Michał Walkiewicz

Nr albumu: \*\*\*\*\*

Grupa zajęciowa: \*\*\*\*\*\*

Raport-projekt BAL

Wylosowałem temat o nr. 23

Brzmi on:

Proszę napisać algorytm, który dla dwuwymiarowej tablicy o wartościach całkowitych wyznacza numer wiersza o maksymalnej i nieparzystej wartości sumy elementów wiersza. Jeśli nie ma rozwiązania to wartością indeksu jest -1.

Niżej w zadaniu podany jest przykład.

Nie będę go przytaczał. Przyjąć należy:

Dane wejściowe: tablica dwuwymiarowa o wartościach całkowitych.

Dane wyjściowe: numer indeksu wiersza o najwyższej nieparzystej sumie wiersza z tablicy (dane wejściowe). Brak rozwiązania indeks -1.

1. Opis słowny algorytmu.
2. Początek programu.
3. Wczytaj dane wejściowe – tablica dwuwymiarowa o wartościach całkowitych.

Wczytaj zmienne pomocnicze oraz pustą tablicę jednowymiarową do przechowywania sum (suma[k]).

1. Iteracje w pętlach w celu obliczenia sum poszczególnych wierszy:

Dla licznika [i] który ma wartość początkową 0, licznika mniejszego od rozmiaru wierszy tablicy dwuwymiarowej zwiększaj licznik co 1 powtarzaj:

\*przypisz tablicy jednowymiarową suma[i]=0

Dla licznika [j] który ma wartość początkową 0, licznika mniejszego od rozmiaru kolumn tablicy dwuwymiarowej zwiększaj licznik co 1

powtarzaj:

\*suma[i]=suma[i]+tablica dwuwymiarowa[rozmiar wierszy]

[rozmiar kolumn]

Zakończ iteracje

Zakończ iteracje

1. Porównanie sum:

Dla licznika [i] który ma wartość początkową 0, licznika mniejszego od rozmiaru wierszy tablicy dwuwymiarowej zwiększaj licznik co 1 powtarzaj:

\*Jeżeli suma[i] będzie większa od zmiennej pomocniczej „naj=0”

**i** „&&-iloczyn logiczny” suma[i] będzie podzielna modulo 2 różna od 0 „!=”

\*zmiennej naj przypisz wartość suma[i]

\*zmiennej indeks przypisz obecne i

Zakończ iteracje

1. Jeśli brak sum nieparzystych:

Jeżeli zmienna indeks będzie równa -1 wyświetl -1;

W przeciwnym wypadku wyświetl indeks.

1. Zakończ program

2.Schemat blokowy z omówieniem:

i=i+1;

N

N

N

T

N

T

T

N

T

T

Wy: indeks

Wy: -1

indeks==-1

i=i+1

naj = suma[i];

indeks = i;

suma[i]>naj &&suma[i]%2!=0

i<n

i=0

j<m

j=j+1

suma[i]=suma[i]+tablica[i][j]

suma[i]=0; j=0;

i<n

We: tablica[n][m]

naj=0, indeks=-1, k=0

suma[k], i=0, j=0;

1. Omówienie schematu blokowego.

Sprawa omówienia jest dużo prostsza. Wczytuję tablice **dwuwymiarową** o wartościach całkowitych. Znam rozmiar tablicy.

Tablica ma **n-wierszy**, **m-kolumn**. Zmiennej naj przypisuję **wartość 0**, posłuży mi ona do warunku przy porównywaniu sum. Zmienna k ustawiam na 0.

