**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i struktury danych | **Zadanie** | 1\_05 |
| **Autor** | Kamil Borkowski | **Grupa** | WCY22IY1S1 |
| **Temat** | Maksymalna i minimalna wartość wektora X | | |

1. Treść

Zaimplementuj algorytm wyszukujący maksymalną i minimalną wartość wektora X zawierającego n liczb.

* 1. Metoda realizacji

Podajemy ilość elementów wektora i uzupełniamy go losowymi liczbami całkowitymi. Porównujemy kolejno wartości wektora w poszukiwaniu wartości minimalnej i maksymalnej

* 1. Założenia / ograniczenia dotyczące danych:
     1. Dane wejściowe *(opis)*

Wartości wektora – wartości losowe między 1 a 6

* + 1. Dane wyjściowe *(opis)*

Wartość minimalna – wyprowadzana na ekranie

Wartość maksymalna – wyprowadzana na ekranie

1. Realizacja
   1. Algorytm *(lista kroków lub schemat blokowy)*

a)Podajemy długość wektora

b)Uzupełniamy wektor losowymi wartościami w pętli i=0 do i<długość wektora

c)W pętli i=0 do i<n porównujemy kolejne wartości wektora w poszukiwaniu wartości najmniejszej i największej

* 1. Kod źródłowy

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

int main()

{

int n,mini,maks;

cin>>n;

int\* V = new int[n];

srand(time(NULL));

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

{

V[i]=(rand()%6)+1;

cout << V[i] << "\n";

}

mini = V[0];

maks = V[0];

for(int i=1;i<n;i++)

{

if(mini>V[i])mini=V[i];

if(maks<V[i])maks=V[i];

}

cout<<"minimalna wartosc: "<<mini<<endl;

cout<<"maksymalna wartosc: "<<maks<<endl;

return 0;

}

* 1. Dane wejściowe *(ekran uruchomienia programu)*

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

* 1. Dane wyjściowe *(ekran uruchomienia programu)*

Obraz zawierający tekst, zrzut ekranu, Czcionka

Opis wygenerowany automatycznie

1. Wnioski *(złożoność asymptotyczna algorytmu przy użyciu notacji* ***O lub innej i inne wnioski)***

Złożoność obliczeniowa programu wynosi O(n), ponieważ posiada pętle, które wykonają się n razy