Министерство образования Республики Беларусь

Учреждение образования

«Белорусский государственный университет информатики и радиоэлектроники»

Кафедра метрологии и стандартизации

**Отчёт по лабораторной работе №1**

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

Вариант № 3

|  |  |
| --- | --- |
| Выполнил: | студент гр. 953505 Басенко К. А. |
| Проверил: | Грибович А.А. |

Минск 2020

Условие:

3. Ввести массив А (5, 7). Найти сумму элементов каждой строки,

максимальную и минимальную из этих сумм. Вывести массив, полученные

суммы, номера строк, где находятся максимальная и минимальная суммы.

**Исходный текст программы с комментариями:**  
#include <iostream>

using std::cin;

using std::cout;

int main()

{

//наш массив

int A[5][7];

//1е значение для суммы строки, 2е - для номера строки

//нет числа больше, чем самое большое

int smin[2] = { INT\_MAX , 0};

//нет числа меньше, чем самое маленькое

int smax[2] = { INT\_MIN , 0};

//для подсчета суммы строки

int buf = 0;

//считываем массив и сразу считаем суммы строк

//идем через строки

{

int i = 0;

while (i < 5)

{

//обнуление для подсчета суммы строки

buf = 0;

//идем через столбцы

int j = 0;

while (j < 7)

{

//вводим значение

cin >> A[i][j];

//считаем сумму строки

buf += A[i][j];

//переход на следующий столбец массива

j++;

}

//смотрим, меньше ли сумма текущей строки чем ранее занесенная минимальная сумма

if (buf < smin[0])

{

//1е значение для суммы

smin[0] = buf;

//2е значение для номера строки

smin[1] = i;

}

//смотрим, больше ли сумма текущей строки чем ранее занесенная максимальная сумма

if (buf > smax[0])

{

//1е значение для суммы

smax[0] = buf;

//2е значение для номера строки

smax[1] = i;

}

//переход на следующую строку массива

i++;

}

}

//вывод массива

//идем через строки

{

int i = 0;

while (i < 5)

{

//идем через столбцы

int j = 0;

while (j < 7)

{

//вывод элемента массива

cout << A[i][j] << "\t";

//переход на следующий столбец массива

j++;

}

//переход на новую строку вывода

cout << "\n";

//переход на следующую строку массива

i++;

}

}

//вывод минимальной суммы и номер ее строки

cout << "min: " << smin[0] << "\t line: " << smin[1] << "\n";

//вывод максимальной суммы и номер ее строки

cout << "max: " << smax[0] << "\t line: " << smax[1] << "\n";

//конец программы

return 0;

}

**Описание программы:**

Входные данные:

1. Массив A типа int, размером 5 на 7.

Внутренние данные:

1. Массив smin типа int, представляющий минимальную сумму строки и ее номер.
2. Массив smax типа int, представляющий максимальную сумму строки и ее номер.
3. Переменная buf типа int, представляющая сумму текущей строки.
4. Переменная i типа int, представляющая текущую строку массива.

9.

Назначение основных блоков программы:

7. Выполняется цикл 5 раз, который считает элементы массива A, считает сумму текущей строки, сравнивает сумму с минимальной и максимальной и заносит результат в smin или в smax.

10. Выполняется цикл 7 раз, который считает введенную i-ю строку массива A и сумму i-й строки, сумму заносит в buf.

22. Выполняется цикл 5 раз, который выводит массив A.

24. Выполняется цикл 7 раз, который выводит i-ю строку массива A.

**Расчет метрик Холстеда для разработанной программы:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| j | Оператор | f1j | i | Операнд | f2i |
| 1 | ; | 23 | 1 | 0 | 15 |
| 2 | [] | 20 | 2 | i | 10 |
| 3 | cout << | 13 | 3 | j | 8 |
| 4 | = | 12 | 4 | buf | 6 |
| 5 | < | 5 | 5 | smax | 5 |
| 6 | ++ | 4 | 6 | smin | 5 |
| 7 | while(…){…} | 4 | 7 | 1 | 4 |
| 8 | if(…){…} | 2 | 8 | A | 3 |
| 9 | > | 1 | 9 | 7 | 3 |
| 10 | += | 1 | 10 | 5 | 3 |
| 11 | return | 1 | 11 | INT\_MAX | 1 |
| 12 | cin>> | 1 | 12 | INT\_MIN | 1 |
| **η1 *= 12*** |  | ***N1 = 87*** | **η2 *= 12*** |  | ***N2 = 64*** |

Словарь программы: **η**= **η1 + η2** = 12 + 12 = 24;

Длина программы: ***N***= ***N1 + N2*** = 87 + 64 = 151;

Объем программы: ***V = N\*log2*(η) =** 151 \* *log2*(24) = 692.

**Расчет спена разработанной программы:**

Спен идентификатора – это число повторных появлений идентификатора (число появлений после его первого появления) в тексте программы. Идентификатор, встречающийся в тексте программы n раз, имеет спен, равный n—1.

Метрика Чепина базируется на анализе характера использования в программе переменных.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Индентификатор | A | i | j | buf | smin | smax | Сумммарный спен программы |
| Спен | 3 | 10 | 8 | 6 | 5 | 5 | 43 |

**Расчет полной метрики Чепина и метрики Чепина ввода/вывода**

1. Р – вводимые переменные, содержащие исходную информацию, но не модифицируемые в программе и не являющиеся управляющими переменными;

2. М – модифицируемые переменные и создаваемые внутри программы константы и переменные, не являющиеся управляющими переменными;

3. С – переменные, участвующие в управлении работой программного модуля (управляющие переменные);

4. Т – не используемые в программе («паразитные») переменные, например, вычисленные переменные, значения которых не выводятся и не участвуют в дальнейших вычислениях.

Q = p + 2m + 3c + 0,5t .

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | Полная метрика Чепина | | | | Метрика Чепина ввода/вывода | | | |
| Группа переменных | P | M | C | T | P | M | C | T |
| Переменные, Относящиеся к группе | A | smin, smax, buf | i, j | -- | A | smin, smax | -- | -- |
| Кол-во переменных в группе | 1 | 2 | 2 | 0 | 1 | 2 | 0 | 0 |
| Метрика Чепина | Q = 1 + 2\*2 + 2\*3 + 0 = 11 | | | | Q = 1 + 2\*2 + 0 + 0 = 5 | | | |