Algorytm najpierw pobiera informacje o ilości wierszy n i kolumn m. W głównej pętli badam wiersze, w zagnieżdżonej- kolumny. Każdą sumę zapisuję pod **suma[i].** Obliczenie jednej sumy wiersza to 1 wykonanie głównej pętli. Zmienne „i” i „j” to liczniki. Warunek **„j<m”** bada czy wszystkie kolumny w 1 wierszu są sumowane. **KAŻDY** wynik wiersza zapisuje odpowiednio suma[0]=wartość, suma[1]=wartość. Wpisuje to od razu do tablicy jednowymiarowej o indeksie początkowym 0. Po przebrnięciu przez pętle zeruje „i” i sprawdzam czy **,,i<n**”. Istnieje sytuacja, że ktoś poda np. n=0 (pustą liczbę wierszy) .

Następnie kluczowy warunek. Sumy zapisane w **tablicy jednowymiarowej** są badane w pętli warunkiem. Warunek sprawdza czy **suma jest nieparzysta** **i większa on zmiennej naj.** Naj przypisuje wartość konkretnej sumy z tablicy jednowymiarowej w 1st. Przypadku suma[0]=wartość. Przypisuję wartość i tej zmiennej. Teraz wiadomo czemu zastosowałem zmienną **indeks -zapamiętuje ona indeks wiersza przy każdej iteracji.** Czynność powtarzam aż przebrnę przez wszystkie sumy.

Jeżeli brak jest sumy nieparzystej na wyjściu jest -1, w przeciwnym wypadku **indeks wiersza największej sumy nieparzystej.**

W skrócie:

\*każdą kolumnę pierwszego wiersza **sumuję**. Sumę wpisuję do tablicy suma[i]

\*powtarzam aż przejdę przez wszystkie wiersze;

\*mam obliczone sumy wszystkich wierszy w suma[i];

\*wyciągam tą jedną największą nieparzystą z suma[i] pamiętając jej indeks;

\*wyświetlam wynik-indeks.

1. Zapis algorytmu w pseudokodzie.

Rozpocznij program.

We: tablica[n][m]

naj=0, indeks=-1, k=0

suma[k], i=0, j=0;

Dla i = 0; i < n;i=i+1 powtarzaj

suma[i] = 0

Dla j = 0; j < m; j=j+1 powtarzaj

suma[i]= suma[i]+tablica[i][j]

zakończ iteracje

zakończ iteracje

Dla i=0; i<n; i=i+1 powtarzaj

Jeśli suma[i]>naj i suma[i]%2!=0

naj = suma[i]

indeks = i

zakończ iteracje

Jeśli (indeks == -1)

Wy :"indeks wiersza wynosi " -1

w.p.w

Wy:” indeks wiersza wynosi " indeks

Zakończ program

1. Symulacja działania dla kilku przykładowych danych wejściowych.

Tablica 2-n- wiersze, 1-m- kolumny

We:

|  |
| --- |
| 3 |
| 2 |

We: tablica[2][1]

naj=0, indeks=-1, k=0

suma[k], i=0, j=0;

czy 0<2 tak

suma[0]=0; j=0;

czy 0/1 tak

suma[0]=suma[0]+tablica[0][0]

suma[0]=3

j=j+1

j=1

czy 1<1 nie

czy 1<2 tak

suma[1]=0; j=0;

czy 0/1 tak

suma[1]=suma[1]+tablica[1][0]

suma[1]=2

j=j+1

j=1

czy 1<1 nie

czy 2<2 nie

//„Sumy obliczone”

//,,Badanie sum”

i=0

czy 0<2 tak

czy 3>naj i mod2 jest różne od 0 ? tak

naj=3

0=indeks

i=i+1

i=1

czy 1<2 tak

czy 2>naj i mod2 jest różne od 0 ? nie, 3 jest większe od 2

uciekam z pętli wyświetlam wynik

indeks 0

Tak wygląda badanie sum. Analogicznie dla n=4, m=3

|  |  |  |
| --- | --- | --- |
| 3 | 2 | 5 |
| 4 | 4 | 4 |
| 1 | 0 | 1 |
| 1 | 4 | 1 |

Sum[0]=10

Sum[1]=12

Sum[2]=2

Sum[3]=6

Indeks -1

1. Zapis algorytmu w języku programowania

Przystępując do wykonania tego projektu niemalże z marszu pomyślałem by zapisać to w C++. Cały ten projekt od tego zacząłem-nawyk zadań maturalnych – algorytm w głowie – mało czasu – pisze kod. Z racji zastosowanego algorytmu suma[i]=0 będę musiał się posłużyć głownie kodem właśnie C++.

