Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и

радиоэлектроники»

Кафедра метрологии и стандартизации

**Отчет по лабораторной работе №1**

**на тему: «Метрики сложности потока управления программ»:**

Вариант № 5

Выполнил: студентка гр. 953501

Голубович Ю. И.

Проверил: старший преподаватель

Грибович А.А.

Минск 2020

***Вариант индивидуального задания 5***

Ввести массивы А (20) и В (20). Образовать массив С (20) из элементов, встречающихся в обоих массивах. Остаток массива С заполнить нулями. Вывести исходные и результирующий массивы.

***Код программы:***

#include <iostream>

using namespace std;

void Output(int X[20]){

for (int i = 0; i < 20; i++)

cout << X[i] << ' ';

}

void Input(int X[20]){

for (int i = 0; i < 20; i++)

cin >> X[i];

}

int main(){

setlocale (LC\_ALL, "Russian"); // for Russian in console

int A[20] = {0}, B[20] = {0}, C[20] = {0}, i, j, k;

cout << "Введите массив A(20):\n";

Input(A);

cout << "Введите массив B(20):\n";

Input(B);

int x = 0;

for(i = 0; i < 20; i++){

for(j = 0; j < 20; j++){

if(A[i] == B[j]){

bool flag = 0; // to exclude repetitions in C(20)

for(k = 0; k < x + 1; k++)

if(C[k] == A[i])

flag = 1;

if(flag == 0){

C[x] = A[i];

x++;

}

break;

}

}

}

cout << "\nИсходный массив A(20): ";

Output(A);

cout << "\nИсходный массив B(20): ";

Output(B);

cout << "\nРезультирующий массив C(20): ";

Output(C);

return 0;

}

***Детализированная схема алгоритма по ГОСТ 19. 701-90***



***Описание блок-схемы:***

Переменные:

Массивы A(20), B(20) – исходные одномерные массивы элементов.

Массив C(20) – результирующий одномерный массив элементов.

Переменная flag – идентификатор для исключения повторяющихся элементов в результирующем массиве.

Переменная x – счетчик (индекс) в результирующем массиве С(20).

Переменные i, j, k – счетчики циклов.

Блоки:

Блок 1– точка входа в программу

Блок 2, 3, 4, 5, 6, 7, 18, 25 – объявление переменных

Блок 8 – вывод на экран строки «Введите массив A(20):» и перехода на новую строку

Блок 9, 14, 19, 36, 41, 46 – инициализация переменной i значением «0»

Блок 10, 15, 20, 37, 42, 47 – условие цикла «i < 20»

Блок 11 – ввод элемента A[i]

Блок 12, 17, 34, 39, 44, 49 – счетчик цикла, увеличение переменной i

Блок 13 – вывод на экран строки «Введите массив B(20):» и перехода на новую строку

Блок 16 – ввод элемента B[i]

Блок 21 – инициализация переменной j значением «0»

Блок 22 – условие цикла «j < 20»

Блок 23 – условие «A[i] == B[j]»

Блок 24 – счетчик цикла, увеличение переменной j

Блок 26 – инициализация переменной k значением «0»

Блок 27 – условие цикла «k < x + 1»

Блок 28 – условие «C[k] == A[i]»

Блок 29 – инициализация переменной flag значением «1»

Блок 30 – счетчик цикла, увеличение переменной k

Блок 31 – условие «flag == 0»

Блок 32 – вычисления C[x] = A[i]

Блок 33 – увеличение переменной x

Блок 35 – вывод на экран перехода на новую строку и строки «Исходный массив A(20): »

Блок 38 – вывод элемента A[i]

Блок 40 – вывод на экран перехода на новую строку и строки «Исходный массив B(20): »

Блок 43 – вывод элемента B[i]

Блок 45 – вывод на экран перехода на новую строку и строки «Результирующий массив С(20): »

Блок 48 – вывод элемента C[i]

Блок 50 – точка выхода из программы

***Расчёт метрики Маккейба для разработанного алгоритма и определение***

***базисных независимых путей:***

Метрика Маккейба предназначена для оценки трудоемкости тестирования программы. Данная метрика определяется по формуле:

***Z(G) = e – v + 2p,***

где е — число дуг ориентированного графа G; ʋ — число вершин; р —

число компонентов связности графа.

