Защита лабораторной работы номер 1

Студент группы А-01-20 Котухов Михаил Дмитриевич

Вариант 9

Проверяющий Мохов Андрей Сергеевич

15.03.2021

Цель работы:

1. Владеть базовыми конструкциями и типами языка C++.
2. Уметь работать в среде программирования CodeBlocks.
3. Уметь автоматически проверять программы по эталонному вводу и выводу.

Задача:

1. Написать программу для построения гистограммы массива чисел.
2. В каждом столбце, если предыдущий столбец ниже, вместо \* используйте ^ на высоте предыдущего столбца. Аналогично для следующего столбца, но v. Если соседние столбцы оба ниже текущего и равны, используйте N.

Реализация:

Для выполнения индивидуальной части нам необходимо изменить только часть вывода программы

1. Чтобы избежать обращения к несуществующему элементу рассмотрим для начала все строки, кроме первой, тем самым выделяя первую строчку, которая больше второй, отдельным случаем:

if (i > 0)

{……………}

else if (bins[i + 1] - 1 == j) cout << "v"; (Случай: первая строка длиннее второй)

else cout << "\*";

2)Дальше у нас идет условие, которое отделяет последнюю строку отдельным случаем, когда последняя строка длиннее предпоследней.

if (i + 1 < bin\_count)

{………………………}

else if (bins[i - 1] - 1 == j && bins[i - 1] != bins[i]) cout << "^"; (Случай: последняя строка длиннее предыдущей)

else cout << "\*";

Тем самым в {……..} пункта 2) мы будем рассматривать 2-ую – предпоследние строки нашей гистограммы.

1. Рассмотрим 2-ую – предпоследнюю строки гистограммы:

if (bins[i - 1] - 1 == j && bins[i - 1] != bins[i + 1]) cout << "^";

(Данная строка длиннее предыдущей)

else if (bins[i + 1] - 1 == j && bins[i - 1] != bins[i + 1]) cout << "v";

(Данная строка длиннее следующей)

else if (bins[i + 1] < bins[i] && bins[i - 1] == bins[i + 1] && bins[i - 1] - 1 == j) cout << "N"; (данная строка длиннее и следующей и предыдущей, которые между собой равны)

else cout << "\*";

Далее для верной отладки вывода гистограммы объединяем данные пункты. Пункт 3 в пункт 2, пункт 2 в пункт 1.

Также необходимо учесть масштабирование стрелочек

Для этого введем новую переменную float c = 1.0;

В условии масштабирования допишем

if (max\_bin > MAX\_ASTERISK)

{

height = MAX\_ASTERISK \* (static\_cast<double>(bins[i]) / max\_bin);

c = MAX\_ASTERISK / max\_bin;

}

И при сравнении элементов гистограммы каждому элементу добавим коэффициент масштабирования с, домножив на него.

Исходный код:

#include <iostream>

#include <vector>

using namespace std;

int main()

{

const size\_t SCREEN\_WIDTH = 80;

const size\_t MAX\_ASTERISK = SCREEN\_WIDTH - 3 - 1;

size\_t number\_count;

cerr << "Enter number count: ";

cin >> number\_count;

vector<double> numbers(number\_count);

cerr << "Enter numbers: ";

for (int i = 0; i < number\_count; i++) cin >> numbers[i];

size\_t bin\_count;

cerr << "Enter number bins: ";

cin >> bin\_count;

vector<size\_t> bins(bin\_count);

double min = numbers[0];

double max = numbers[0];

for (double x : numbers) {

if (x < min) {

min = x;

}

else if (x > max) {

max = x;

}

}

double bin\_size = (max - min) / bin\_count;

for (size\_t i = 0; i < number\_count; i++) {

bool found = false;

for (size\_t j = 0; (j < bin\_count - 1) && !found; j++) {

auto lo = min + j \* bin\_size;

auto hi = min + (j + 1) \* bin\_size;

if ((lo <= numbers[i]) && (numbers[i] < hi)) {

bins[j]++;

found = true;

}

}

if (!found) {

bins[bin\_count - 1]++;

}

}

size\_t max\_bin = bins[0];

for (size\_t x : bins) {

if (x > max\_bin) {

max\_bin = x;

}

}

size\_t height;

for (int i = 0; i < bin\_count; i++)

{

height = bins[i];

float c = 1.0;

if (max\_bin > MAX\_ASTERISK)

{

height = MAX\_ASTERISK \* (static\_cast<double>(bins[i]) / max\_bin);

c = MAX\_ASTERISK / max\_bin;

}

if (bins[i] < 100) cout << " ";

if (bins[i] < 10) cout << " ";

cout << bins[i] << "|";

for (int j = 0; j < height; j++)

{

if (i > 0)

{

if (i + 1 < bin\_count)

{

if ((bins[i - 1] - 1)\*c == j && bins[i - 1]\*c != bins[i + 1]\*c) cout << "^";

else if ((bins[i + 1] - 1)\*c == j && bins[i - 1]\*c != bins[i + 1]\*c) cout << "v";

else if (bins[i + 1]\*c < bins[i]\*c && bins[i - 1]\*c == bins[i + 1]\*c && (bins[i - 1] - 1)\*c == j) cout << "N";

else cout << "\*";

}

else if ((bins[i - 1] - 1)\*c == j && bins[i - 1]\*c != bins[i]\*c) cout << "^";

else cout << "\*";

}

else if ((bins[i + 1] - 1)\*c == j) cout << "v";

else cout << "\*";

}

cout << endl;

}

}