**Лабораторная работа №1**

**Отчет**

Вариант 10

Маркелов Н.

А-01-19

**1)Постановка индивидуального задания**

Отображайте гистограмму вертикально без подписей, например:

\*\*\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*

\*\*\*\*

\*\*\*

\*\*\*

\*\*

\*

**2)Логика решения**

Для удобства и универсальности программы была создана отдельная функция, выводящая результат программы в требуемом виде. Логика её работы заключается в следующем: на функцию подается массив групп элементов (каждая ячейка массива содержит количество элементов, содержащихся в той или иной группе), размерность этого массива, т.е. количество групп. Внутри самой функции содержится 2 вложенных цикла. Внешний цикл ходит по строкам до тех пор, пока на строке печатается хотя бы 1 символ \*. Внутренний цикл ходит по всем элементам массива и печатает \* только в том случае, если в группе содержится столько же или больше элементов, чем номер строки, на которой в данный момент находится курсор, в противном случае печатается пробел. Связь между 2мя циклами осуществляется посредством логической переменной.

**3)Исходный код**

#include <iostream>

#include <vector>

using namespace std;

void FormatOut(vector <size\_t> korzins, size\_t kolvo\_korzins)

{

bool fl;

size\_t stroka;

stroka = 1;

do

{

fl = false;

for (size\_t i = 0; i < kolvo\_korzins; i++)

{

if (korzins[i] >= stroka)

{

cout << "\*";

fl = true;

}

else

{

cout << " ";

}

}

stroka++;

cout << "\n";

} while (fl);

return;

}

int main()

{

const size\_t MAX\_WIDTH = 80; //Макс. длина строки

const size\_t MAX\_ASTERISK = 76; //Макс. длина с учетом первых 4 симв.

size\_t number\_count; //Кол-во всех эл-тов

size\_t bin\_count; //Кол-во корзин

cerr << "bin\_count=";

cin >> bin\_count;

cerr << "number\_count=";

cin >> number\_count;

vector <double> numbers(number\_count); //Массив с элементами

cerr << "Vvedite massive" << "\n";

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

{

cin >> numbers[i];

}

vector <size\_t> bins(bin\_count, 0); //Массив групп эл-тов

double max = numbers[0];

double min = numbers[0];

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

{

if (numbers[i] < min)

min = numbers[i];

if (numbers[i] > max)

max = numbers[i];

}

for (double x : numbers)

{

size\_t bin\_index = (size\_t)((x - min) \* bin\_count / (max - min));

if (bin\_index == bin\_count)

{

bin\_index = bin\_index--; //Нумерация идет с 0

}

bins[bin\_index]++;

}

size\_t max\_count = 0;

for (size\_t b : bins)

{

if (max\_count > b)

max\_count = b; //Поиск самой здоровой корзины

}

for (size\_t bin : bins) //Вывод первых 4 симв.

{

if (bin < 100)

{

cout << " ";

if (bin < 10)

{

cout << " ";

}

}

cout << bin << "|";

size\_t height = 76 \* 1.0;

if (max\_count > MAX\_ASTERISK)

{

//Если кол-во > 76, масштабим

height = MAX\_ASTERISK \* ((static\_cast<double>(bin)) / max\_count); //Приведение типов

}

else

{

height = bin;

}

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

{

cout << "\*";

}

cerr << "\n";

}

cout << "\n\n";

size\_t out; //Форматный вывод

cout << "Need format output? 1/0 (Yes/No) >> ";

cin >> out;

if (out == 1)

{

cout << "\n";

FormatOut(bins, bin\_count);

}

return 0;

}