Co zrobiłem dodatkowo ?

Wpisuję pętlami for do tablicy losowe wartości z przedziału

rand()% (5+1); <0;5>

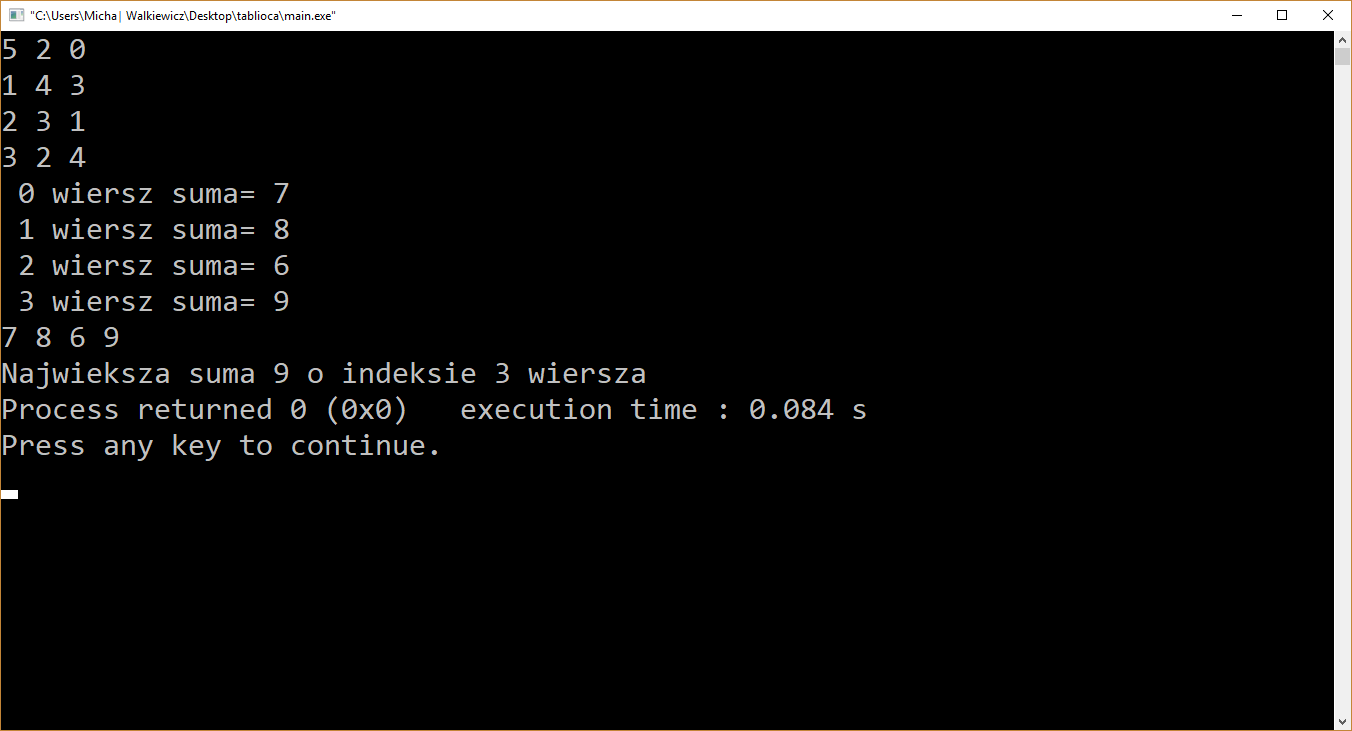
Wyświetlam tablice, wyświetlam sumy i wiersze, wyświetlam indeks.

