Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

Информатики и радиоэлектроники”

Кафедра программного обеспечения информационных технологий

Практическая работа № 3

Вариант № 6

Название работы

“ **метрики сложности потока управления программ.**”

Проверила: Болтак С.В

Выполнили

Студенты

гр. 221703

Мартысюк М.Ю.

Корнееенко Е.Д.

Минск 2023

***Код программы***

#include <iostream>

using namespace std;

int i, pluse1, pluse2 , n, k, l[2], result = 1;

double \*a;

int main()

{

cout << "Vvedite razmer massiva: "; // Вывод сообщения

cin >> n; a = new double[n];

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

{

cout << "Vvedite " << i + 1 << " element massiva: ";

cin >> a[i];

cout << endl; // Вывод сообщения

}

cout << endl; // Перевод каретки

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

{

cout << a[i] << " ";

}

cout << endl; // Вывод сообщения

k = 0;

for (i = n - 1; i >= 0; i--)

{

if (a[i] > 0 && k < 2)

{

l[k] = i;

k++;

}

}

if (k == 0)

{

cout << "Polozhitelnih elemntov net"; return 1;

}

if (l[0] <= l[1])

for (i = l[0] + 1; i < l[1]; i++)

result \*= a[i];

else

for (i = l[0] - 1; i > l[1]; i--)

result \*= a[i];

cout << result; // Вывод сообщения

delete[]a;

return 0;

}

***Код программы(with switch)***

#include <iostream>

using namespace std;

int main() {

int i, n, k, l[2];

double\* a, result = 1.0;

cout << "Vvedite razmer massiva: ";

cin >> n;

a = new double[n];

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

cout << "Vvedite " << i + 1 << " element massiva: ";

cin >> a[i];

cout << endl;

}

cout << endl;

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

switch (i)

{

case 1:cout << a[i]; break;

case 2:cout << a[i] \* a[i]; break;

case 3:cout << a[i - 1]; break;

default: cout << a[i - 2];

}

}

cout << endl;

k = 0;

for (i = n - 1; i >= 0; i--)

{

if (a[i] > 0 && k < 2)

{

l[k] = i;

k++;

}

}

if (k == 0)

{

cout << "Polozhitelnih elemntov net"; return 1;

}

if (l[0] <= l[1])

for (i = l[0] + 1; i < l[1]; i++)

result \*= a[i];

else

for (i = l[0] - 1; i > l[1]; i--)

result \*= a[i];

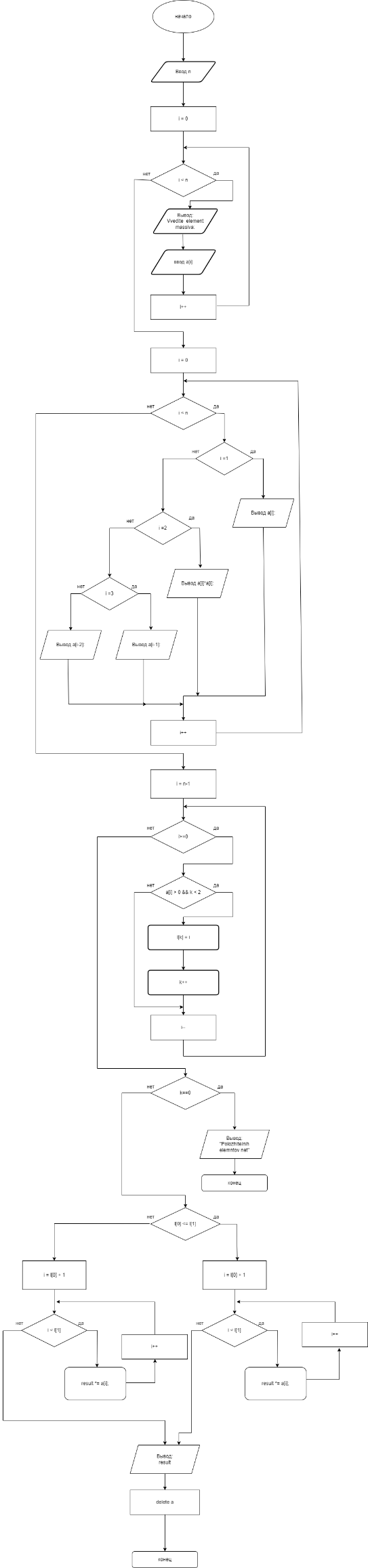
cout << result; // Вывод сообщения

delete[]a;

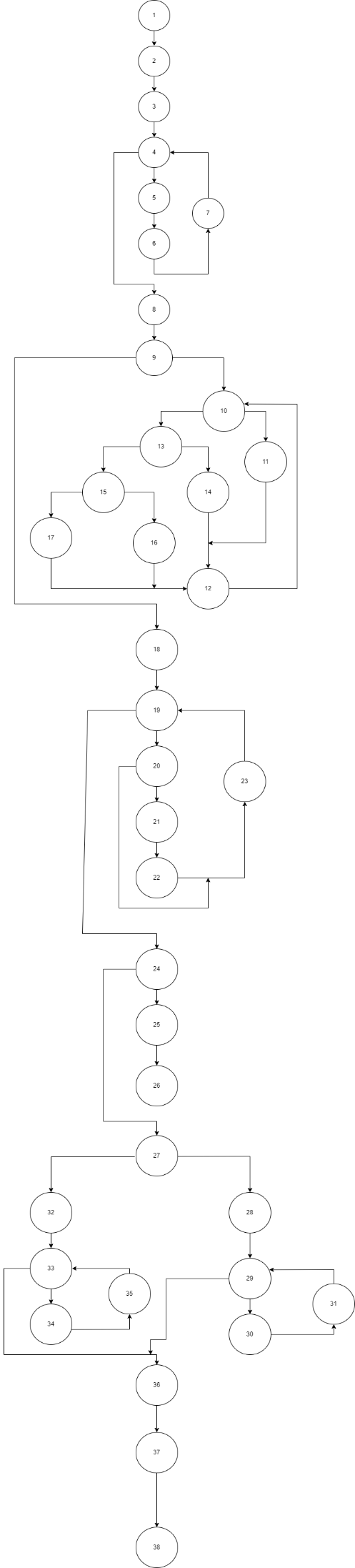
return 0;

}

***БЛОК-СХЕМА***



***ГРАФ***





**Метрика Маккейба**

1-2-3-4(нет)-8-9(нет)-18-19(нет)-24(нет)-27(нет)-32-33(нет)-36-37-38;

Количество вершин(e) =38;

Количество дуг(v) = 47;

Компонент связности(p) = 2;

Z(G) = e – v + 2p = 47– 38 + 2\*2 = 13(базисных независимых путей);

**Метрика Джилба**

CL=11(количество условных операторов, характеризующее абсолютную сложность программы)

сl=CL/(общее число операторов)=11/38=0.29(насыщенность программы условными операторами, характеризующее относительную сложность программы)

CLI=3(вершина 10 вложена в 9,вершина 13 вложена в 10, вершина 15 вложена в 13).Максимальный уровень вложенности условного оператора.