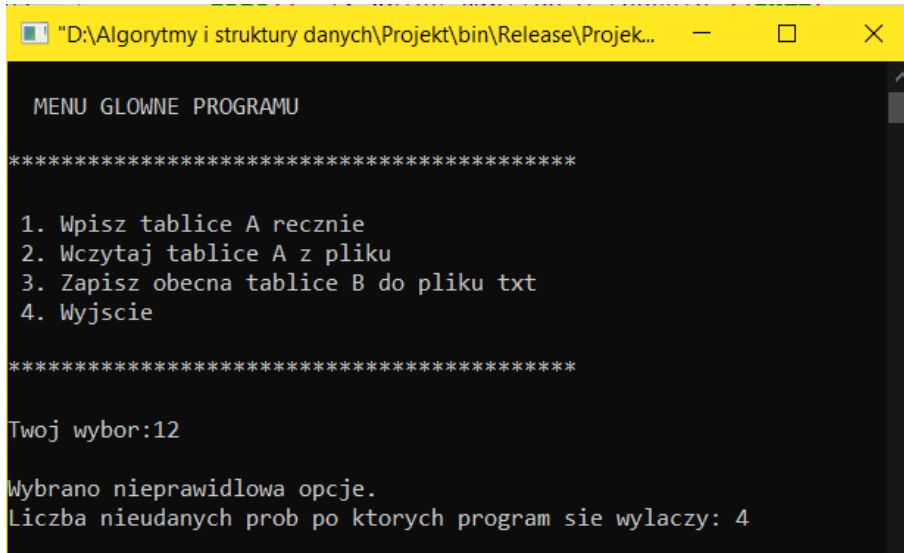
4.5. void funkcja()

```
121 void funkcja()
122 {
123     int wybor;
124     int blad=5;
125     while(wybor!=4)
126     {
127         cout<<"\n MENU GLOWNE PROGRAMU \n\n";
128         cout<<"*****" <<endl<<endl;
129         cout<<" 1. Wpisz tablice A recznie"<<endl;
130         cout<<" 2. Wczytaj tablice A z pliku\n";
131         cout<<" 3. Zapisz obecna tablice B do pliku txt\n";
132         cout<<" 4. Wyjscie"<<endl<<endl;
133         cout<<"*****" <<endl<<endl;
134         cout<<"Twój wybór:";
135         cin>>wybor;
136         cout<<endl;
137         switch(wybor)
138         {
139             case 1:
140                 wpisz();
141                 stworzenieb();
142                 break;
143             case 2:
144                 wczytaj_z_pliku();
145                 stworzenieb();
146                 break;
147             case 3:
148                 zapisz_do_pliku();
149                 break;
150             case 4:
151                 cout<<"Dziekuje za skorzystanie z programu :) "<<endl;
152                 break;
153             default:
154                 cout<<"Wybrano nieprawidlowa opcje.Liczba nieudanych prob po ktorzych program sie wylaczy: "<<blad-1<<endl;
155                 //informujemy uzytkownika o zabezpieczeniu
156                 //dekrementujemy wartosc testowa
157                 blad--;
158                 if(blad<=0)
159                     wybor=4;
160         }
161     }
162     cout<<"Do widzenia!"<<endl;
163     delete [] taba;
164     delete [] tabb;
165 }
```

Obraz 5 Kod void funkcja()

5. Testy programu

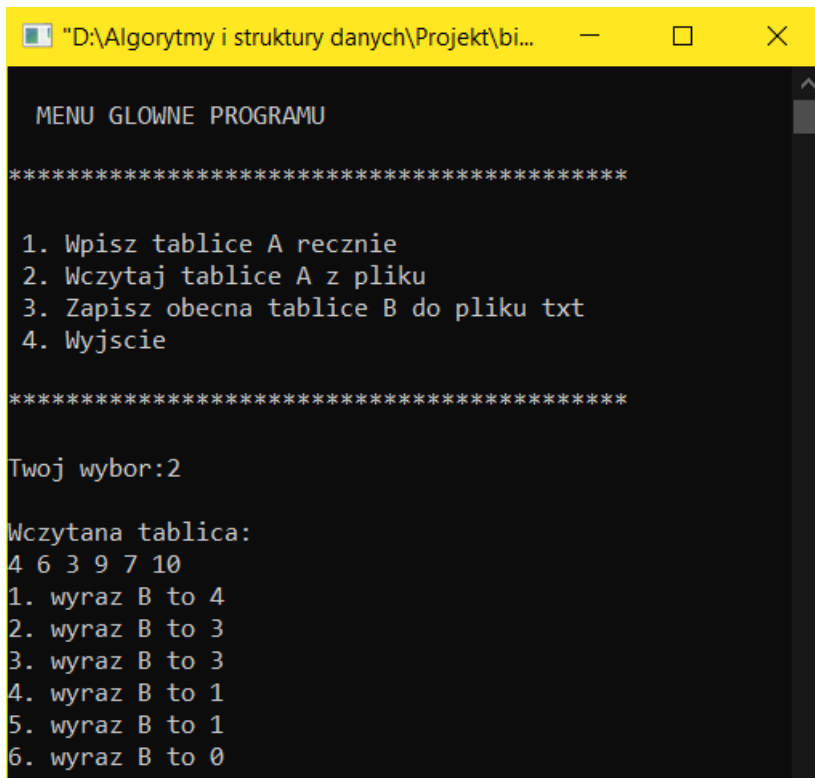
Wprowadzenie niewłaściwej opcji z menu głównego programu, skutkuje informacją od konsoli. Dodatkowo otrzymujemy informację o zabezpieczeniu, które po 4 kolejnych błędnych operacjach zamknie program.



```
"D:\Algorytmy i struktury danych\Projekt\bin\Release\Projek...
MENU GLOWNE PROGRAMU
*****
1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście
*****
Twoj wybor:12
Wybrano nieprawidlowa opcje.
Liczba nieudanych prob po ktorych program sie wylaczy: 4
```