Cały algorytm powstał dokładnie na tym kodzie. Co można jeszcze zrobić? Z racji tego, że program napisałem liniowo, można dodatkowo zastosować funkcje dla każdych z etapów.

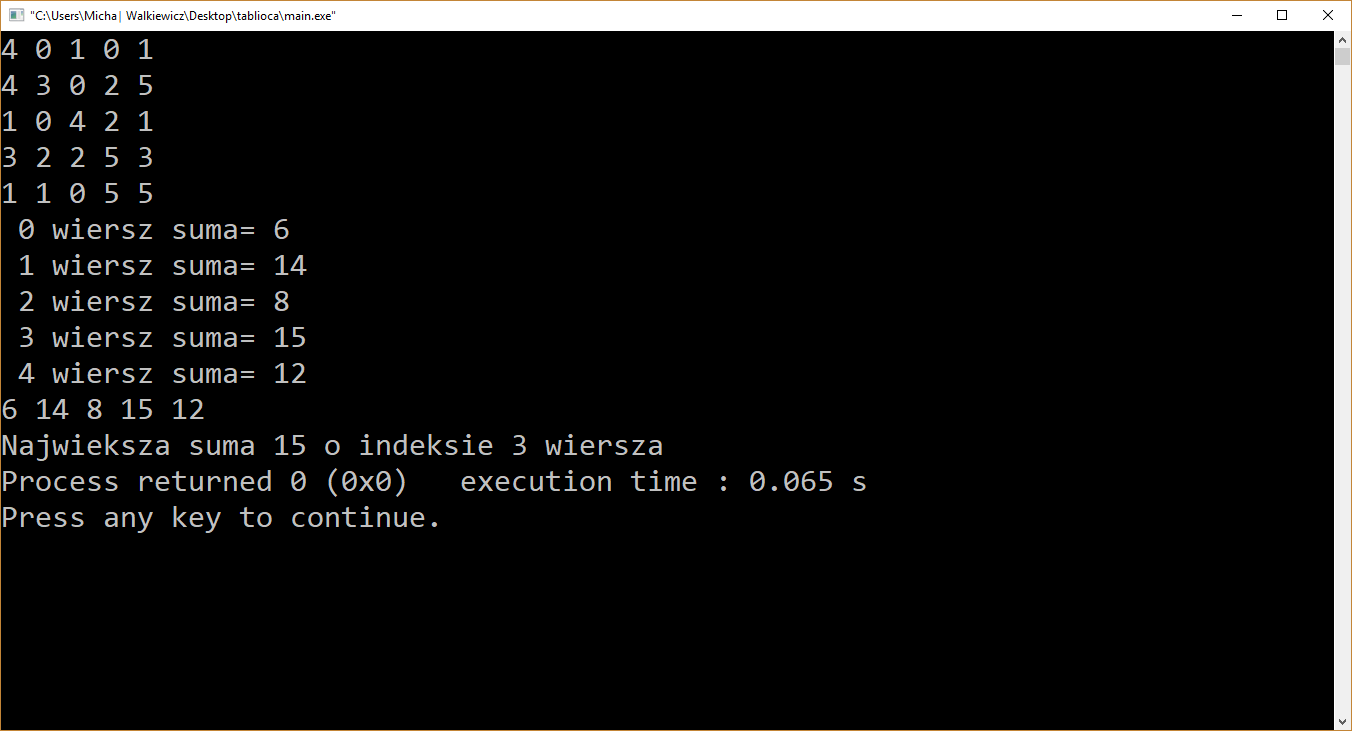
Poświęciłem na to sporo czasu. Kod programu, projekt i aplikacja konsolowa będzie zamieszczona w tym projekcie. Hiperłącze: <sumy.cpp>

Uważam, że zapis w Scilabie na tablicach statycznych byłby mało rozbudowany. A jak wspominałem wcześniej zastosowałem kodowanie array[][].