Число дуг: e = 60.

Число вершин: ʋ = 50, p = 1.

Цикломатическое число Маккейба равно:

Z(G) = 60-50+2\*1 = 12.

1-й путь.1-2-3-4-5-6-7-8-9-10(да)-11-12-10(нет)-13-14-15(нет)-18-19-20 (нет)-35-36-37(нет)-40-41-42(да)-43-44-42(нет)-45-46-47(да)-48-49-47(нет)-50.

2-й путь. 1-2-3-4-5-6-7-8-9-10(нет)-13-14-15(нет)-18-19-20(да)-21-22(да)-23(да)-25-26-27(да)-28(да)-29-30-27(нет)-31(нет)-34-20(нет)-35-36-37(нет)-40-41 -42(нет)-45-46-47(нет)-50.

3-й путь. 1-2-3-4-5-6-7-8-9-10(нет)- 13-14-15(да)-16-17-15(нет)-18-19-20 (нет)-35-36-37(да)-38-39-37(нет)-40-41-42(нет)-45-46-47(нет)-50.

***Метрика Джилба***

Данная метрика определяет логическую сложность программы как насыщенность ее условными операторами IF–THEN–ELSE.

CL – количество условных операторов, характеризующее абсолютную сложность программы;

сl – насыщенность программы условными операторами, характеризующая относительную сложность программы (cl определяется как отношение CL к общему количеству операторов программы).

CL = 11,

cl = 11/(50-2-10) = 0.29 (количество операторов программы равно 38, поскольку блоки ввода-вывода, а также блоки «начало» и «конец», используемые при вычислении метрики Маккейба, не принимаются во внимание при вычислении метрики Джилба)

Максимальный уровень вложенности условного оператора CLI = 4.

***Метрика граничных значений***

Таблица 1 – Свойства подграфов программы.

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Свойства подграфов программы | Номер вершины выбора | | | | | | | | | | |
| 10 | 15 | 20 | 22 | 23 | 27 | 28 | 31 | 37 | 42 | 47 |
| Номера вершин  перехода | 11, 13 | 16, 18 | 21, 35 | 23, 34 | 24, 25 | 28, 31 | 29, 30 | 32, 34 | 38, 40 | 43, 45 | 48, 50 |
| Скорректированная сложность  вершины выбора | 4 | 4 | 16 | 13 | 13 | 5 | 2 | 3 | 4 | 4 | 4 |
| Номера вершин  подграфа | 10-12 | 15-17 | 20-34 | 22-33 | 22-33 | 27-30 | 29 | 32-33 | 37-39 | 42-44 | 47-49 |
| Номер нижней  границы подграфа | 13 | 18 | 35 | 34 | 34 | 31 | 30 | 34 | 40 | 45 | 50 |

Таблица 2 – Скорректированные сложности вершин графа программы.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины  графа программы | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| Скорректированная  сложность  вершины графа | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины  графа программы | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| Скорректированная  сложность  вершины графа | 1 | 1 | 1 | 16 | 1 | 13 | 13 | 1 | 1 | 1 | 5 | 2 | 1 | 1 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины  графа программы | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 |
| Скорректированная  сложность  вершины графа | 3 | 1 | 1 | 1 | 1 | 1 | 4 | 1 | 1 | 1 | 1 | 4 | 1 | 1 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Номер вершины  графа программы | 45 | 46 | 47 | 48 | 49 | 50 | Sa |
| Скорректированная  сложность  вершины графа | 1 | 1 | 4 | 1 | 1 | 0 | 110 |

Относительная граничная сложность программы S o определяется по формуле:

So = 1 – (*v*-1)/Sa

So = 1-(50-1)/ 110 = 0.55

***Итоговая таблица:***

|  |  |
| --- | --- |
| **Метрики сложности потока управления программ** | **Значения** |
| Метрика Маккейба, Z(G) | 12 |
| Абсолютная сложность программы CL по метрике Джилба | 11 |
| Относительная сложность программы cl по метрике Джилба | 0.29 |
| Максимальный уровень вложенности условного оператора CLI по метрике Джилба | 4 |
| Метрика граничных значений (абсолютная граничная сложность программы) Sa | 110 |
| Метрика граничных значений (относительная граничная сложность программы) So | 0.55 |