Obraz 6 Obraz konsoli. Test programu nr 1. Błąd. Nieprawidłowy wybór.

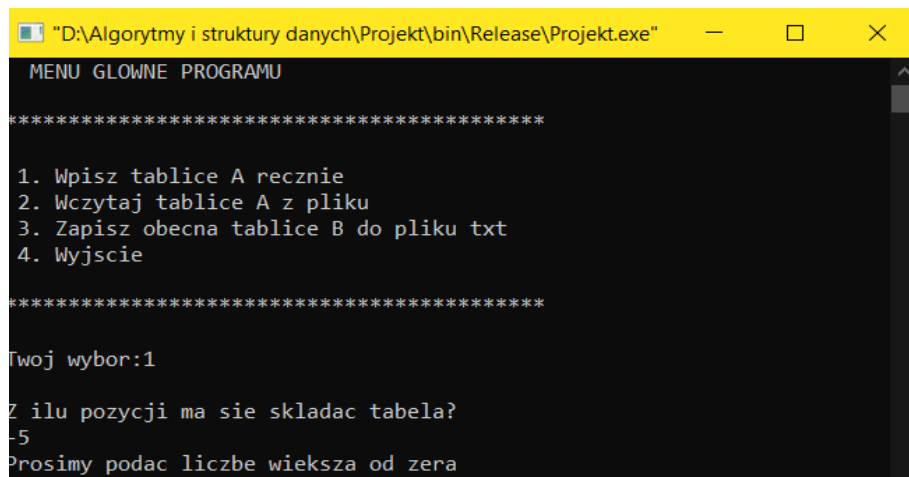
Test 2 opcji programu. Próba wczytania tablicy A z pliku „dane.txt”. Program zadziałał prawidłowo.



```
"D:\Algorytmy i struktury danych\Projekt\bi...
MENU GLOWNE PROGRAMU
*****
1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście
*****
Twoj wybor:2
Wczytana tablica:
4 6 3 9 7 10
1. wyraz B to 4
2. wyraz B to 3
3. wyraz B to 3
4. wyraz B to 1
5. wyraz B to 1
6. wyraz B to 0
```

Obraz 7 Obraz konsoli. Test programu nr 2. Import elementów tablicy z pliku „dane.txt”

