Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

Информатики и радиоэлектроники”

Кафедра программного обеспечения информационных технологий

Практическая работа № 5

Вариант № 30

“Метрики сложности потока управления программ”

Проверила: Выполнили:

Болтак С.В. ст. гр. 221703

Лукашевич А. Д.

Веркович Е. В.

Минск 2023

#include <iostream>

using namespace std;

int check\_prime(int);

int main() {

int n1, n2;

bool flag;

cout << "Enter two positive integers: ";

cin >> n1 >> n2;

// swapping n1 and n2 if n1 is greater than n2

if (n1 > n2) {

n2 = n1 + n2;

n1 = n2 - n1;

n2 = n2 - n1;

}

cout << "Prime numbers between " << n1 << " and " << n2 << " are:\n";

for (int i = n1 + 1; i < n2; ++i) {

// if i is a prime number, flag will be equal to 1

flag = check\_prime(i);

if (flag)

cout << i << ", ";

}

return 0;

}

// user-defined function to check prime number

int check\_prime(int n) {

bool is\_prime = true;

// 0 and 1 are not prime numbers

if (n == 0 || n == 1) {

is\_prime = false;

}

for (int j = 2; j <= n / 2; ++j) {

if (n % j == 0) {

is\_prime = false;

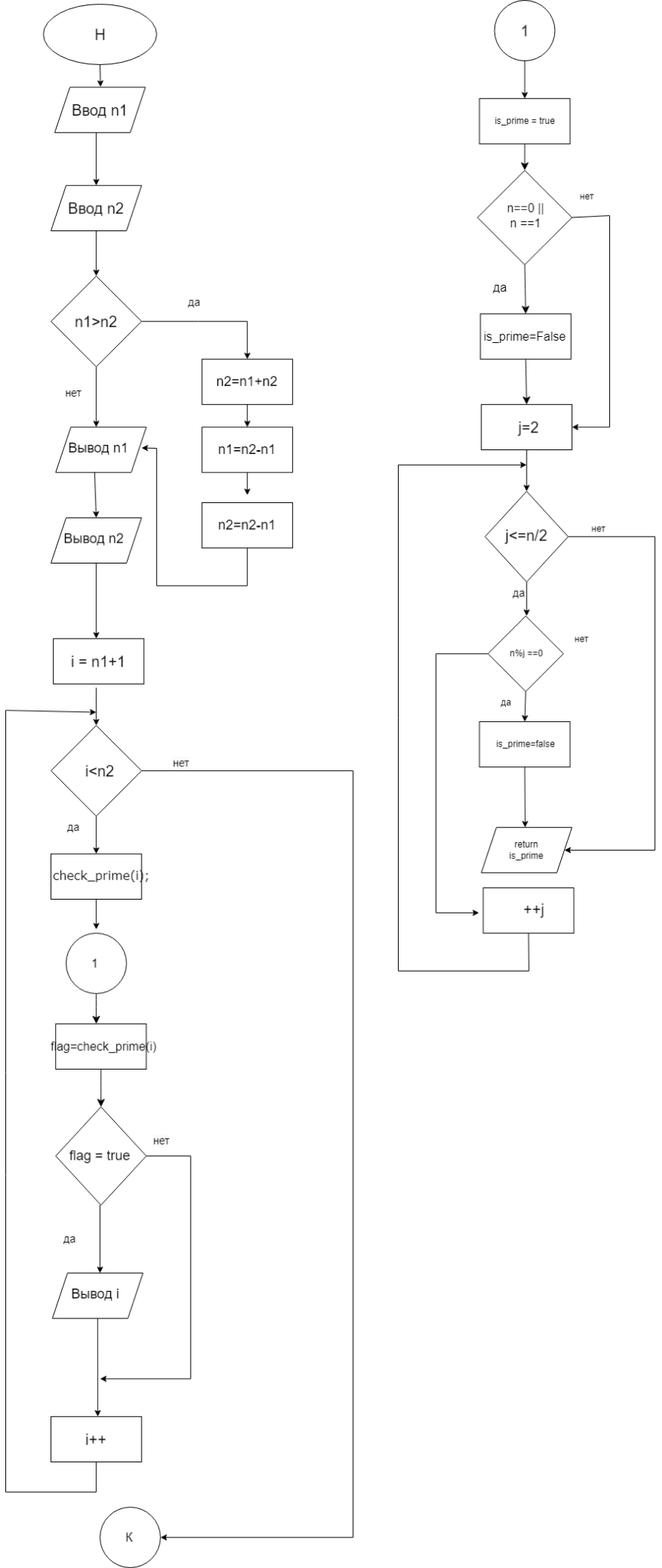
break;

