**SPRAWOZDANIE**

|  |  |  |  |
| --- | --- | --- | --- |
| **Przedmiot** | Algorytmy i struktury danych | **Zadanie** | 3\_01 |
| **Autor** | Kamil Borkowski | **Grupa** | WCY22IY1S1 |
| **Temat** | Sito Erastotenesa | | |

1. Treść

Zaimplementuj algorytm wyznaczania liczb pierwszych z zadanego przedziału [2,n].(Użyj listy jednokierunkowej zaimplementowanej wskaźnikowo)

Metoda realizacji

Wypełniamy wszystkie elementy listy liczbami od 2 do n. Usuwamy te elementy z listy, które nie są liczbami pierwszymi według sita Erastotenesa.

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

Górna granica przedziału – n – wczytywana z klawiatury

* + 1. Dane wyjściowe *(opis)*

Liczby pierwsze w przedziale [2,n] – wyświetlone na ekranie

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

Deklaracja struktury listy

Uzupełnienie listy poprzez tworzenie nowych elementów listy

Usunięcie elementów nie będącymi liczbami pierwszymi

Wyświetlenie wszystkich elementów listy

* 1. Kod źródłowy

//================================================

//Zadanie 3\_01 jezyk C++

//Sito Erastotenesa

//WCY21IY1S1 Borkowski Kamil

//================================================

#include <iostream>

#include <cstdlib>

using namespace std;

struct Node

{

int value;

Node\* next;

};

int main()

{

int n,i=2,j=2,k=1;

cout<<"Podaj zakres gorny liczb pierwszych: ";

cin>>n;

cout<<"Liczby pierwsze w przedziale [2,"<<n<<"] to:"<<endl;

Node\* head = NULL; //pierwszy element listy, poczatkowo pusty

Node\* current = head; //pomocniczy element listy, sluzacy do poruszania sie po liscie

for(;i<=n;i++)//Poczatkowe uznanie wszystkich liczb z zakresu jako pierwsze i wpisanie ich do lsity

{

Node\* new\_node = (Node\*)malloc(sizeof(Node));

new\_node->value = i;

new\_node->next = NULL;

if(head != NULL)//tworzymy kolejne elementy jesli mamy juz element pierwszy

{

current->next = new\_node;

current = new\_node;

}

else//jesli pierwszy element listy jest NULL przypisyjemy pod niego nowo utworzony element

{

head = new\_node;

current = new\_node;

}

}

for(;j\*j<=n;j++)

{

current=head;

Node\* prev = NULL;//pomocniczy element listy trzymajacy dane poprzedniego elementu

while(current != NULL)

{

int num = current->value;

if(num == j)

{

prev = current;

current = current->next;

}

else

{

if(num % j == 0)//usuwanie liczb podzielnych przez j

{

if(prev != NULL)

{

prev->next = current->next;

}

else

{

head = current->next;

}

free(current);

current = prev->next;

}

else

{

prev = current;

current = current->next;

}

}

}

}

current = head;

while(current != NULL)

{

cout << current->value <<'\t';

if(k==10)

{

cout<<endl;

k=0;

}

Node\* pom = current;

current = current->next;

free(pom);//zwolniamy pammiec

}

return 0;

}

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

Obraz zawierający tekst, zrzut ekranu, oprogramowanie, Oprogramowanie multimedialne

Opis wygenerowany automatycznie

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

Obraz zawierający tekst, zrzut ekranu, wyświetlacz, oprogramowanie

Opis wygenerowany automatycznie

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

Złożoność obliczeniowa programu wynosi: O(n), ponieważ najbardziej złożone obliczeniowo jest wykonanie utworzenie wszystkich elementów lsity.