Testowanie programu pod kątem wprowadzenia niewłaściwych wartości, podczas generowania tablicy i wprowadzenia jej wymiarów. (Błąd występuje zarówno w przypadku próby podania liczby ujemnej, jak i litery.)



```
"D:\Algorytmy i struktury danych\Projekt\bin\Release\Projekt.exe"
MENU GLOWNE PROGRAMU
*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście

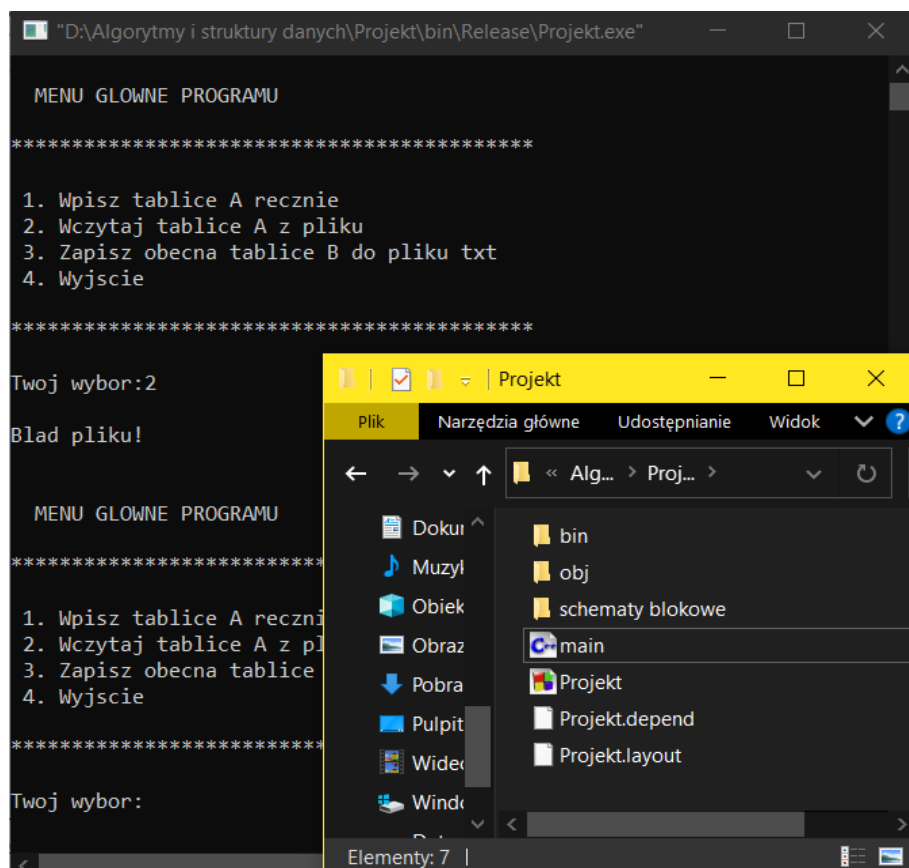
*****

Twój wybór:1

Z ilu pozycji ma sie skladac tabela?
-5
Prosimy podac liczbe wieksza od zera
```

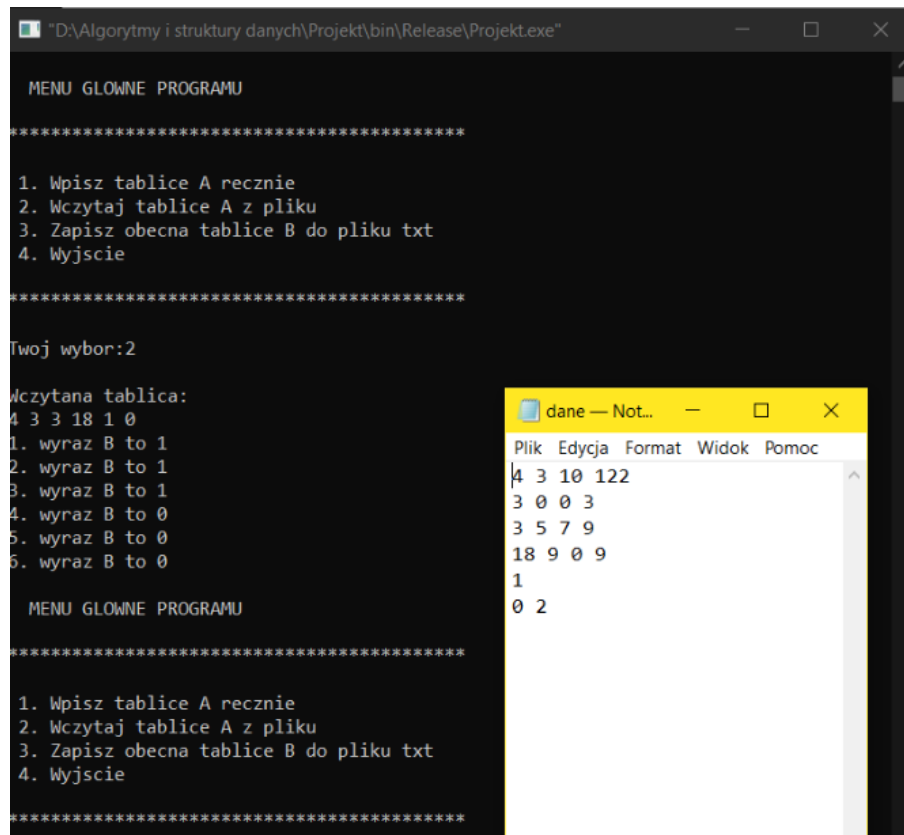
Obraz 8 Obraz konsoli. Test programu nr 3. Błąd. Ujemna wartość zadeklarowanej tablicy.

Nieudana próba importowania tablicy A z pliku zewnętrznego. Przyczyną błędu jest brak pliku „dane.txt” w folderze projektu.