}

}

return is\_prime;

}

Рис. 1

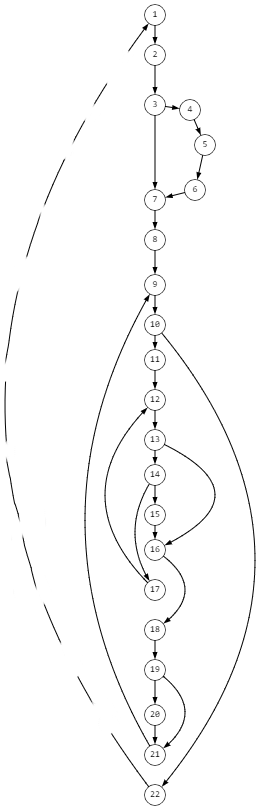


Рис. 2 Граф метрики Маккейба

Метрика Маккейба:

*Z(G) = e - v + 2p*

е — число дуг ориентированного графа G равно 26

ʋ — число вершин равно 22

р — число компонентов связности графа равно 1

Z= 26 - 22 + 2 = 6

1 путь: 1-2-3(нет)-8-9-10(нет)-22

2 путь: 1-2-3(да)-4-5-6-7-8-9-10(да)-11-12-13(нет)-16(нет)-18-19(нет)-21-9-10(нет)-22

#include <iostream>

using namespace std;

int check\_prime(int);

int main() {

int n1, n2;

string ch;

bool flag;

cout << "Enter two positive integers: ";

cin >> n1 >> n2;

// swapping n1 and n2 if n1 is greater than n2

if (n1 > n2) {

n2 = n1 + n2;

n1 = n2 - n1;

n2 = n2 - n1;

}

cout << "Prime numbers between " << n1 << " and " << n2 << " are:\n";

for (int i = n1 + 1; i < n2; ++i) {

// if i is a prime number, flag will be equal to 1

flag = check\_prime(i);

switch (i) {

case 1:

cout << "i equals 1" << endl;

break;

case 2:

cout << "i equals 2" << endl;

break;

case 3:

cout << "i equals 3" << endl;

break;

default:

cout << "i not equal 1, 2 or 3" << endl;

break;

}

if (flag)

cout << i << ", " << endl;

}

return 0;

}

// user-defined function to check prime number

int check\_prime(int n) {

bool is\_prime = true;

// 0 and 1 are not prime numbers

if (n == 0 || n == 1) {

is\_prime = false;

}

for (int j = 2; j <= n / 2; ++j) {

if (n % j == 0) {

is\_prime = false;

break;

}

}

return is\_prime;