Program można zmieniać poprzez edycję n i m- rozmiaru tablicy. Przykładowe screeny: dla n=4; m=3;

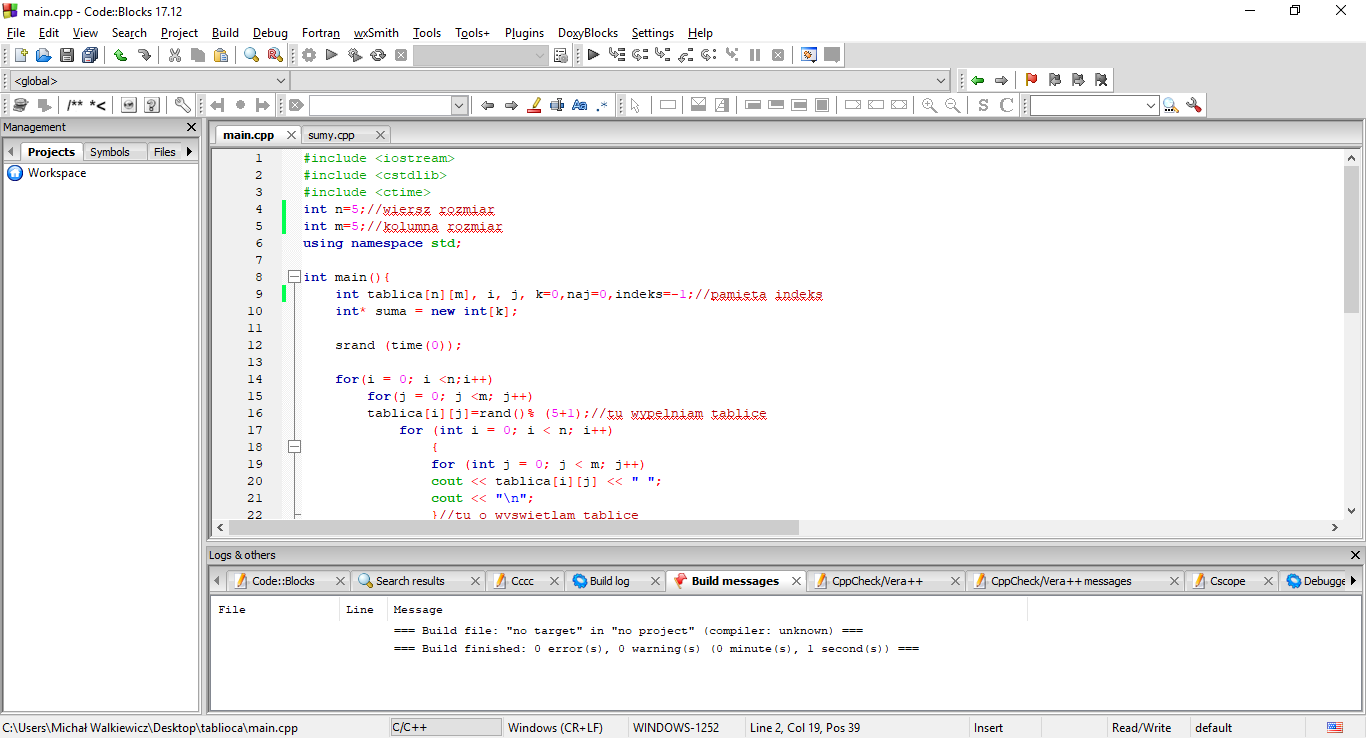


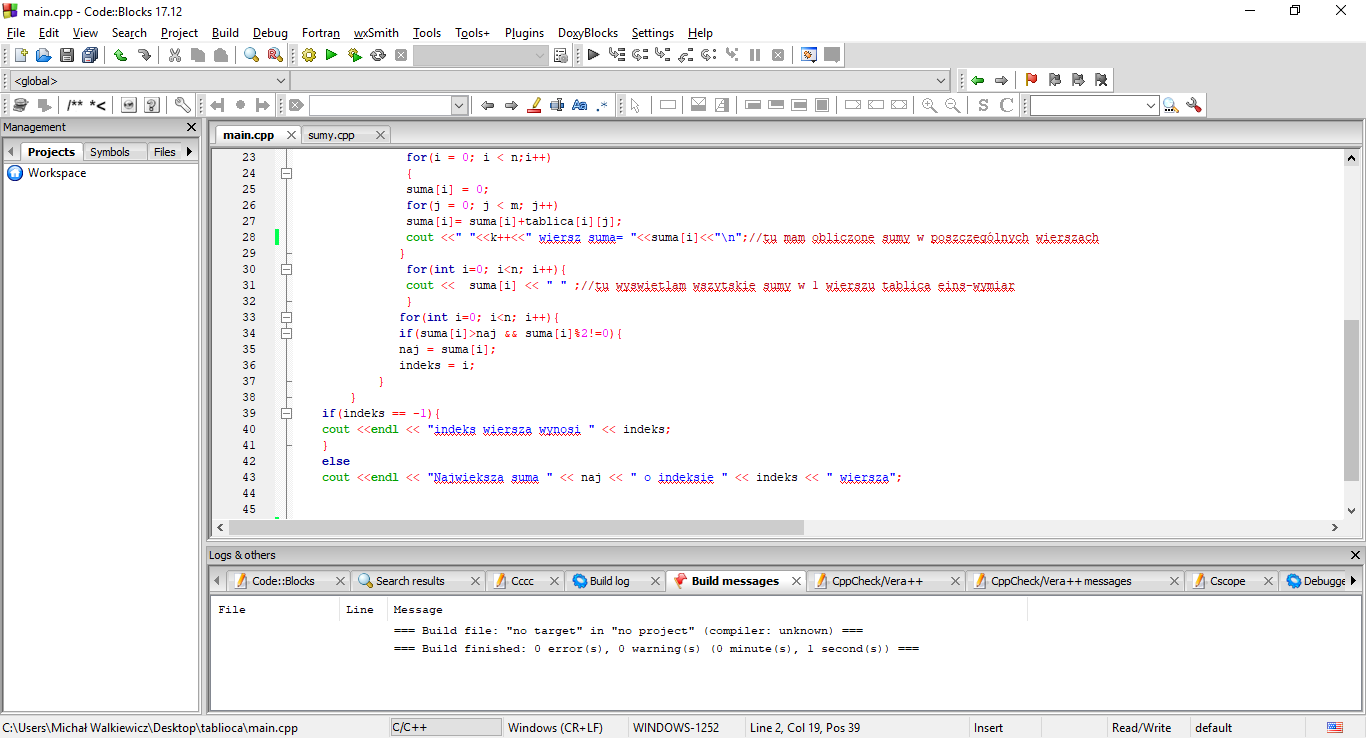
Tablica kwadratowa n=5;m=5;