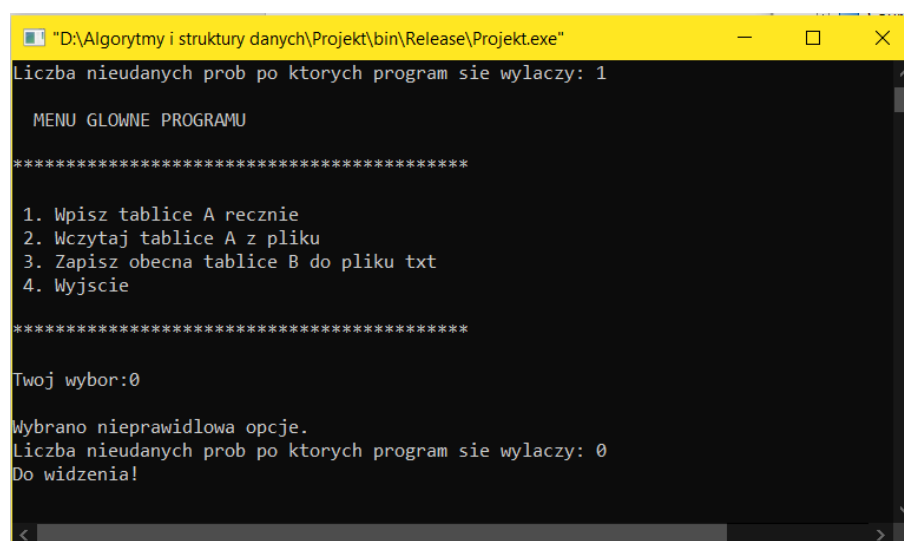
Obraz 9 Obraz konsoli. Test programu nr 4. Błąd podczas próby importu elementów tablicy z pliku „dane.txt”. Brak wspomnianego dokumentu w folderze projektu.

Błąd importowania tablicy, w przypadku, gdy w dokumencie tekstowym („dane.txt”) znajduje się kilka wierszy, w których zapisane zostały kolejno liczby, zastosowując różną ilość elementów. Wówczas konsola tworzy tablicę wykorzystując jedynie 1 element każdego z wierszy.



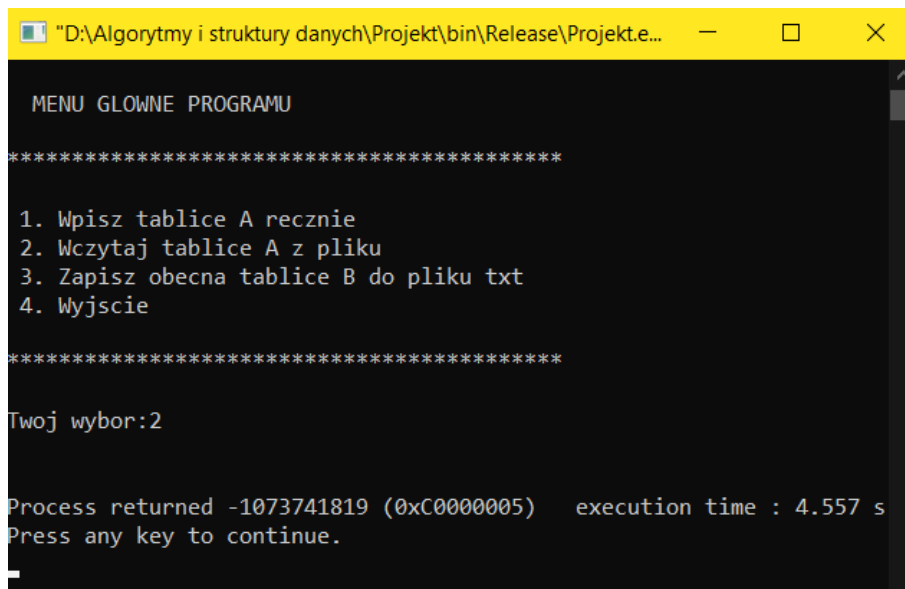
Obraz 10 Obraz konsoli. Test programu nr 6. Błąd importu tablicy z pliku.

Proces kilkukrotnego wyboru innej spośród czterech domyślnych opcji programu (1-4), skutkuje wyłączeniem programu, o czym wcześniej informuje nas komunikat:



Obraz 11 Obraz konsoli. Test programu nr 7. Błąd. Nieprawidłowa opcja wyboru.