}

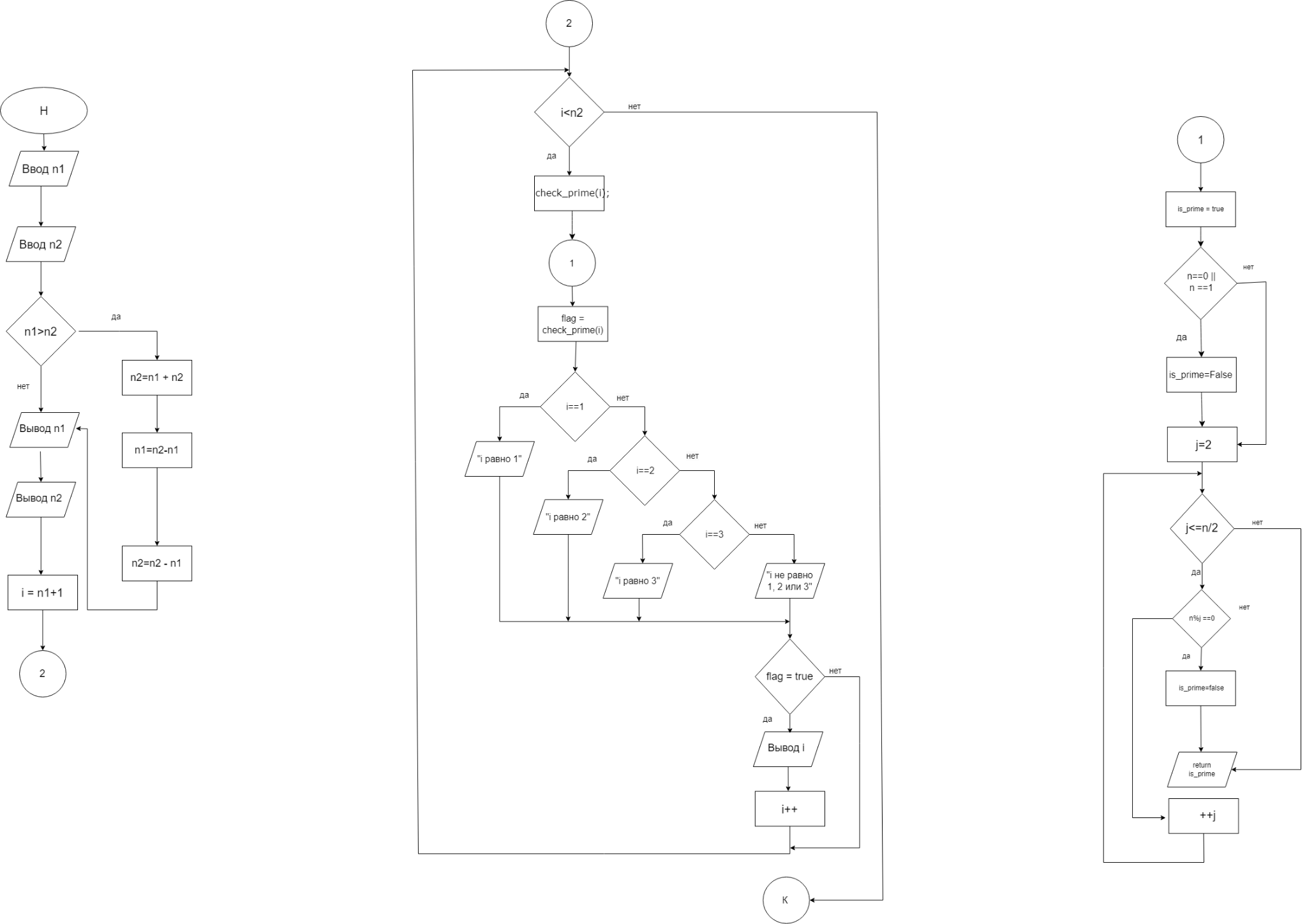


Рис. 3

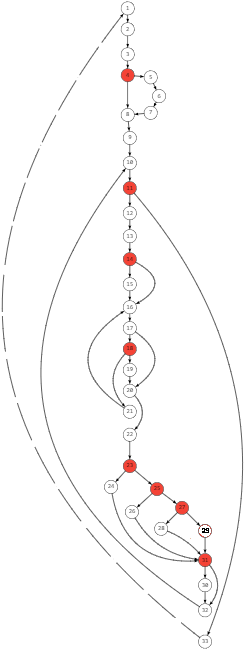


Рис. 4 Граф метрики Джилба

Метрика Джилба:

*CL* = 9 – количество условных операторов;

*cl* = 9 / 31 = 0,29 – отношение условных операторов к общему количеству операторов;

*CLI* = 3 – максимальный уровень вложенности условных операторов