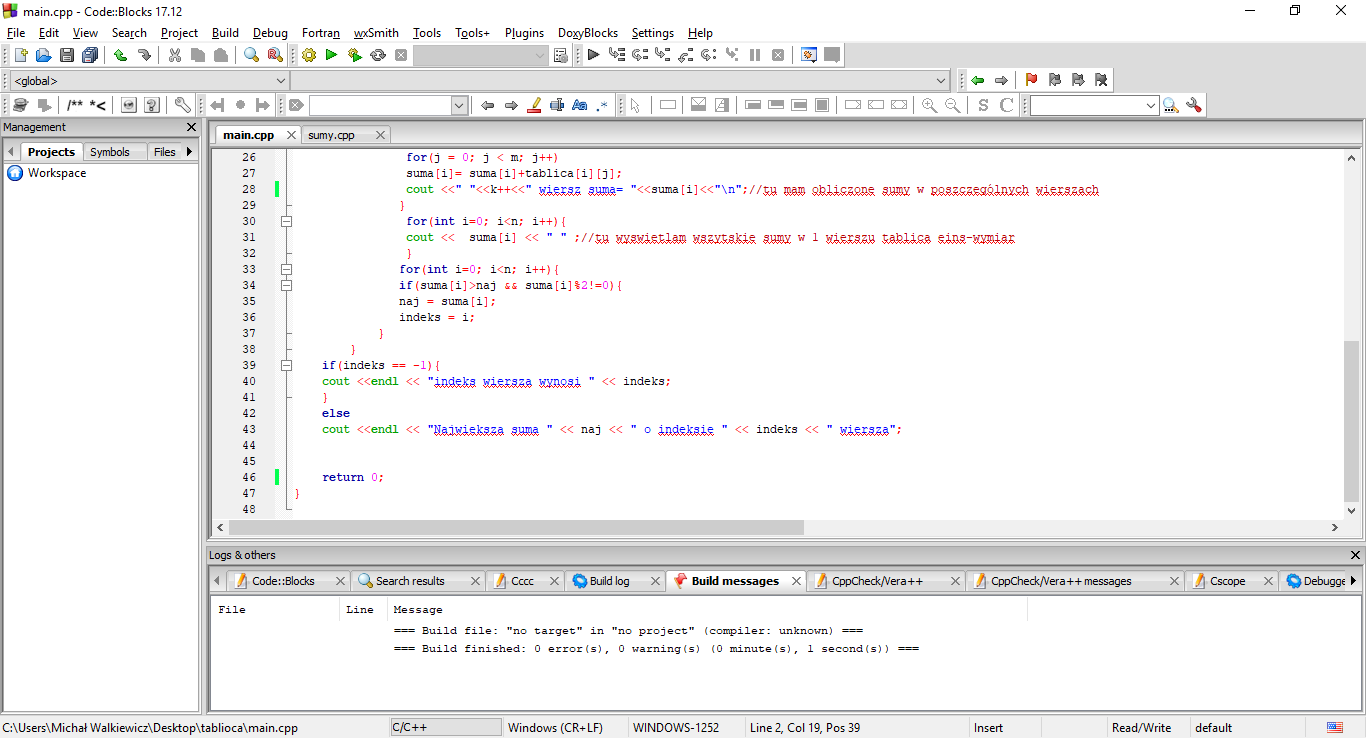


Kod prgramu: Code::blocs 17.12 biblioteki

<iostream> <cstdlib> <ctime>







1. Oszacowanie złożoności czasowej.

Oszacuję to na podstawie schematu blokowego, gdy oszacuje czas dla programu c++ wystąpi złożoność liniowa proporcjonalna do wprowadzonego n i m

Pętla wierszowa 1

Operator przypisania suma[i]=0; 1

Pętla zagnieżdżona w pętli wierszowej (sumowanie kolumn) 1

Przypisanie nowych sum 1… +7 reszta

Gdzie n to wejście tablicy nxm

Dla tego programu jest to złożoność obliczeniową rzędu 0(nlog(n))

1. Będę testował na programie c++ gdzie jest generowanie tablicy pętlami i dla przykładowych danych wejściowych obliczymy czas realizacji.

Dane wejściowe w moim programie są losowe, zrobiłem to specjalnie gdyż podczas obrony projektu można ładnie pokazać wyniki na różnych danych testowych.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | 0 | 3 | 5 | 7 | 11 | 20 | 30 | 100 |  |  |  |  |  |  |
| m | 0 | 2 | 5 | 6 | 11 | 20 | 30 | 100 |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| czas | 0,034 | 0,035 | 0,044 | 0,068 | 0,074 | 0,189 | 0,331 | 3,331 |  |  |  |  |  |  |
|  |  |  |  |  |  |  | | | | | | | | |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Dziękuję za poświęcony czas na przeczytanie.

Wykonał Michał Walkiewicz