Podczas próby importowania tablicy zawierającej więcej elementów, niż zadeklarowana maksymalna wartość, otrzymujemy błąd programu, zasygnalizowany następującym komunikatem:



```
"D:\Algorytmy i struktury danych\Projekt\bin\Release\Projekt.e...  —  □  ×

MENU GLOWNE PROGRAMU

*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście

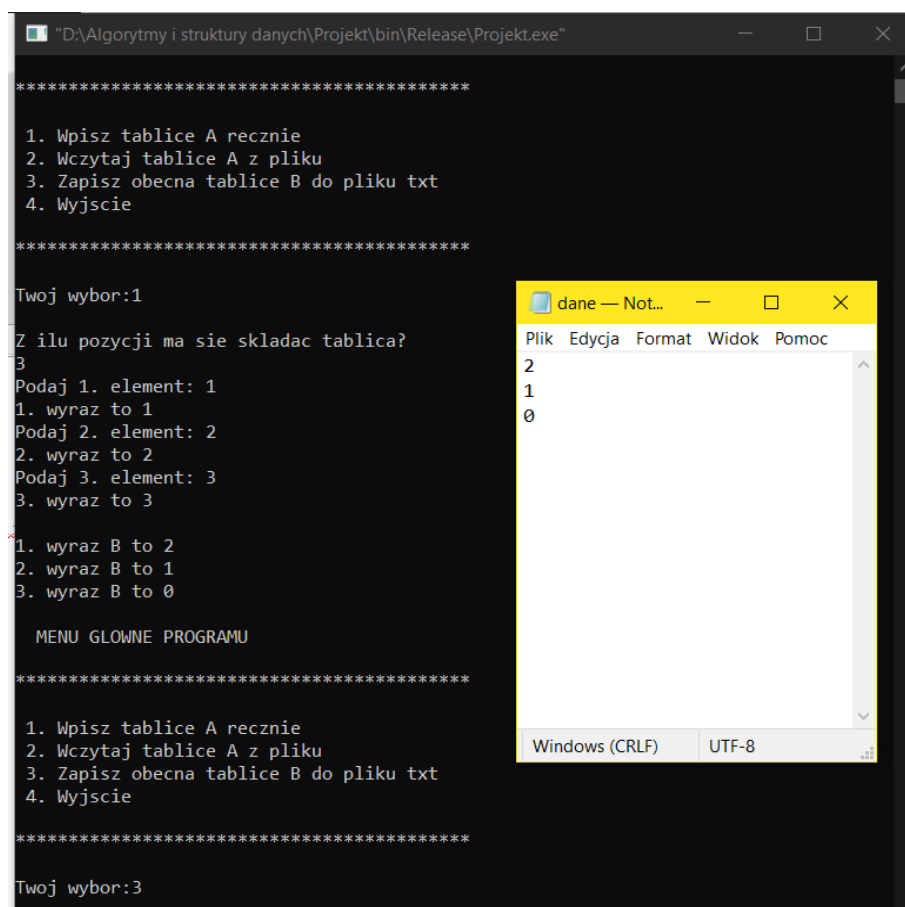
*****

Twój wybór:2

Process returned -1073741819 (0xC0000005)   execution time : 4.557 s
Press any key to continue.
_
```

Obraz 12 Obraz konsoli. Test programu nr 8. Błąd. Zbyt duża zadeklarowana wartość tablicy.

Podczas zapisywania wyników tablicy b w dokumencie „dane.txt” następuje nadpisywanie wyników uzyskanych podczas poprzednich operacji programu



```
"D:\Algorytmy i struktury danych\Projekt\bin\Release\Projekt.exe"

*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście

*****

Twój wybór:1
Z ilu pozycji ma sie skladac tablica?
3
Podaj 1. element: 1
1. wyraz to 1
Podaj 2. element: 2
2. wyraz to 2
Podaj 3. element: 3
3. wyraz to 3

1. wyraz B to 2
2. wyraz B to 1
3. wyraz B to 0

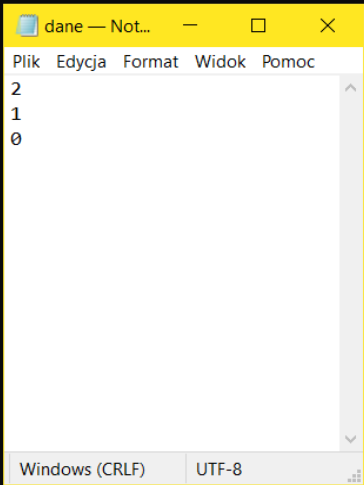
MENU GLOWNE PROGRAMU

*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście

*****

Twój wybór:3
```



```
dane — Not...  —  □  ×
Plik  Edycja  Format  Widok  Pomoc
2
1
0
Windows (CRLF)  UTF-8
```

Obraz 13 Obraz konsoli. Test programu nr 9. Wyniki zapisane w dokumencie "dane.txt" przed operacją nadpisu.

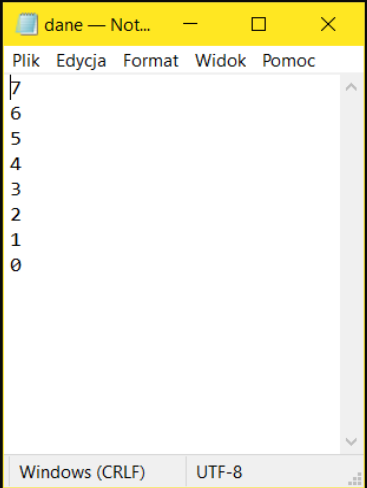
```
1. wyraz to 1
Podaj 2. element: 2
2. wyraz to 2
Podaj 3. element: 3
3. wyraz to 3
Podaj 4. element: 4
4. wyraz to 4
Podaj 5. element: 5
5. wyraz to 5
Podaj 6. element: 6
6. wyraz to 6
Podaj 7. element: 7
7. wyraz to 7
Podaj 8. element: 8
8. wyraz to 8

1. wyraz B to 7
2. wyraz B to 6
3. wyraz B to 5
4. wyraz B to 4
5. wyraz B to 3
6. wyraz B to 2
7. wyraz B to 1
8. wyraz B to 0

MENU GLOWNE PROGRAMU
*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjście
*****

Twój wybór:3
```



Obraz 14 Obraz konsoli. Test programu nr 10. Wyniki zapisane w dokumencie "dane.txt" po operacji nadpisu.

Próba wprowadzenia danych wejściowych poprzez ręczne wprowadzenie tablicy. Wynikiem działania aplikacji jest uzyskanie wyników w postaci podtablicy, zapisanych do dokumentu tekstowego.

```
"D:\Algorytmy i struktury danych\Projekt\bin\Release\P...
MENU GLOWNE PROGRAMU
*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
3. Zapisz obecna tablice B do pliku txt
4. Wyjscie

*****

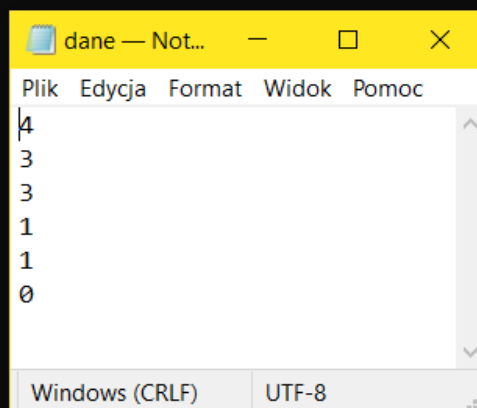
Twoj wybor:1

Z ilu pozycji ma sie skladac tablica?
6
Podaj 1. element: 4
1. wyraz to 4
Podaj 2. element: 6
2. wyraz to 6
Podaj 3. element: 3
3. wyraz to 3
Podaj 4. element: 9
4. wyraz to 9
Podaj 5. element: 7
5. wyraz to 7
Podaj 6. element: 10
6. wyraz to 10

1. wyraz B to 4
2. wyraz B to 3
3. wyraz B to 3
4. wyraz B to 1
5. wyraz B to 1
6. wyraz B to 0

MENU GLOWNE PROGRAMU
*****

1. Wpisz tablice A recznie
2. Wczytaj tablice A z pliku
```



Obraz 15 Obraz konsoli. Test programu nr 11. Wprowadzenie ręczne wprowadzenie danych oraz zapis do tabb.

6. Szczegółowy opis implementacji problemu

Funckja main ()

Główną funkcję (a właściwie metodę, dla uproszczenia dalej również nazywaną funkcją) programu wywołujemy z funkcji main(), w której użytkownik ma dostęp do przygotowanego oraz zabezpieczonego interfejsu, z którego wybiera jedno (z kilku) poleceń konsoli. Interfejs ten jest pętlą while o warunku dopóki (wybór nie jest równy 4.)

Budowa menu programu

Gdy użytkownik wybierze:

1-Uruchamia funkcję wpisz()* , a następne funkcję() stworzenieb(),

2-Uruchamia funkcję wczytaj_z_pliku(), a następne funkcję stworzenieb(),

3-Uruchamia funkcję zapisz_do_pliku(),

4-Uruchamia proces zamykania programu, (a właściwie przerywa funkcję while, która żegna go stosownym komunikatem, po tej komendzie funkcja main zwraca już wartość 0, co skutkuje zakończeniem działania aplikacji konsolowej.

Gdy użytkownik nie wybierze żadnej z powyższych opcji. Program poinformuje go o nieprawidłowej funkcji wyboru oraz wyświetli ile prób pozostało, do momentu wyłączenia programu. Po odliczeniu do 0. Program wyłączy się wyświetlając stosowny (inny) komunikat.

Opis zastosowanych funkcji

Funkcja	Opis postępowania
wpisz()	Jest to funkcja prosząca użytkownika o podanie rozmiaru pożądanej przez niego tablicy oraz manualne wprowadzenie elementów, z których będzie składała się tabela. Całość działa przy użyciu pętli for oraz jest zabezpieczona przed wprowadzeniem niepoprawnej wartości rozmiaru tabeli (Program nie podejmie próby stworzenia tabeli o rozmiarze zdefiniowanym za pomocą liczb ujemnych, czy liter etc. Wyskoczy komunikat i wrócimy do menu)
stworzenieb()	Funkcja, zawiera odpowiedni algorytm zastosowany do tablicy A. Zadaniem stworzenieb() jest także utworzenie tablicy oraz wyświetlenie powstałej tabeli B.
wczytaj_z_pliku()	Funkcja w której tworzymy obiekt „ifstream” oraz używamy kombinacji metod oraz funkcji w celu przeniesienia danych ze strumienia, do tablicy A. Dokładniej rzecz ujmując tworzymy dodatkową tablicę, zdolną zmieścić do 500 elementów, pomocniczy tabelę znaków znaną również jako typ string, do tymczasowego przechowywania danych. Sprawdzając przy tym, czy funkcja good nie zwraca żadnych błędów związanych z odczytem pliku. By później użyć także funkcji getline pobierającej każdą pojedynczą linię z importowanego pliku tekstowego. Ta funkcja oprócz sprawdzania, czy jest już na samym końcu pliku(getline zwraca wartość 0 kiedy skończy się jej możliwość pobierania danych, w przeciwnym przypadku zwraca wartość 1) kopiuje dane z każdego pojedynczego wiersza do naszego pomocniczego stringa. W ciele tej funkcji znajduje

	się przypisanie kolejnych i skonwertowanych na typ int elementów do kolejnych okienek tabeli pomocniczej, oraz inkrementacja zmiennej odpowiadającej za późniejsze przekazanie informacji do zmiennej w-jak duża jest ta tablica.
zapisz_do_pliku()	Podobnie jak powyżej, lecz w tym przypadku otwieramy „plik” w trybie tylko do zapisu, a następnie przy pomocy pętli for linijka po linijce przekazujemy dane do zewnętrznego pliku txt o nazwie "dane.txt". Który zostaje nadpisany lub stworzony (jeśli wcześniej nie istniał w tym katalogu).

Tabela 1 Spis zastosowanych funkcji w programie

7. Wnioski i podsumowanie projektu

Umiejętności programistyczne wykorzystane w podczas tworzenia projektu:

- Znajomość tablic,
- Wykorzystywanie pętli for oraz while,
- Zastosowanie warunku wielokrotnego wyboru switch ... case,
- Implementacja odpowiednich bibliotek, współpracujących z funkcjami programu (fstream, cstdlib, string)
- znajomość ogólnej składni języka c++,
- podstawowa umiejętność pisania pseudokodu,
- umiejętność ilustrowania procesu programistycznego programu w postaci schematów blokowych,

Ogólne wnioski:

Podział programu na funkcje, które posiadają pojedyncze zadania znacząco ułatwia lokalizowanie błędów, ale również umożliwiając rozbudowywanie programu w łatwy sposób bez konieczności powtarzania kodu.

Wprowadzenie tablicy dynamicznej (rezygnując poprzez to z użycia tablicy statycznej) umożliwia nam swobodne manipulowanie rezerwowanymi danymi w zakresie tworzenia tablic. Czynność ta pozwala nam uniknąć zmartwień dotyczących konieczności pamiętania o zmianie wymiarów tablicy, w chwili gdy ilość elementów byłaby inna, niż ostatnio implementowana. Dzięki czemu oszczędzamy cenny czas oraz unikamy błędów podczas tworzenia programu.

Choć w zakresie tablic, możemy pamiętać także o tym, iż istnieje możliwość obliczenia wielkości tablicy B także, za pomocą wzoru:¹
$$\text{wielkość_tablicyB} = (\text{sizeof}(\text{tablicaA}) / \text{sizeof}(\text{typ\ pojennika\ tablicy\ A}))$$
[int w naszym przypadku] (wikipedia, 2019)

Projektując aplikację powinniśmy również pamiętać o uwzględnieniu ograniczeń, w zakresie każdej możliwej opcji, dotyczącej wprowadzanych danych w polu wyboru użytkownika, w celu uniknięcia błędów programu (m.in. wprowadzenie ujemnej wartości podczas ustalania wymiarów tablicy.)

W programie unikaliśmy zastosowanie polskich liter, przez wzgląd na formę otwieraną w konsoli. Komputery miewają różne rodzaje kodowania. Są już różnice pomiędzy aktywną stroną kodową 850, a 852. Zatem prosta forma została wprowadzona, aby uniknąć „krzaczków”.² (Zelent, 2014)

Należy pamiętać o zwalnianiu pamięci kiedy przewidujemy, iż zapisane dane nie będą nam już dłużej potrzebne, podczas użytkowania programu. Wszystko to, aby uniknąć m.in. wycieku pamięci. „Skutkiem każdego wycieku jest nadmierne zużycie pamięci operacyjnej przez aplikację. W zależności od tego, ile jej tracimy to zjawisko jest mniej lub bardziej groźne. Zauważymy, że jeżeli jakaś funkcja, która nie zwalnia pamięci alokowanej dynamicznie jest wywoływana setki lub tysiące razy w ciągu każdej godziny działania aplikacji, to po pewnym

¹ Źródło: https://pl.wikibooks.org/wiki/C/Tablice_-_wi%C4%99cej

² Źródło: Mirosław Zelent, Kurs C++

czasie program może zużywać kilkukrotnie więcej pamięci niż faktycznie jest mu to potrzebne.”
(Kołodziej, 2015)³

Pozostawienie tablic jako tych do których można mieć dostęp z każdego miejsca programu sprawdza się w przypadku mojego programu, jednak przy większym projekcie programistycznym różne funkcje mogłyby zmieniać wprowadzone dane, zatem klasa publiczna nie jest wskazana do implementacji, m.in. we wspomnianym przypadku.

Słowem podsumowania możemy uznać, iż program stworzony na potrzeby projektu według zaleceń został przetestowany w celu sprawdzenia działania funkcji oraz występowania ewentualnych błędów, wynikających z niepoprawnych danych wejścia, czego skutkiem są stosowne komunikaty, wyjaśniające użytkownikowi jakie kroki powinien podjąć, w celu poprawnego działania konsoli.

³ Źródło: Kacper Kołodziej, 2015

Bibliografia

Kołodziej, K. (2015, styczeń 17). <https://kacperkolodziej.pl>. Pobrano z lokalizacji <https://kacperkolodziej.pl/programowanie/zwalczanie-wyciekow-bledow-pamieci-cpp11.html>

wikipedia. (2019, grudnia 15). Pobrano z lokalizacji https://pl.wikibooks.org/wiki/C/Tablice_-_wi%C4%99cej

Zelent, M. (2014, maj 25). Pobrano z lokalizacji <https://miroslawzelent.pl/kurs-c++/polskie-znaki-konsola-terminal-windows-linux-macos/>

Spis ilustracji

OBRAZ 1 KOD VOID WPISZ()	14
OBRAZ 2 KOD VOID WCZYTAJ_Z_PLIKU	14
OBRAZ 3 KOD VOID STWORZENIEB()	15
OBRAZ 4 KOD VOID ZAPISZ_DO_PLIKU	15
OBRAZ 5 KOD VOID FUNCKCJA()	15
OBRAZ 6 OBRAZ KONSOLI. TEST PROGRAMU NR 1. BŁĄD. NIEPRAWIDŁOWY WYBÓR.	16
OBRAZ 7 OBRAZ KONSOLI. TEST PROGRAMU NR 2. IMPORT ELEMENTÓW TABLICY Z PLIKU „DANE.TXT”	16
OBRAZ 8 OBRAZ KONSOLI. TEST PROGRAMU NR 3. BŁĄD. UJEMNA WARTOŚĆ ZADEKLAROWANEJ TABLICY.	17
OBRAZ 9 OBRAZ KONSOLI. TEST PROGRAMU NR 4. BŁĄD PODCZAS PRÓBY IMPORTU ELEMENTÓW TABLICY Z PLIKU „DANE.TXT”. BRAK WSPOMNIANEGO DOKUMENTU W FOLDERZE PROJEKTU.	17
OBRAZ 10 OBRAZ KONSOLI. TEST PROGRAMU NR 6. BŁĄD IMPORTU TABLICY Z PLIKU.	18
OBRAZ 11 OBRAZ KONSOLI. TEST PROGRAMU NR 7. BŁĄD. NIEPRAWIDŁOWA OPCJA WYBORU.	18
OBRAZ 12 OBRAZ KONSOLI. TEST PROGRAMU NR 8. BŁĄD. ZBYT DUŻA ZADEKLAROWANA WARTOŚĆ TABLICY.	19
OBRAZ 13 OBRAZ KONSOLI. TEST PROGRAMU NR 9. WYNIKI ZAPISANE W DOKUMENCIE "DANE.TXT" PRZED OPERACJĄ NADPISU.	19
OBRAZ 14 OBRAZ KONSOLI. TEST PROGRAMU NR 10. WYNIKI ZAPISANE W DOKUMENCIE "DANE.TXT" PO OPERACJI NADPISU.	20
OBRAZ 15 OBRAZ KONSOLI. TEST PROGRAMU NR 11. WPROWADZENIE RĘCZNE WPROWADZENIE DANYCH ORAZ ZAPIS DO TABLICY B.	21

Spis schematów blokowych

SCHEMAT BLOKOWY 1 FUNKCJA WYPISZ()	3
SCHEMAT BLOKOWY 2 FUNKCJA WCZYTAJ_Z_PLIKU (CZ.1)	4
SCHEMAT BLOKOWY 3 FUNKCJA WCZYTAJ_Z_PLIKU (CZ.2)	5
SCHEMAT BLOKOWY 4 FUNKCJA STWORZENIEB()	6
SCHEMAT BLOKOWY 5 FUNKCJA ZAPISZ_DO_PLIKU()	7
SCHEMAT BLOKOWY 6 FUNKCJA () (CZ.1)	8
SCHEMAT BLOKOWY 7 FUNKCJA () (CZ.2)	9